

NAIST-IS-DD1361207

**Doctoral Dissertation**

**Data-Intensive Science of  
Jamu Medicines**

**Sony Hartono Wijaya**

August 8, 2016

Department of Applied Informatics  
Graduate School of Information Science  
Nara Institute of Science and Technology  
Japan

A Doctoral Dissertation  
submitted to the Graduate School of Information Science,  
Nara Institute of Science and Technology  
in partial fulfillment of the requirements for the degree of  
Doctor of Science.

Thesis Committee:

Professor Shigehiko Kanaya	(Supervisor)
Professor Keiichi Yasumoto	(Co-supervisor)
Associate Professor Md. Altaf-Ul-Amin	(Co-supervisor)
Assistant Professor Naoaki Ono	(Co-supervisor)
Assistant Professor Tetsuo Sato	(Co-supervisor)

# Data-Intensive Science of Jamu Medicines<sup>1</sup>

Sony Hartono Wijaya

## Abstract

Popular traditional medicines from Indonesia are known as Jamu. A Jamu formula is composed of a single plant or a mixture of several plants, and the formulation of Jamu is generally developed based on the experience of users for decades or even hundreds of years. This study is intended to explore and identify interesting patterns in the formulation of Indonesian Jamu medicines by utilizing data-intensive science and machine learning methods. Initially, we proposed a new method to predict the relation between plant and disease using network analysis and supervised clustering. The plant-disease relations predicted by our method were evaluated in the context of previously published results and were found to produce around 90% successful predictions. Furthermore, we assessed the capability of binary similarity and dissimilarity measures to classify the Jamu pairs into match and mismatch efficacies by using Receiver Operating Characteristic (ROC) analysis. Hence, the selection of binary similarity and dissimilarity measures for multivariate analysis of Jamu medicines is data dependent. Out of 79 equations, the Forbes-2 similarity measure is recommended for studying the relationship between Jamu formulas. In addition, we extended our analysis by including metabolites information of the plants used as Jamu ingredients for predicting Jamu efficacy and identifying important metabolites. The Support Vector Machine (SVM) with linear kernel and Random Forest (RF) produced good classification models if we combined these classifiers with Single Filtering algorithm and Regularized RF. We also identified 94 significant metabolites associated to 12 efficacy groups by applying inTrees framework.

**Keywords:** Jamu, machine learning, binary data, similarity measures, ROC curve, SVM, Random Forest, feature selection

---

<sup>1</sup>Doctoral Dissertation, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1361207, August 8, 2016.

## Acknowledgements

In the name of Allah, the Most Gracious and the Most Merciful. Alhamdulillah, all praises to Allah for giving me the light and for enabling me to complete this dissertation.

Special appreciation goes to my supervisor, **Professor Shigehiko Kanaya**, for his supervision and continuous support. I deeply appreciated his directions, help, constructive comments, and suggestions during my study.

I am grateful to **Professor Keiichi Yasumoto** for taking his time to review my thesis and for his insightful recommendations. I would like to thank **Associate Professor Md. Altaf-Ul-Amin**, **Assistant Professor Naoaki Ono** and **Assistant Professor Tetsuo Sato** for all their valuable comments, suggestions and knowledge sharing. I want to extend particular thanks to **Professor Takaaki Nishioka**, **Associate Professor Tadao Sugiura**, and **Assistant Professor Ming Huang**.

I thank my fellow lab mates in **Computational Systems Biology Laboratory** for making the laboratory such a lovely place to not only conduct but also to research. In particular, I am grateful to **Mrs. Minako Ohashi** for her indefatigable aid and **Mrs. Aki Hirai Morita** for her support while accessing KNApSAcK Family Databases. To those who indirectly contributed to my study, I greatly appreciated your support and help. Thank you.

For **my parents, parents in law, brothers**, and **sisters**, I am running out of praise. There are no words that can testify my affection for you. Thank you very much. For the unfailing love and support, thanks dearest **Vassi Astrid Arasyi** for the patience, prayers, encouragement and understanding during my doctorate works. My lovely daughter **Mazaya Nauri Nashita**, has been a major motivation.



Finally, I wish to thank the staff of **International Student Affairs, NAIST** for their precious help and the Japanese **Ministry of Education, Culture, Sports, Science and Technology** for their wonderful scholarship, which I am a recipient of.

August 8, 2016

Sony Hartono Wijaya

## List of Abbreviations

AUC	Area Under the ROC curve
BA	Barabasi-Albert
CART	Classification and Regression Tree
CEC	Conjugated Enzyme Concentrate
CNN	Connecting Nearest Neighbor
D	Dissimilarity
DMB	Disorders of Mood and Behavior
DOA	Disorders of Appetite
E	Efficacy
EC	Efficacy Classes
ER	Erdos-Renyi
FML	Female Reproductive Organ Problems
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GC-MS	Gas Chromatography-Mass Spectrometry
GST	Gastrointestinal Disorders
HMS	Herbal Medicine Systems
ICD	International Classification of Diseases
inTrees	Interpretable Trees
KNN	k-Nearest Neighbor
MSC	Musculoskeletal and Connective Tissue
NADFC	The National Agency of Drug and Food Control
NCBI	National Center for Biotechnology Information
NMR	Nuclear Magnetic Resonance
OAO	One-Against-One

OOB	Out-of-bag
OTU	Operational Taxonomic Unit
PIN	Pain and Inflammation
PLS	Partial Least Squares
PPI	Protein-protein interaction
PR	Precision-Recall
RBF	Radial Basis Function
RF	Random Forest
ROC	Receiver Operating Characteristic
RRF	Regularized Random Forest
RSP	Respiratory Disease
S	Similarity
SF	Single Filtering algorithm
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
TPR	True Positive Rate
URI	Urinary Related Problems
WHO	World Health Organizations
WND	Wounds and Skin Infections

# Contents

<b>Chapter 1. Introduction</b> .....	<b>1</b>
1.1. Background.....	1
1.2. Objectives.....	5
1.3. Dissertation outline .....	5
<b>Chapter 2. Prediction of plant-disease relations in Jamu medicines</b> .....	<b>7</b>
2.1. Background.....	7
2.2. Datasets .....	9
2.3. Methods.....	12
2.3.1. Prediction of plant-disease relations .....	12
2.3.2. Development of Herbal Medicine Systems .....	15
2.4. Results and discussion .....	16
2.4.1. Construction and comparison of Jamu and random networks .....	16
2.4.2. Supervised clustering based on DPCLUSO.....	20
2.4.3. Assignment of plants to disease .....	23
2.4.4. Evaluation of supervised clustering based on DPCLUSO .....	23
2.4.5. Herbal Medicine Systems application .....	26
2.4.5.1. Requirements definition .....	27
2.4.5.2. System and software design.....	27
2.4.5.3. Implementation, integration and testing .....	30
2.4.5.4. Operation and maintenance .....	31
2.5. Summary .....	32
<b>Chapter 3. Finding a suitable binary similarity and dissimilarity</b>	
<b>measures</b> .....	<b>35</b>
3.1. Background.....	35

3.2. Datasets .....	38
3.3. Methods.....	39
3.3.1. Reducing the candidate equations .....	41
3.3.2. An ROC analysis and Cohen's Kappa .....	50
3.4. Results and discussion .....	51
3.4.1. Preliminary verification of the equations .....	51
3.4.2. ROC analysis of selected equations .....	56
3.5. Summary .....	67
<b>Chapter 4. Metabolomic studies of Jamu medicines .....</b>	<b>69</b>
4.1. Background.....	69
4.2. Datasets .....	72
4.3. Methods.....	73
4.3.1. Single Filtering algorithm .....	75
4.3.2. Support Vector Machine .....	76
4.3.3. Random Forest.....	77
4.3.3.1. Regularized Random Forest .....	79
4.3.3.2. Interpretable Trees .....	79
4.4. Results and discussion .....	81
4.4.1. Filtering efficacy-metabolite data.....	81
4.4.2. Prediction of Jamu efficacy.....	83
4.4.3. Identification of important metabolites .....	87
4.5. Summary .....	94
<b>Chapter 5. Conclusions .....</b>	<b>95</b>
<b>Appendices .....</b>	<b>113</b>
<b>A. List of diseases from ICD-10 .....</b>	<b>115</b>
<b>B. List of plants assigned to each disease .....</b>	<b>119</b>
<b>C. The mean AUCs for each disease class .....</b>	<b>127</b>

D. The inTrees framework.....	129
E. Feature selection using Regularized RF .....	131
F. List of compounds and extracted rules from Jamu medicines.....	135
G. Source code .....	173

## List of Figures

1.1.	Some plants used in the Jamu medicines.....	2
1.2.	Jamu products for treating urinary system diseases.....	4
2.1.	An illustration of a network connecting efficacy-Jamu-plant.....	10
2.2.	The concept of the methodology: Network construction based on ingredient similarity between individual Jamu medicines, network clustering and classification of medicinal plants to dominant disease.....	13
2.3.	The Waterfall Method used for developing Herbal Medicine Systems application.....	15
2.4.	The network consisting of 0.7% Jamu pairs with correlation value greater than or equal to 0.596.....	17
2.5.	Degree distributions of three Jamu networks roughly follow a power law.....	19
2.6.	Distribution of clusters based on matching score.....	21
2.7.	(a) Success rate and (b) the number of predicted plants with respect to matching score thresholds.....	22
2.8.	Distribution of 135 plants assigned based on 0.7% dataset with respect to the number of diseases they are assigned to.....	24
2.9.	Use case diagram of Herbal Medicine Systems.....	28
2.10.	Class diagram of Herbal Medicine Systems.....	30
2.11.	The Herbal Medicine Systems application: (a) find herbal medicines, (b) detail of crude drugs, and (c) maintain the HMS database.....	31
2.12.	The Herbal Medicine Systems' page on Google Play Store.....	32
3.1.	An illustration of the experimental flow.....	40
3.2.	Clustering of 56 binary similarity and dissimilarity measures in the context of Jamu data after removing algebraically redundant equations and equations that produce invalid coefficients.....	53

3.3.	The heatmap and dendrogram of remaining binary similarity and dissimilarity measures using Jamu data.....	55
3.4.	The heatmap and dendrogram of remaining binary similarity and dissimilarity measures using Kampo data. ....	56
3.5.	Scatter plot of the minimum distance vs. the mean of AUC. ....	62
3.6.	The ROC curves of Michael and Forbes-2 similarities for Jamu and random datasets.....	63
3.7.	The ROC curves for every disease class in Jamu data using Forbes-2 similarity coefficients.....	66
4.1.	The illustration of a network connecting efficacy, Jamu, plant, and metabolite.....	73
4.2.	The schematic diagram of the prediction of Jamu efficacy and identification of important metabolites for each efficacy group.....	75
4.3.	Illustration of the inTress framework. ....	80
4.4.	The distribution of Jamu data. SVM classifier with three different kernels was used as Single Filtering algorithm. ....	82
4.5.	The relationship between the number of trees and error rate in Random Forest tuneRF.....	84
4.6.	Confusion matrix from the prediction of Jamu efficacy based on its metabolites using JamuR data and SVM with a linear kernel. ....	86
4.7.	Distribution of 3,490 metabolites with respect to the number of efficacy groups they are assigned to.....	86
4.8.	The performance of RRF for selected values of $\lambda$ . (a) shows the number of features selected by RRF. (b) shows the mean accuracy of classifiers applied to the feature subsets selected by RRF for different $\lambda$ .....	88
4.9.	Illustration of rule selection for muscle and bone class (E11). ....	93



## List of Tables

2.1.	Representation of efficacy, Jamu, and plant in Fig. 2.1 as a two-dimensional matrix. ....	11
2.2.	The distribution of Jamu formulas according to 18 classes of disease.....	11
2.3.	Statistics of three datasets. ....	17
2.4.	The relation between disease classes defined in this study (E) and efficacy classes (EC) reported by Afendi et al. (2010).....	24
2.5.	The prediction result of plant-disease relations using matching score > 0.6.....	26
3.1.	List of 79 binary similarity and dissimilarity measures. ....	43
3.2.	Groups of identical equations.....	51
3.3.	Transformation of an equation into another by adding or multiplying by constants.....	54
3.4.	The ROC analysis and Cohen's Kappa score of Jamu data. ....	59
3.5.	The ROC analysis and Cohen's Kappa score of Kampo data.....	63
4.1.	Representation of Jamu, metabolites and efficacy in Fig. 4.1 as a two-dimensional matrix. ....	73
4.2.	The distribution of Jamu formulas according to its metabolites taken from plants used as main ingredients. ....	74
4.3.	The summary of filtered datasets.....	82
4.4.	The classification results of non-filter and filtered Jamu data using SVM and Random Forest classifiers.....	85
4.5.	The performance of classifiers after feature selection with RRF.....	88
4.6.	Distribution of extracted rules for each disease class. ....	90
4.7.	List of important metabolites extracted from selected rules. ....	91



## Chapter 1

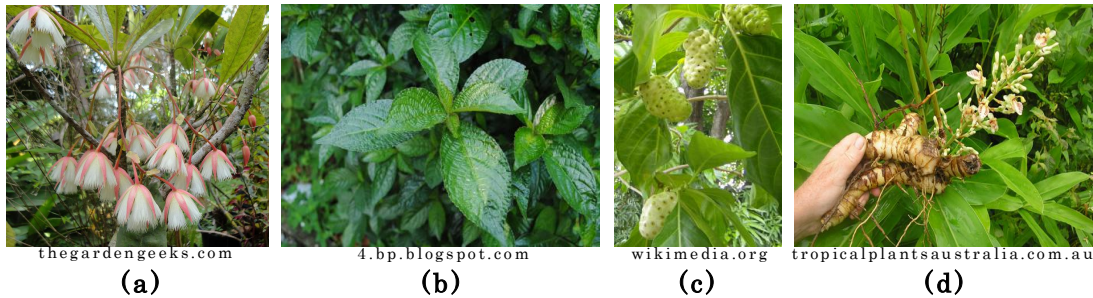
# Introduction

### 1.1. Background

The number of medicinal plants is estimated to be 40,000 to 70,000 around the world (Varpoorte et al. 2006). Many countries utilize these plants as blended herbal medicines for disease treatment and maintaining people in good health, such as China (traditional Chinese medicine), Japan (Kampo medicine), India (Ayurveda, Siddha, and Unani), and so on. The utilization of herbal medicines becomes popular in the last decades because of no side effect (Furnham 1996; Ernst 2003). As a country with the largest medicinal plant species in the world (Schippmann et al. 2002; Schippmann et al. 2006; Hanafi et al. 2006; Afendi, Okada, et al. 2012), Indonesian also uses medicinal plants as a constituent of herbal medicines, and it is known as Indonesian Jamu.

Jamu medicines are prepared from a single plant or a mixture of several plants as its constituents. It might be from a specific part of plants or a whole plant, such as rhizome of Java ginger (temulawak/*Curcuma xanthorriza*), leaf of kecibeling (*Strobilanthes crispus*), fruit of tamarind (asam jawa/*Tamarindus indica*) and so on. For some other plants, Jamu medicines utilize other parts of plant i.e. flower, bark, seed, aerial part, etc. In order to simplify the notation, we used the term plant to refer a part of the plant or a whole plant as Jamu ingredient. Traditionally, the formulation of Jamu is composed based on the experience of the users for decades or even hundreds of years. Some plants were selected and mixed as Jamu ingredients to obtain the desired effect, known as efficacy. In term of Jamu medicines, efficacy refers to the ability of Jamu formula to treat a disease and also a condition that prevents the body or mind working normally. For instance Jamu formula for urinary system disease, Indonesian will blend leaves of Ceylon olive (anyang-anyang/*Elaeocarpus grandiflorus*), leaves of

kecibeling (*Strobilanthes crispus*), fruits of cheese fruit (mengkudu/*Morinda citrifolia*) and rhizomes of galangal (lengkuas/*Alpinia galanga*) as Jamu constituents. Then, these plants were boiled with water before it was served. Fig. 1.1 illustrates some plants used as Jamu ingredients. Hence, the versatile scientific analysis is needed to support Jamu efficacy and safety. It leads to the exploration of a scientific aspect of Jamu for finding a reasonably ground truth, why some Jamu formulas can be used to cure a disease. Attaining this objective is in accordance with the 2010 policy of the Ministry of Health of Indonesian Government about Scientification of Jamu. Thus, it is required to systemize the formulations and develop basic scientific principles of Jamu to meet the requirement of Indonesian Healthcare System.



**Figure 1.1.** Some plants used in the Jamu medicines. (a) *Elaeocarpus grandiflorus*, (b) *Strobilanthes crispus*, (c) *Morinda citrifolia*, (d) *Alpinia galanga*.

Nowadays, Jamu medicine is produced commercially on an industrial scale, and the National Agency of Drug and Food Control (NADFC) of Indonesia supervises the production of Jamu medicines before its release for public use. Up to 2014, there were 1,247 Jamu factories in Indonesia (Indonesian Food Technologist 2014). Jamu companies have produced a lot of Jamu products with various efficacies. We can easily find Jamu products in the market for treating diseases such as urinary system disease, digestive system disease, diseases of the

immune system and so on. In case of urinary system disease, there are some Jamu products that can be applied i.e. Kejibeling from PT. Industri Jamu Borobudur (Fig. 1.2a), Lancar Seni from PT. Sido Muncul (Fig. 1.2b) and also the same product's name from IKOT Jamu Putri Gunung Jati (Fig. 1.2c). These Jamu are created by utilizing different composition of plants as Jamu ingredients, but it has the same efficacy for treating the same disease. Although Jamu companies produce their own Jamu formulas, it is clear that the efficacy of Jamu is mainly determined by the composition of plants used as Jamu ingredients (Pramono 2007). In order to provide a comprehensive understanding of Jamu efficacy, analysis of the relationship between Jamu and its efficacy is needed. The analysis is not only considering plants as Jamu ingredients but also including active compounds contained in those plants. Then, the discovery-driven research of Jamu through omics methodologies, i.e. metabolomics, proteomics, transcriptomics and genomics (Romero et al. 2006), can be used to discover Jamu medicines by exploring Jamu into a molecular level. Thus, it provides a better understanding of the formulation of Jamu, opens opportunities for developing new Jamu formulas and finds alternative plants as Jamu ingredients. Meanwhile, a standardized tool to disseminate information about various herbal medicines, especially Jamu, is needed, such as the formulation of herbal medicines, plants used as ingredients, omics, and also research findings related to herbal medicine studies. A mobile application can be chosen as an alternative solution to overcome the problems and also to promote the use of Jamu medicines.

Data-intensive science has become popular in the last decades. This study takes a data-driven approach by organizing large volumes of data from multiple sources and analyzing them using techniques tailored to explore and identify interesting patterns in the high-dimensional data through visualizations, simulations and various type of model building (Kelling et al. 2009). Recently, there are many databases and publications available from various fields that provide information about herbal medicines, plants, omics, and chemicals (Weber

& Kim 2016; Afendi, Okada, et al. 2012). Jamu studies can exploit those resources to build a scientific background of Jamu medicines. Some patterns of Jamu medicines can be mined by exploiting data-intensive science on the existing databases or newly collected data. We can extract information about a relationship between Jamu efficacy and plant as Jamu ingredients, and also a relationship between Jamu efficacy and its omics.



**Figure 1.2.** *Jamu products for treating urinary system diseases. (a) Kejibeling from PT. Industri Jamu Borobudur, (b) Lancar Seni from PT. Sido Muncul, and (c) Lancar Seni from IKOT Jamu Putri Gunung Jati.*

In the Jamu studies, Jamu formula and plant/crude drug relations are represented as binary feature vectors, denoting whether a particular plant is used or not as an ingredient. The relationships between Jamu formulas are not only reflected by the efficacy similarity but also by the ingredient similarity. One Jamu formula can be suggested as an alternative to the other one if they have relatively similar ingredients. For mathematical analysis, each Jamu formula is represented as a binary vector using 1 to indicate the presence of a plant and 0 otherwise. Hence, it is necessary to examine the binary similarity and dissimilarity coefficients of Jamu formulas to determine the appropriate measurement for finding a suitable mixing alternative of a crude drug. To our knowledge, there is no published work presenting a systematic way of finding an appropriate equation to measure binary similarity that performs well for certain data type or application.

## 1.2. Objectives

The objective of this research is to explore and identify interesting patterns in the formulation of Jamu medicines through studying the relations between plant and disease using network analysis and machine learning methods. When the effectiveness of a plant against a disease is firmly established, then further analysis of that plant can be extended to the molecular level to pinpoint the drug targets. Moreover, a mobile application is developed as a tool to promote and to disseminate information about herbal medicines, especially Jamu medicines. Furthermore, a suitable binary similarity or dissimilarity equations to measure the similarity and dissimilarity between Jamu pairs is required to obtain relatively similar Jamu formulas and better classification results. Therefore, a proper method to select binary similarity and dissimilarity measures is also examined.

## 1.3. Dissertation outline

This dissertation outline is organized as follows. In Chapter 2, the relationship between Jamu efficacies and plants used as Jamu ingredients is examined. We proposed a new method to predict the relation between Jamu and efficacy using network analysis and supervised clustering. Additionally, we also explained the development process of Herbal Medicine Systems application by using Waterfall method. Chapter 3 describes our proposed approach to select a suitable equation for studying the relationship between herbal medicine formulas using Receiver Operating Characteristic (ROC) analysis. Then, we extend Jamu study by considering plant's metabolites in Chapter 4. We utilized plant's active compounds to predict Jamu efficacy. Moreover, we determined important metabolites for each efficacy class. Finally, Chapter 5 gives conclusion remarks of this dissertation.





## Chapter 2

# Prediction of plant-disease relations in Jamu medicines

In this chapter, we explore the relationship between Indonesian herbal plants used in the Jamu medicines and the diseases, which are treated using Jamu medicines. We proposed a new approach to predict the relation between plant and disease using network analysis and supervised clustering. At the preliminary step, we collected Jamu data from KNApSAcK Family Databases and assigned the efficacy of 3,138 Jamu formulas to 116 diseases of International Classification of Diseases (ICD) version 10, which belong to 18 classes of disease from National Center for Biotechnology Information (NCBI). The correlation coefficients between Jamu pairs were determined based on their ingredient similarity. Networks were constructed and analyzed by selecting highly correlated Jamu pairs. Clusters were then generated by using the network-clustering algorithm DPCLUSO. By using matching score of a cluster, the dominant disease and high-frequency plant associated to the cluster were determined. The plant-disease relations predicted by our method were evaluated in the context of previously published results. In addition, we also utilized the KNApSAcK Family Databases for developing a mobile application called as Herbal Medicine Systems.

### 2.1. Background

A Jamu medicine is made by a single plant or a mixture of several plants. The combination of plants used as Jamu ingredients determines the efficacy of a Jamu formula. Therefore, in order to comprehensively understand the formulation of Jamu medicines, it may be useful to model the ingredients of Jamu and use this model to predict the efficacy of Jamu formulas.

The KNApSAcK Family Database can be used to comprehensively understand the medicinal usage of plants based on traditional and modern knowledge (Afendi, Okada, et al. 2012; Afendi, Ono, et al. 2013). This database system mainly consists of species-metabolite relations, but it also includes other databases such as Jamu and Kampo medicines, gene annotations, biological activity, food and health databases, and metalloprotein database.

Afendi et al. initiated and conducted scientific analysis of Jamu for finding the correlation between plants, Jamu, and their efficacies using statistical methods (Afendi et al. 2010; Afendi, Darusman, et al. 2013; Afendi, Darusman, et al. 2012). They used Biplot analysis, Partial Least Squares (PLS), and bootstrapping methods to summarize the data and also focused on prediction of Jamu formulations. These methods gave a good understanding about the relationship between plants, Jamu, and its efficacies. Among 465 plants used in 3,138 Jamu, 190 plants were shown to be effective for at least one efficacy and these plants were considered to be the main ingredients of Jamu. The other 275 plants are considered to be supporting ingredients in Jamu because their efficacy has not been established yet (Afendi et al. 2010).

Network biology can be defined as the study of the network representations of molecular interactions, both to analyze such networks and to use them as a tool to make biological predictions (Winterbach et al. 2013). This study includes modeling, analysis, and visualizations, which holds important task in life science today (Bachmaier et al. 2013). Network analysis has been increasingly utilized in interpreting high throughput data on omics information, including transcriptional regulatory networks (Chen et al. 2006), co-expression networks (Langfelder & Horvath 2008), and protein-protein interactions (Martin et al. 2010). By utilizing network biology, we can easily describe the relationship between entities in the network and also concentrate on the part of the network consisting of important nodes or edges. These advantages can be adopted for analyzing medicinal usage of plants in Jamu and diseases. Network analysis provides information about groups

of Jamu that are closely related to each other in terms of ingredient similarity and thus allows a precise investigation to relate plants to diseases. On the other hand, multivariate statistical methods such as PLS can assign plants to efficacy by global linear modeling of the Jamu ingredients and efficacies. However, there is still a lack of appropriate network based methods to learn how and why many plants are grouped in certain Jamu formula, and the combination rule embedding numerous Jamu formulas.

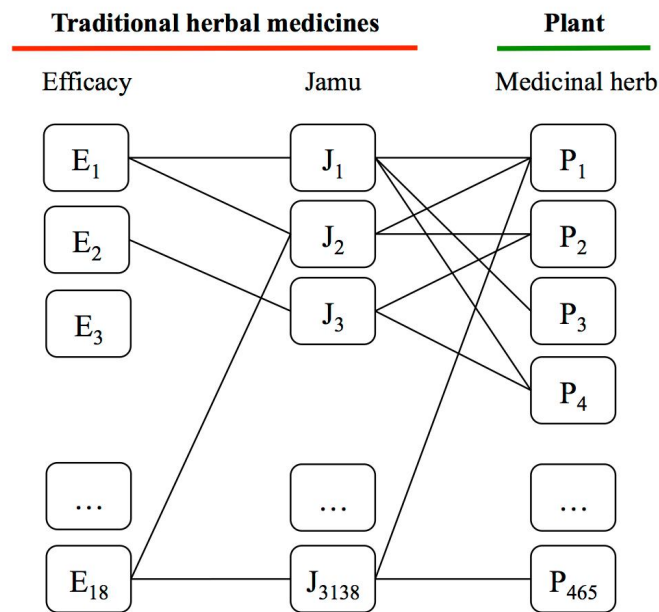
This chapter presents a network-based approach for prediction of plant-disease relations. We utilized the Jamu data from the KNApSAcK database. A Jamu network was constructed based on the similarity of their ingredients, and then Jamu clusters were generated using the network clustering algorithm DPCLUSO (Altaf-Ul-Amin et al. 2012; Altaf-Ul-Amin et al. 2006). Plant-disease relations were then predicted by determining the dominant diseases and plants associated with selected Jamu clusters. In addition, we also developed a standardized tool to disseminate our findings so that it could be utilized by the public. We expected a tool that not only provided information about Jamu medicines, but also other herbal medicines. Nowadays, a mobile application is the best option for disseminating research findings because of its flexibility and easiness. Most of the peoples can easily get information about what they need directly on their mobile phone.

## 2.2. Datasets

Fig. 2.1 illustrates the relationship between efficacy, Jamu, and plant. We used 3,138 Jamu formulas as the same as previous research taken from KNApSAcK Family Databases (Afendi et al. 2010; Afendi, Okada, et al. 2012). The set union of all formulas consists of 465 plants. In accordance with its ingredients, each Jamu is the assigned to plants it contains. Each Jamu formula consists of 1 to 26 plants, with average 4.904, and standard deviation 2.969. Jamu formulas were represented by binary vectors, which express the binary status of plants as

ingredients, 1 (presence) and 0 (absence). Thus, Jamu vs. plant relations were then organized as a 3138x465 binary matrix.

We assigned 3,138 Jamu formulas to 116 diseases of ICD version 10 from World Health Organizations (WHO, Appendix A) (World Health Organization 2010). Those 116 diseases are mapped to 18 classes of disease, which contain 16 classes of disease from NCBI (National Center for Biotechnology Information 1998) and two additional classes. For each Jamu formula, it corresponds to one or more efficacy groups/disease classes. Then, the efficacy-Jamu-plant relations can be represented as a matrix with dimension 3138x466 (Table 2.1), with rows represent Jamu formulas and columns represent plant as Jamu ingredient. We added one additional column to accommodate Jamu efficacy.



**Figure 2.1.** An illustration of a network connecting efficacy-Jamu-plant. The relations between efficacy and Jamu exhibit the efficacy of Jamu medicines. Conversely, the relations between Jamu and medicinal herbs represent utilization of plants in Jamu formula.

**Table 2.1.** Representation of efficacy, Jamu, and plant in Fig. 2.1 as a two-dimensional matrix.

Jamu	Plants						Efficacy
	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	...	P <sub>M</sub>	
J <sub>1</sub>	1	0	1	1	...	0	1
J <sub>2</sub>	1	1	0	0	...	0	1,18
J <sub>3</sub>	0	1	0	1	...	0	2
...	...	...	...	...	...	...	
J <sub>N</sub>	1	0	0	0	...	1	18

Table 2.2 shows the distribution of 3,138 Jamu into 18 classes of disease. According to this classification, most Jamu formulas are useful for relieving muscle and bone (E11), nutritional and metabolic diseases (E14), and the digestive system (E3). Furthermore, there is no Jamu formula classified into glands and hormones (E7), and neonatal disease classes (E12). We excluded 4 Jamu formulas, which are used to treat fever in the evaluation process because this symptom is very general and almost appeared in all disease classes.

**Table 2.2.** The distribution of Jamu formulas according to 18 classes of disease (disease classes are determined by NCBI in ID<sub>1</sub> to ID<sub>16</sub> and by the present study in ID<sub>17</sub> and ID<sub>18</sub> - represented by asterisks in the Ref. column).

IDs	Class of diseases	Ref.	Num. of Jamu	Percentage
E1	Blood and Lymph Diseases	NCBI	201	6.41
E2	Cancers	NCBI	32	1.02
E3	The Digestive System	NCBI	457	14.56
E4	Ear, Nose, and Throat	NCBI	2	0.06
E5	Diseases of the Eye	NCBI	1	0.03
E6	Female-Specific Diseases	NCBI	382	12.17
E7	Glands and Hormones	NCBI	0	-
E8	The Heart and Blood Vessels	NCBI	57	1.82
E9	Diseases of the Immune System	NCBI	22	0.70

<b>IDs</b>	<b>Class of diseases</b>	<b>Ref.</b>	<b>Num. of Jamu</b>	<b>Percentage</b>
E10	Male-Specific Diseases	NCBI	17	0.54
E11	Muscle and Bone	NCBI	649	20.68
E12	Neonatal Diseases	NCBI	0	-
E13	The Nervous System	NCBI	32	1.02
E14	Nutritional and Metabolic Diseases	NCBI	576	18.36
E15	Respiratory Diseases	NCBI	313	9.97
E16	Skin and Connective Tissue	NCBI	163	5.19
E17	The Urinary System	*	90	2.87
E18	Mental and Behavioral Disorders	*	21	0.67
	The number of Jamu classified into multiple disease classes		119	3.79
	The number of Jamu unclassified		4	0.13
	Total Jamu formulas		3138	100.00

## 2.3. Methods

### 2.3.1. Prediction of plant-disease relations

Jamu medicines consist of a combination of medicinal plants and are used to treat versatile diseases. In this work, we exploit the ingredient similarity between Jamu medicines to predict plant-disease relations. The concept of the proposed method is depicted in Fig. 2.2. In step 1, a network is constructed where a node is a Jamu medicine, and an edge represents high ingredient similarity between the corresponding Jamu pair. The similarity is represented by Pearson correlation coefficient (Sam Kash Kachigan 1991; Rodgers & Nicewander 1988), that is,

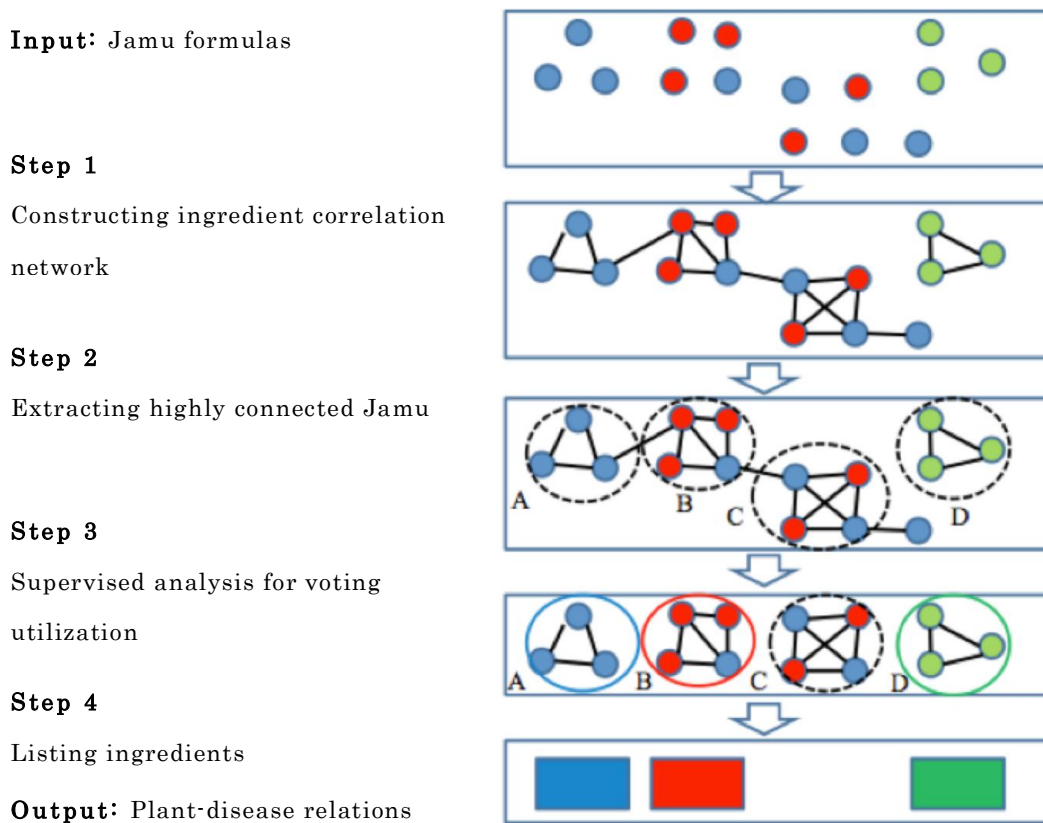
$$corr(X, Y) = \frac{\sum_{i=1}^l (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^l (x_i - \bar{x})^2 \sum_{i=1}^l (y_i - \bar{y})^2}}$$

where  $x_i$  is the weight of plant- $i$  in Jamu  $X$ ,  $y_i$  is the weight of plant- $i$  in Jamu  $Y$ ,  $\bar{x}$  is mean of Jamu  $X$ , and  $\bar{y}$  is mean of Jamu  $Y$ . The higher is similarity between Jamu pairs the higher the correlation value. In the present study,  $x_i$  and  $y_i$  are

assigned as 1 or 0 in cases the  $i$ -th plant is respectively included or not included in the formula. Under such condition, the Pearson correlation corresponds to four-fold point correlation coefficient, that is,

$$\text{corr}(X, Y) = \frac{ad - bc}{\sqrt{(a + b)(a + c)(b + d)(c + d)}}$$

where  $a$ ,  $b$ ,  $c$ , and  $d$  represent the numbers of plants included in both  $X$  and  $Y$ , in only  $X$ , in only  $Y$ , and in neither  $X$  nor  $Y$ , respectively.



**Figure 2.2.** *The concept of the methodology: Network construction based on ingredient similarity between individual Jamu medicines, network clustering and classification of medicinal plants to dominant disease. The nodes of the same color indicate the Jamu medicines used for the same disease.*

In step 2, the Jamu clusters are generated using network-clustering algorithm DPCLUSO. DPCLUSO is a general-purpose clustering algorithm and useful for finding overlapping cohesive groups in an undirected simple graph for any type of application. It ensures coverage and performs robustly in case of random addition, removal, and re-arrangement of edges in protein-protein interaction (PPI) networks (Altaf-Ul-Amin et al. 2012). DPCLUSO can generate clusters characterized by high density and identified by periphery i.e. the Jamu medicines belong to a cluster are highly cohesive and separated by a natural boundary. Such clusters contain potential information about plant-disease relations.

In step, 3 we assess disease-dominant clusters based on matching score represented by the following equation.

$$\text{matching score} = \frac{\text{number of Jamu belong to the same disease}}{\text{total number of Jamu in the cluster}}$$

Matching score of a cluster is the ratio of the highest number of Jamu associated to the same disease to the total number of Jamu in the cluster. We assign a disease to a cluster for which the matching score is greater than a threshold value. In step 4, we determine the frequency of plants associated to a cluster if and only if a disease is assigned to it in the previous step. The highest frequency plant associated to a cluster is considered to be related to the disease assigned to that cluster. True Positive Rate (*TPR*) or Sensitivity was used to evaluate resulted plants. *TPR* is the proportion of the true positive predictions out of all the true predictions, defined by the following formula (Li et al. 2008):

$$TPR = \frac{TP}{TP + FN}$$

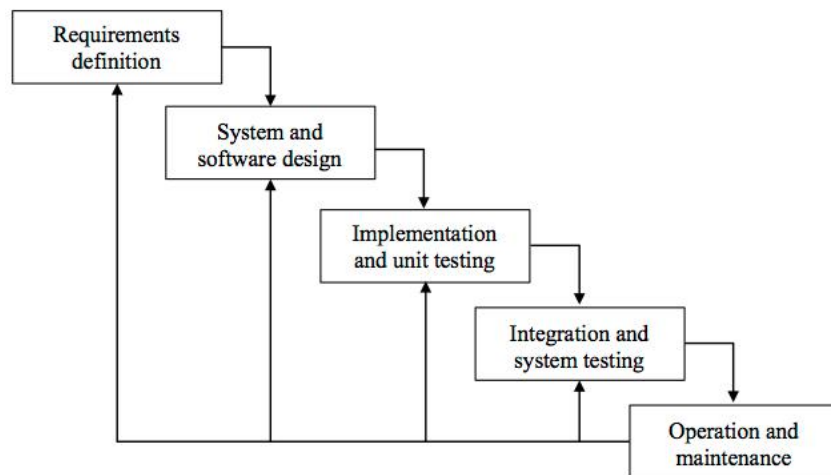
where true positive (*TP*) is the number of correctly classified and false negative (*FN*) is the number of incorrectly rejected entities. We refer the proposed method as supervised clustering because after generation of the clusters we narrow down the candidate clusters for further analysis based on supervised learning and thus



improve the accuracy of prediction of the proposed method.

### 2.3.2. Development of Herbal Medicine Systems

We developed a mobile application, called Herbal Medicine Systems (HMS), for promoting various kinds of herbal medicines and also disseminating research findings in the Android platform. The Waterfall method was used as a guideline to develop the HMS application (Khalifa & Verner 2000). The development of HMS with the Waterfall method mainly consists of five phases as follows: requirements definition, system and software design, implementation and unit testing, integration and system testing, and operation and maintenance (Fig. 2.3) (Sommerville 2010). The Unified Modeling Language (UML) ver. 2.0 was used in the early stages of the Waterfall Model as a modeling language (Burd et al. 2004). In the implementation phase, Android Studio (Zapata 2013; Zapata 2015) and SQLiteStudio (Halder 2015) were used to develop the mobile application and the database of crude drug systems, respectively. For testing, we applied Black-box testing to examine the functionality of HMS application. Then, the resulted application was deposited at Google Play Store as freely downloadable.



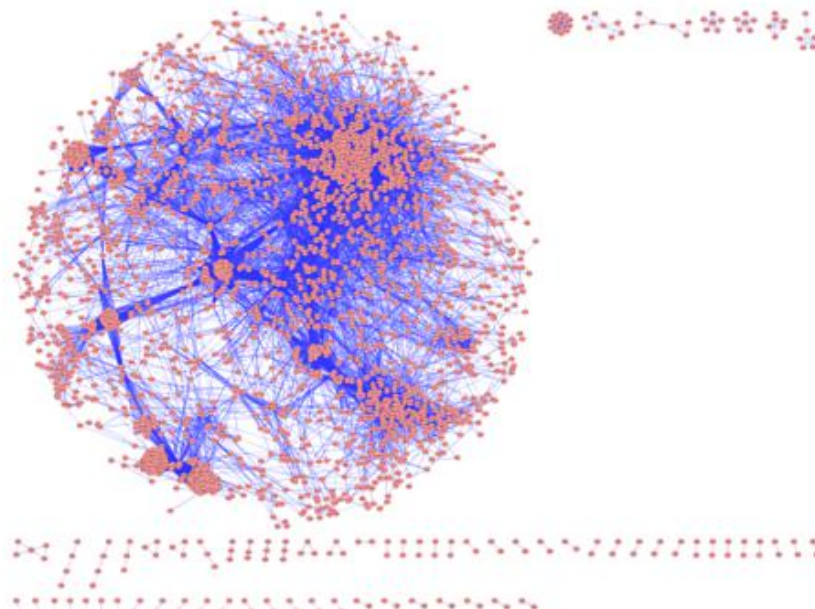
**Figure 2.3.** *The Waterfall Method used for developing Herbal Medicine Systems application.*

## 2.4. Results and discussion

### 2.4.1. Construction and comparison of Jamu and random networks

Initially, we calculated the similarity between Jamu pairs using correlation measure. The similarity measures between Jamu pairs were determined based on their ingredients. Corresponding to  $N$  Jamu formulas (3,138 in present case), there can be maximum  $(N \times (N - 1) / 2) = (3138 \times \frac{3137}{2}) = 4,921,953$  Jamu pairs. We sorted the Jamu pairs based on correlation value using descending order and selected top- $m$  (0.7%, 0.5%, and 0.3%,) pairs of Jamu formula to create three sets of Jamu pairs. The number of Jamu pairs for 0.7%, 0.5%, and 0.3% datasets are 34,454 pairs, 24,610 pairs, and 14,766 pairs and the corresponding minimum correlation values are 0.596, 0.665, and 0.718, respectively. The three datasets of Jamu pairs can be regarded as three undirected networks (step 1 in Fig. 2.2) consisting of 2,779, 2,496, and 2,085 Jamu formulas respectively (Table 2.3). Fig. 2.4 shows a visualization of 0.7% Jamu networks using Cytoscape software with Spring Embedded layout (Shannon et al. 2003). We verified that the degree distributions of the Jamu networks are somehow close to those of scale-free networks i.e. roughly are of power law type. However, in the high-degree region the power law structure is broken (Fig. 2.5). Nearly accurate relation of power laws between medicinal herbs and the number of formulas utilized them was observed in Jamu system but not in Kampo (Japanese crude drug system) (Afendi, Okada, et al. 2012). The difference of formulas between Jamu and Kampo can be explained by herb selection by medicinal researchers based on the optimization process of selection. Thus, the broken structure of power law corresponding to Jamu networks is associated with the fact that selection of Jamu pairs based on ingredient correlation leads to non-random selection. We also constructed random networks according to Erdos-Renyi (ER) model (Erdős & Rényi 1959), Barabasi-Albert (BA) model (Barabási & Albert 1999), and Vazquez's Connecting Nearest Neighbor (CNN) model (Vázquez 2003) of the same size corresponding to each of the real Jamu network. We used Cytoscape Network Analyzer plugin (Max

Planck Institut Informatik 2013) and R software for analyzing the characteristics of both the Jamu and the random networks.



**Figure 2.4.** *The network consisting of 0.7% Jamu pairs with correlation value greater than or equal to 0.596.*

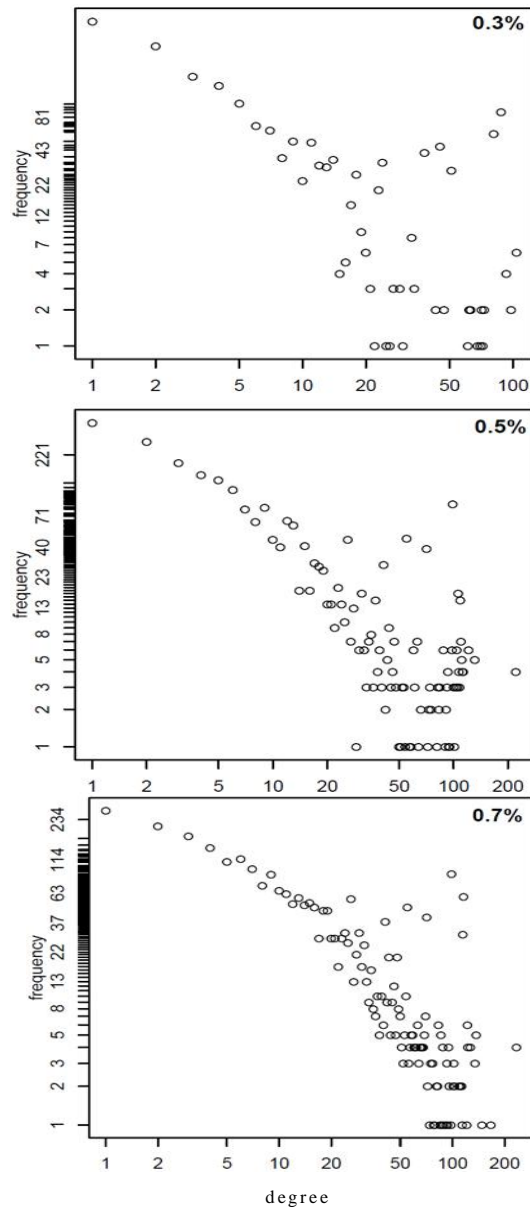
**Table 2.3.** *Statistics of three datasets.*

	<b>Parameters</b>	<b>0.7%</b>	<b>0.5%</b>	<b>0.3%</b>
Network Statistics	Total pairs	34,454	24,610	14,766
	Minimum correlation	0.596	0.665	0.718
	Number of Jamu formulas	2,779	2,496	2,085
	Average degree	24.8	19.7	14.2
	(Random network: ER)	(24.8±0.0)	(19.7±0.0)	(14.2±0.0)
	(Random network: BA)	(24.7±0.1)	(19.7±0.1)	(14.1±0.1)
	(Random network: CNN)	(24.7±0.4)	(19.7±0.4)	(14.0±0.4)
	Clustering coefficient	0.521	0.520	0.540
	(Random network: ER)	(0.009±0.000)	(0.008±0.000)	(0.007±0.000)
	(Random network: BA)	(0.030±0.001)	(0.028±0.001)	(0.026±0.001)
	(Random network: CNN)	(0.246±0.008)	(0.239±0.008)	(0.233±0.010)

	<b>Parameters</b>	<b>0.7%</b>	<b>0.5%</b>	<b>0.3%</b>
	Number of connected components	69	119	254
	(Random networks: ER, BA, CNN)	(1)	(1)	(1)
	Network diameter	15	17	20
	(Random network: ER)	(4.0±0.0)	(4.0±0.0)	(5.0±0.0)
	(Random network: BA)	(10.8±0.8)	(11.2±1.5)	(10.8±0.9)
	(Random network: CNN)	(14.6±1.9)	(14.1±1.4)	(14.7±1.3)
	Network density	0.008	0.008	0.007
	(Random network: ER)	(0.009±0.000)	(0.008±0.000)	(0.007±0.000)
	(Random network: BA)	(0.009±0.000)	(0.008±0.000)	(0.007±0.000)
	(Random network: CNN)	(0.009±0.000)	(0.008±0.000)	(0.007±0.000)
	Total number of clusters	1,746	1,411	938
DPclusO	Number of clusters with more than 2 Jamu	1,296	873	453
	(%)	(74.2)	(61.9)	(48.3)
	Number of Jamu formulas in the biggest cluster	118	104	89

We determined five statistical indexes i.e. average degree, clustering coefficient, the number of connected components, network diameter and network density of each Jamu network and also of each random network. The clustering coefficient  $C_n$  of a node  $n$  is defined as  $C_n = 2e_n/(k_n(k_n - 1))$ , where  $k_n$  is the number of neighbors of  $n$  and  $e_n$  is the number of connected pairs between all neighbors of  $n$ . The network diameter is the largest distance between any two nodes. If a network is disconnected, its diameter is the maximum of all diameters of its connected components. A network's density is the ratio of the number of edges in the network over the total number of possible edges between all pairs of nodes (which is  $n(n - 1)/2$ , where  $n$  is the number of vertices, for an undirected graph). The average number of neighbors and the network density are the same for the real and random networks of the same size as it is shown in Table 2.3. In case of 0.7% and 0.5% real networks, the clustering coefficient is roughly the same

and in case of 0.3% the clustering coefficient is somewhat larger. The number of connected components and the diameter of the Jamu networks gradually decrease as the network grows bigger by the addition of more nodes and edges.



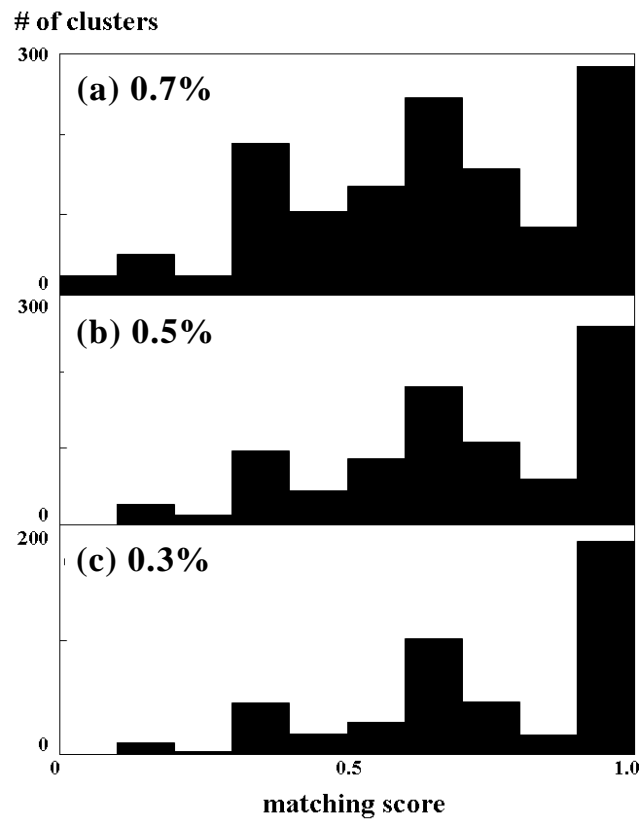
**Figure 2.5.** Degree distributions of three Jamu networks roughly follow a power law. The x-axis corresponds to the log of the degree of a node in the Jamu network, and the y-axis corresponds to the log of the number of Jamu.

Very different values corresponding to clustering coefficient, connected component, and network diameter imply that the Jamu networks are quite different from all three types of random networks. The differences between Jamu networks and ER random networks are the largest. Random networks constructed based on other two models are also substantially different from Jamu networks. Based on the fact i.e. the random networks constructed based on all three types of models are different from the Jamu networks, it can be concluded that structure of Jamu networks are reasonably biased and thus might contain certain information about plant-disease relations. Specially, a much higher value corresponding to clustering coefficient indicates that there are clusters in the networks worthy to be investigated. To extract clusters from the Jamu networks (step 2 in Fig. 2.2) we applied DPCLUSO network clustering algorithm (Altaf-Ul-Amin et al. 2012) to generate overlapping clusters based on density and periphery tracking.

#### 2.4.2. Supervised clustering based on DPCLUSO

While applying DPCLUSO, the parameter values of density and cluster property that we used in this experiment were 0.9 and 0.5, respectively (Altaf-Ul-Amin et al. 2006). The summary of clustering result by DPCLUSO is shown in Table 2.3. Because clusters consisting of two Jamu formulas are trivial clusters, for next steps we only use clusters each of which consists of 3 or more Jamu formulas. The number of total clusters increases along with the larger dataset, although the threshold correlation between Jamu pairs decreases. We evaluated the clustering result using matching score to determine dominant disease for every cluster (step 3 in Fig. 2.2). Thus, matching score is a measure to indicate how strongly a disease is associated with a cluster. Fig. 2.6 shows the distribution of the clusters with respect to matching score from three datasets. All dataset have the highest frequency of clusters at matching score  $> 0.9$  and overall most of the clusters have a higher matching score, which means most of the DPCLUSO generated clusters can be confidently related to a dominant disease.

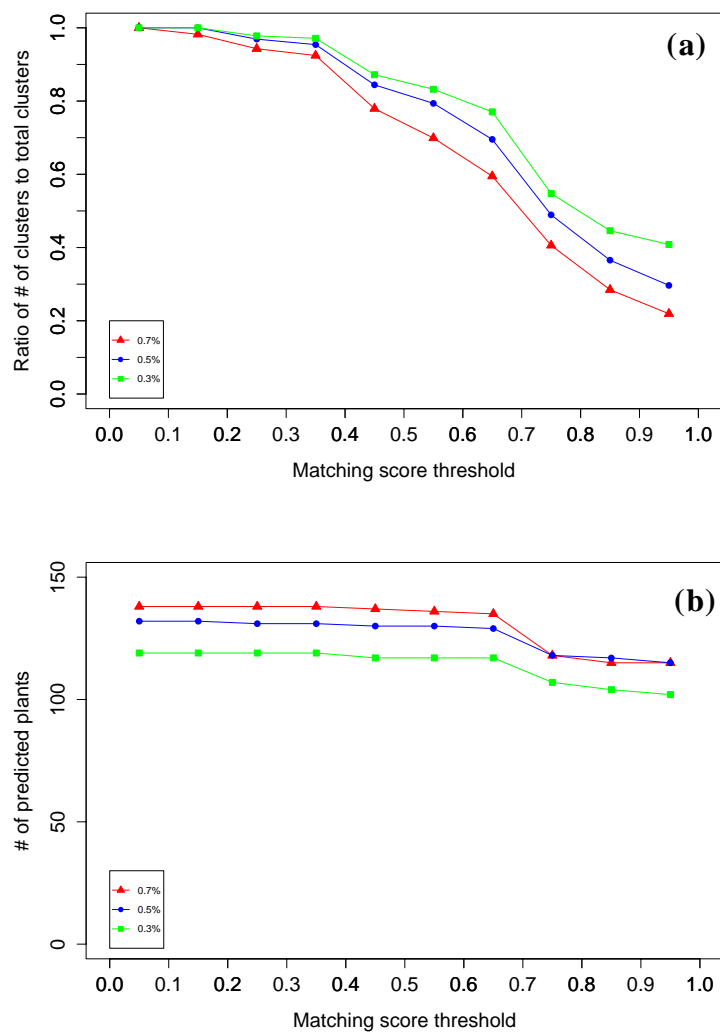
Furthermore, the number of clusters with matching score  $> 0.9$  is remarkably larger compared to the same in other ranges of matching score in case of the 0.3% dataset (Fig. 2.6c). If we compare the ratio of frequency of clusters at matching score  $> 0.9$  for every dataset, the 0.3% dataset has the highest ratio with 40.84% (of 453), compared to 29.67% (of 873) and 21.91% (of 1296), in case of 0.5% and 0.7% datasets, respectively. Thus, the most reliable species to disease relations can be predicted at matching score  $> 0.9$  corresponding to the clusters generated from 0.3% dataset.



**Figure 2.6.** *Distribution of clusters based on matching score.*

Fig. 2.7a shows the success rate for all three datasets with respect to threshold matching scores. The success rate is defined as a ratio of the number of clusters with matching score larger than the threshold to the total number of

clusters. As expected it tends to produce lower success rate if we decrease correlation value to create the datasets. However, more clusters are generated, and more information can be extracted when we lower the threshold correlation value. The success rate increases rapidly as the matching score decreases from 0.9 to 0.6 and after that the slope of increase of success rate decreases. Therefore, we empirically decide 0.6 as the threshold matching score to predict plant-disease relations.



**Figure 2.7.** (a) Success rate and (b) the number of predicted plants with respect to matching score thresholds.



#### 2.4.3. Assignment of plants to disease

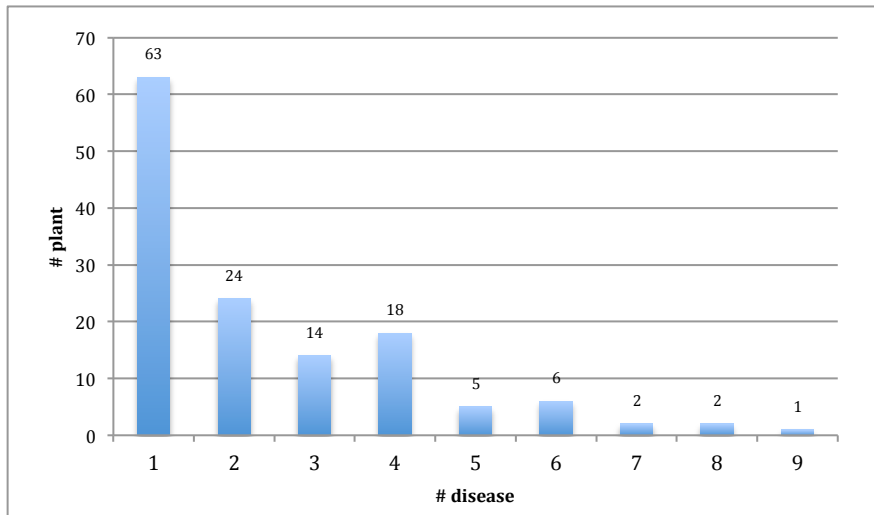
By using DPCLUSO resulted clusters, we assigned plants to classes of disease. Based on a threshold matching score, we assigned dominant disease to a cluster. Then, we assign a plant to a cluster by way of analyzing the ingredients of the Jamu formulas belonging to that cluster and determining the highest frequency plant i.e. the plant that is used for maximum number Jamu belonging to that cluster (step 4 in Fig. 2.2). Thus, we assign a disease and a plant to each cluster having matching score greater than a threshold. Our hypothesis is that the disease and the plant assigned to the same cluster are related.

The total number of assigned plants depends on matching score value. Fig. 2.7b shows the number of predicted plants that can be assigned to diseases in the context of matching score. With higher matching score value, the number of predicted plants assigned to classes of disease is supposed to remain similar or decrease, but the reliability of prediction increases. In Fig. 2.7b, a sudden change in the number of predicted plants is seen at matching score 0.6, which we consider as an empirical threshold in this work. Based on the 0.7% dataset, the largest number of plants (135 plants, Appendix B) was assigned to diseases. There are 63 plants assigned to only one class of disease, whereas the other 72 plants are assigned to at least two or more classes of disease (Fig. 2.8).

#### 2.4.4. Evaluation of supervised clustering based on DPCLUSO

We used previously published results (Afendi et al. 2010) as a gold standard to evaluate our results. The previous study assigned plants to 9 efficacies whereas we assigned the plants to 18 disease classes (16 from NCBI and two additional classes). For the sake of evaluation, we got done a mapping of the 18 disease classes to 9 efficacy classes by a professional doctor, which is shown in Table 2.4. In addition, Table 2.5 shows the prediction result of plant-disease relations for all three datasets corresponding to clusters with matching score greater than 0.6.

This table consists of corresponding efficacy, the number of assigned plants (#plant), the number of correctly predicted plants (TP), and TPRs respectively.



**Figure 2.8.** *Distribution of 135 plants assigned based on 0.7% dataset with respect to the number of diseases they are assigned to.*

**Table 2.4.** *The relation between disease classes defined in this study (E) and efficacy classes (EC) reported by Afendi et al. (2010).*

Class of disease	Ref.	Efficacy class
<b>E1</b> Blood and Lymph Diseases	NCBI	<b>EC7</b> Pain/inflammation (PIN)
<b>E2</b> Cancers	NCBI	<b>EC7</b> Pain/inflammation (PIN)
<b>E3</b> The Digestive System	NCBI	<b>EC4</b> Gastrointestinal Disorders (GST) <b>EC7</b> Pain/inflammation (PIN)
<b>E4</b> Ear, Nose, and Throat	NCBI	<b>EC7</b> Pain/inflammation (PIN)
<b>E5</b> Diseases of the Eye	NCBI	<b>EC7</b> Pain/inflammation (PIN)
<b>E6</b> Female-Specific Diseases	NCBI	<b>EC5</b> Female Reproductive Organ Problems (FML)
<b>E7</b> Glands and Hormones	NCBI	<b>EC7</b> Pain/inflammation (PIN)
<b>E8</b> The Heart and Blood Vessels	NCBI	<b>EC7</b> Pain/inflammation (PIN)
<b>E9</b> Diseases of the Immune System	NCBI	<b>EC7</b> Pain/inflammation (PIN)
<b>E10</b> Male-Specific Diseases	NCBI	<b>EC6</b> Musculoskeletal and Connective Tissue

<b>Class of disease</b>	<b>Ref.</b>	<b>Efficacy class</b>
		Disorders (MSC)
<b>E11</b> Muscle and Bone	NCBI	<b>EC6</b> Musculoskeletal and Connective Tissue Disorders (MSC)
<b>E12</b> Neonatal Diseases	NCBI	<b>EC7</b> Pain/inflammation (PIN)
<b>E13</b> The Nervous System	NCBI	<b>EC7</b> Pain/inflammation (PIN)
<b>E14</b> Nutritional and Metabolic Diseases	NCBI	<b>EC2</b> Disorders of Appetite (DOA) <b>EC4</b> Gastrointestinal Disorders (GST)
<b>E15</b> Respiratory Diseases	NCBI	<b>EC8</b> Respiratory Disease (RSP) <b>EC7</b> Pain/inflammation (PIN)
<b>E16</b> Skin and Connective Tissue	NCBI	<b>EC9</b> Wounds and Skin Infections (WND)
<b>E17</b> The Urinary System	*	<b>EC1</b> Urinary Related Problems (URI)
<b>E18</b> Mental and Behavioral Disorders	*	<b>EC3</b> Disorders of Mood and Behavior (DMB)

We determined TPR corresponding to a disease/efficacy class by calculating the ratio of the number of correct prediction to the number of all predictions. When a disease corresponds to more than one efficacy, the highest TPR can be considered as the TPR for the corresponding disease. For all three datasets, the TPR corresponding to each disease is roughly 90% or more. The 0.3% dataset consists of Jamu pairs with higher correlation values and based on this dataset 117 plants are assigned to 14 disease classes. The 0.7% dataset contains more Jamu pairs and assigned plants to 11 disease classes, one less disease class compared to 0.5% dataset. The two disease classes covered by 0.3% dataset but not covered by 0.5% and 0.7% datasets are the nervous system (E13) and disease of the immune system (E9). The only disease class covered by 0.3% and 0.5% datasets but not covered by 0.7% dataset is mental and behavioral disorders (E18). The larger dataset network tends to have lower coverage of disease classes. The number of Jamu pairs, i.e. the number of edges in the network, affects the number of DPCLUSO resulted clusters and number of Jamu formulas per cluster. As a consequence, for the larger dataset networks, the success rate becomes lower, and

the coverage of disease classes is lower, but the prediction of more plant-disease relations can be achieved.

**Table 2.5.** *The prediction result of plant-disease relations using matching score > 0.6.*

Disease Corresponding		0.7% dataset			0.5% dataset			0.3% dataset		
class	efficacy	#plant	TP	TPR	#plant	TP	TPR	#plant	TP	TPR
E1	EC7	26	22	0.85	24	20	0.83	24	20	0.83
E2	EC7	1	1	1.00	5	5	1.00	1	1	1.00
E3	EC4	42	42	1.00	33	33	1.00	28	28	1.00
	EC7		38	0.90		30	0.91		25	0.89
E4	EC7	0	0	-	0	0	-	0	0	-
E5	EC7	0	0	-	0	0	-	0	0	-
E6	EC5	38	38	1.00	37	37	1.00	32	32	1.00
E7	EC7	0	0	-	0	0	-	0	0	-
E8	EC7	10	8	0.80	8	7	0.88	6	5	0.83
E9	EC7	0	0	-	0	0	-	1	1	1.00
E10	EC6	6	4	0.67	2	0	-	3	1	0.33
E11	EC6	65	65	1.00	71	71	1.00	60	60	1.00
E12	EC7	0	0	-	0	0	-	0	0	-
E13	EC7	0	0	-	0	0	-	5	5	1.00
E14	EC2	54	44	0.81	45	36	0.80	35	26	0.74
	EC4		54	1.00		45	1.00		35	1.00
E15	EC7	38	37	0.97	34	34	1.00	33	33	1.00
	EC8		31	0.82		30	0.88		29	0.88
E16	EC9	32	31	0.97	32	32	1.00	27	27	1.00
E17	EC1	13	13	1.00	9	9	1.00	8	8	1.00
E18	EC3	0	0	-	5	5	1.00	4	4	1.00
Total assigned plants		135			129			117		

#### 2.4.5. Herbal Medicine Systems application

We followed the Waterfall method to develop the HMS application. First, we defined the HMS requirements. Second, we designed the HMS systems, including database and also the user interface. Third, we implemented each use case obtained in the previous step as a module and also tested it. At the same time, we

did the integration of the new module with the main module and also applied integration testing. The last was operation and maintenance of the HMS application.

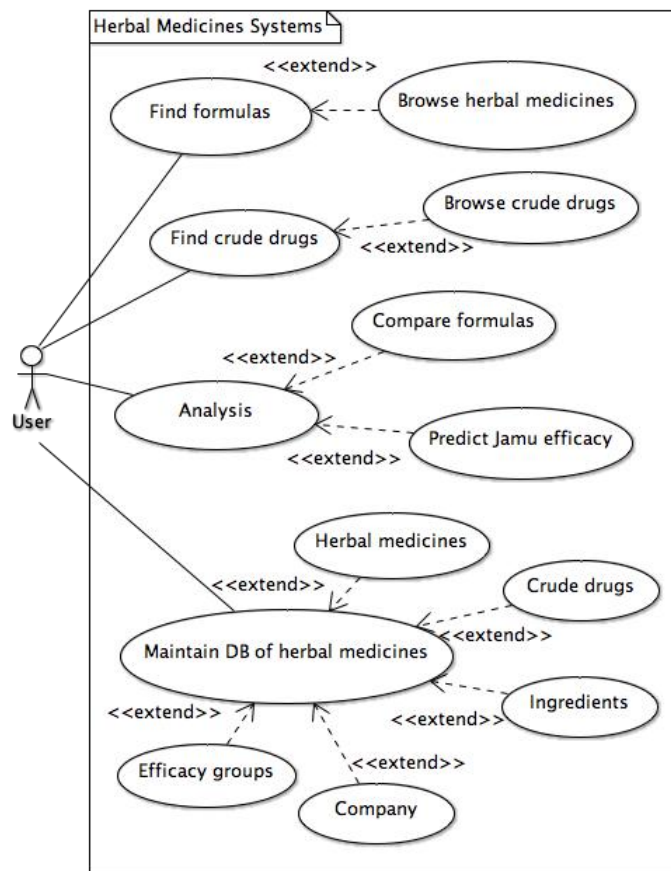
#### 2.4.5.1. Requirements definition

Initially, we defined and captured the functional requirements of HMS by examining KNApSAcK Family Databases, i.e. KNApSAcK Jamu and KNApSAcK Kampo ([http://kanaya.naist.jp/KNApSAcK\\_Family/](http://kanaya.naist.jp/KNApSAcK_Family/)). Next, we modeled the interactions between a user and HMS by utilizing use case diagram. Fig. 2.9 shows the use case diagram of HMS, which encompasses one actor and 13 use cases (Burd et al. 2004). Each use case in the use case diagram represents a list of actions functionality of the HMS to achieve a goal. The HMS mainly consists of four features, i.e. find formulas/recipes based on their name or efficacy, find plants/crude drugs based on their name or effect, analyze herbal medicines, and maintain the database of HMS. The user of HMS will input a query to obtain their expected information or interact with the application by touching the menu buttons. The analysis of herbal medicines menu covers a comparison between ingredients of herbal medicine formulas and prediction of Jamu efficacy based on plants used as Jamu ingredients. Furthermore, the HMS application can be used for modifying and cross-searching various kinds of crude drug systems.

#### 2.4.5.2. System and software design

We used Jamu and Kampo databases to develop a database of Herbal Medicine Systems by using data warehouse pre-processing technique, i.e. extraction, cleaning, transformation, loading and refreshing (Han et al. 2011). Both Jamu and Kampo databases consist of four tables: gakumei (scientific name), haigo (combination), Kampo/Jamu, and syouyaku (raw drug). Each table is composed of various columns, where each column indicates an attribute. The attributes of every table in the Jamu database are as follows: gakumei (gname, sid), haigou (fid, jid, company, brand, sid\_g), Jamu (jid, cname, jname, effect,

effectgroup, jamusource) and syouyaku (sid, sname, iname, ename, gname, position, effect, comment, reference). On the other hand, the attributes of every table in the Kampo database are as follows: gakumei (gname, sid), haigou (fid, kid, reference, sid\_g), Kampo (kid, kname\_j, kname, kname\_k, effect), and syouyaku (sid, sname\_j, sname\_k, name, gname, effect, comment, reference).








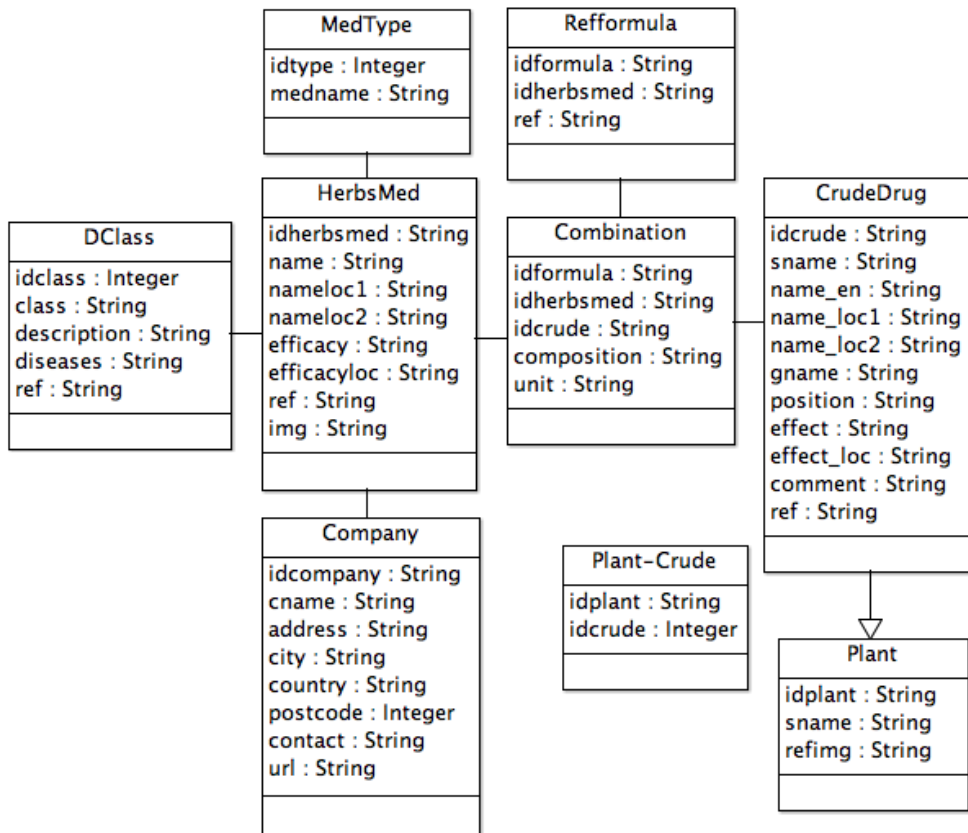
**Figure 2.9.** Use case diagram of Herbal Medicine Systems.

We analyzed and *extracted* important attributes from the Jamu and Kampo databases of the KNapSack database systems, related to the aforementioned use case in the requirements definition, i.e. we obtained all information related to finding crude drug use case in the syouyaku table of Jamu and Kampo databases.

Furthermore, we designed the class diagram to develop the database of Herbal Medicine Systems (Burd et al. 2004). Fig. 2.10 indicates the HMS domain model class diagram. The HMS class diagram consists of nine classes, i.e. type of herbal medicines (MedType), efficacy groups (DClass), list of herbal medicines (HerbsMed), Company, reference of used formulas (Reformula), Combination, list of crude drugs (CrudeDrug), Plant, and relationship between plant and crude drugs (Plant-Crude).

Jamu and Kampo formulas were considered as preloaded herbal medicines in the HMS application. Each Jamu has only one formula, and one Kampo might consist of more than one formula. We started to *clean* Jamu and Kampo data by excluding formulas without efficacy information. For all Jamu and Kampo formulas in the KNApSAcK Family Databases, only 3,027 Jamu and 234 Kampo have information related to their efficacies. In total, both data with efficacy information are composed by 1,023 kinds of crude drugs. We also *transformed* some attributes from Jamu and Kampo databases to maintain its consistency and to satisfy the second normal form (Connolly & Begg 2005), e.g. split the data in the sid column of gakumei table into two or more rows. The rest of data warehouse pre-processing technique will be explained in the next sections.

Furthermore, we also designed the user interface of HMS application. The user interface of HMS is divided into three sections, i.e. header, body, and footer. In the header and body sections, it contains application title and main activities, i.e. input, output, or modify the data. In the footer section, there is a list of application features as follows: find herbal medicine formulas based on their name or efficacy () , find crude drugs based on their name or effect () , analyze (compare herbal medicine formulas and predict Jamu efficacy, ) , maintain the database of HMS () , and the description of HMS () .



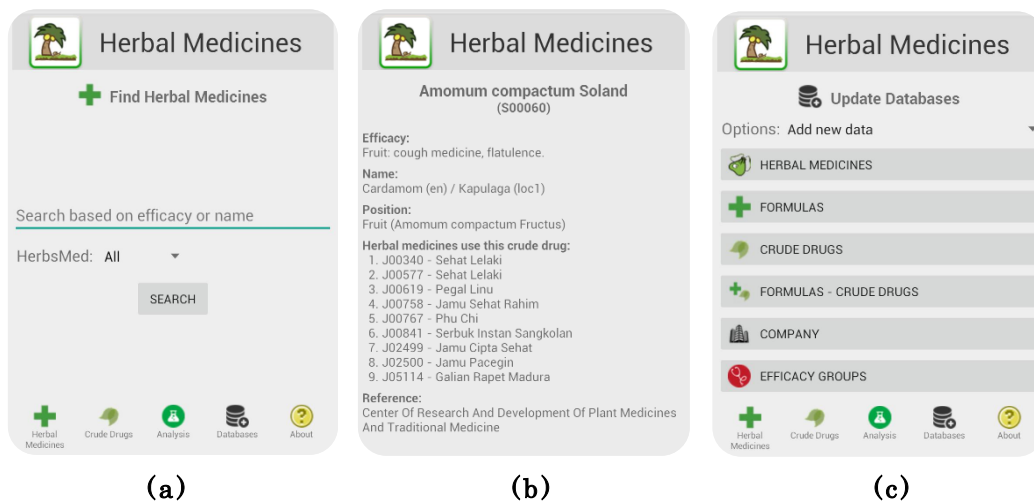
**Figure 2.10.** Class diagram of Herbal Medicine Systems.

### 2.4.5.3. Implementation, integration and testing

We created the database of HMS from domain class diagram (Fig. 2.10) by using SQLiteStudio 3.0.6. Each class in the Fig. 2.10 becomes a two-dimensional table, where columns represent attributes and rows represent objects. The HMS database is composed of nine tables with the attributes in each table are equal to attributes of HMS domain class diagram. After the implementation of HMS database, the selected data from Jamu and Kampo databases were *loaded* into HMS database. In addition, we translated Kampo efficacies from Japanese to English to accommodate bilingual transaction processing. The tools used to develop the HMS application were as follows: Java and XML as scripting languages and Android Studio Build #AI-141.2168647 as implementation tools.



Fig. 2.11 shows some of the implementation results of HMS. The unit testing was done by using Black-box testing. Initially, we defined testing scenarios and compared the expected result to the actual result. The HMS application was tested by using Android Virtual Device (with specification 4.7" 768x1280 pixels, OS Android Jelly Bean) and Sony Xperia M2 Dual D2302 (OS Android KitKat). The HMS application worked perfectly when we did the unit testing activities i.e. installation, loading the application and database, searching the herbal medicines and crude drugs, analyzing (comparing the herbal medicine formulas and prediction of Jamu efficacy), and maintaining the HMS database. Based on the system testing results, we concluded that the HMS conforms to the requirement definitions.

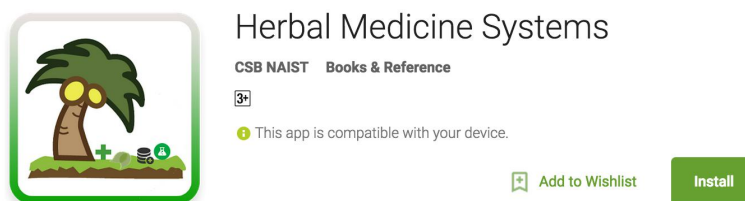


**Figure 2.11.** *The Herbal Medicine Systems application: (a) find herbal medicines, (b) detail of crude drugs, and (c) maintain the HMS database.*

#### 2.4.5.4. Operation and maintenance

Fig. 2.12 shows the main page of HMS application on Google Play store. Meanwhile, we have refreshed the HMS database with new data collected from many references, i.e. data from NADFC of Indonesia, packages of herbal medicines, books of herbal medicine, the Internet and other related databases. An

updated version of HMS database will be included in the latest update of the HMS application.



**Figure 2.12.** *The Herbal Medicine Systems' page on Google Play Store.*

## 2.5. Summary

In this chapter, we applied our proposed method for data mining of Jamu formulas accumulated in KNAPSAcK database by integrating network clustering and selection of clusters based on supervised learning. Jamu networks were constructed based on correlation similarities between Jamu formulas and then network-clustering algorithm DPCLUSO was applied to generate high-density Jamu modules. For the analysis of the next steps, potential clusters were selected by supervised learning. The successful clusters containing several Jamu related to the same disease might be useful for finding plants as the main ingredients for that disease, and the lower matching score value clusters will be associated with varying plants which might be supporting ingredients. By applying the proposed method, important plants from Jamu formulas for every class of disease were determined. The plant-disease relations predicted by proposed network-based method were evaluated in the context of previously published results and were found to produce a TPR of 90%. For the larger dataset networks, success rate and the coverage of disease classes become lower, but the prediction of more plant-disease relations can be achieved.

In addition, we also developed a mobile application called Herbal Medicine Systems to disseminate our finding, to promote the use of herbal medicine and also to share information about various herbal medicines for disease treatment

and maintaining people in good health. HMS is preloaded with Indonesia Jamu and Japanese Kampo medicines, collected from the KNAPSAcK Family Databases. The HMS application mainly consists of four features i.e. find formulas/recipes based on their name and efficacy, find crude drugs based on their name and effect, analyze (compare herbal medicine formulas and predict the efficacy of crude drug combinations based on the formulation of Indonesian Jamu), and maintain the database of HMS. Furthermore, the user of HMS application can modify the existing traditional medicines by adding other traditional medicine systems or new formulas. The black-box testing results show that the functionality of HMS application conforms all functional requirements. The Herbal Medicine Systems application has been released at Google Play store, which can be downloaded and used freely.



## Chapter 3

# Finding a suitable binary similarity and dissimilarity measures

We presented an organized way to select a suitable equation for studying the relationship between herbal medicine formulas, i.e. Indonesian Jamu and Japanese Kampo. We started our study by collecting 79 binary similarity and dissimilarity equations from literature. In the early stages, we reduced algebraically redundant equations and equations that produce invalid values or relatively similar coefficients when applied to all datasets. In addition, we eliminated some equations based on agglomerative hierarchical clustering result because they were very closely related to other equations in the same cluster. Finally, we selected 45 equations for our analysis. The ROC curve analysis was then performed to assess the capabilities of these equations to separate herbal medicine pairs having the same and different efficacies.

### 3.1. Background

Binary features have been commonly used to represent a great variety of data (Consonni & Todeschini 2012; Legendre & Legendre 1998; Batagelj & Bren 1995), expressing the binary status of samples as presence/absence, yes/no, or true/false. It has many applications in the bioinformatics, chemometrics, and medical fields (Afendi et al. 2010; Afendi, Okada, et al. 2012; Auer & Bajorath 2008; Kedariseti et al. 2014; Zhou et al. 2015; Tibshirani et al. 2004; Pinoli et al. 2015; Kangas et al. 2014; Ohtana et al. 2014; Abe et al. 1990; Willett et al. 1998; Flower 1998; Godden et al. 2000; Agrafiotis et al. 2001; Rojas-Cherto et al. 2012; Fligner et al. 2002), as well as in pattern recognition, information retrieval,

statistical analysis, and data mining (Zhang & Sargur N. Srihari 2003; Zhang & Sargur N Srihari 2003). The choice of an appropriate coefficient of similarity or dissimilarity is necessary to evaluate multivariate data represented by binary feature vectors because different similarity measures may yield conflicting results (Kosman & Leonard 2005). Choi *et al.* collected binary similarity and dissimilarity measures used over the last century and revealed their correlation through the hierarchical clustering technique (Choi et al. 2010). They also classified equations into two groups based on inclusion and exclusion of negative matches. Consonni & Todeschini proposed five new similarity coefficients and compared those coefficients with some well-known similarity coefficients (Consonni & Todeschini 2012). Three of the five similarity coefficients are less correlated with the other common similarity coefficients and need an investigation to understand their potential. Meanwhile, Todeschini *et al.* reported an analysis of 44 different similarity coefficients for computing the similarities between binary fingerprints by using simple descriptive statistics, correlation analysis, multidimensional scaling Hasse diagrams, and their proposed method ‘atemporal target diffusion model’ (Todeschini et al. 2012).

Kampo formulas are traditional medicines from Japan. These are generally prepared by a combination of crude drugs. In total, 294 Kampo formulas are listed in the Japanese Pharmacopoeia of 2012, and it can be used for self-medication (Okada et al. 2016). Currently, many researchers have done Kampo studies to unveil the complex systems of Kampo medication and to reveal the scientific aspect of its relevance to modern healthcare.

In the Jamu and Kampo studies, the relationships between plants, Jamu, and efficacies are represented as binary feature vectors, indicate whether a particular plant is used or not as Jamu ingredients. This leads to determine important plants for every disease class using global and local approaches (Afendi et al. 2010; Afendi, Okada, et al. 2012; Wijaya et al. 2014). However, each Jamu

formula usually uses a few plants. Thus, most of the Jamu vectors contain a few 1s and many 0s. Consequently, the number of plants that are used simultaneously in Jamu pairs is much smaller than the number of plants that are not used simultaneously as Jamu ingredients. Therefore, in order to find relatively similar Jamu formulas, the high number of negative matches might influence the calculation of binary similarity or dissimilarity between Jamu pairs. On the other hand, there is no guarantee that negative co-occurrence between two entities is identical (da Silva Meyer et al. 2004). Hence, it is necessary to examine the coefficients of binary similarity/dissimilarity measures of Jamu formulas to select a proper measurement for finding a suitable mixing alternative of a target crude drug.

Currently, there are several methods to measure the quality of classifiers (Demšar 2006; Lim et al. 2000) such as the Receiver Operating Characteristic (ROC) curves (Metz 1978; Davis & Goadrich 2006), Precision-Recall (PR) curves (Manning & Schütze 1999; Davis & Goadrich 2006), Cohen's Kappa scores (Cohen 1960; Ben-David 2007), and so on. The ROC curve is a very powerful tool for measuring classifiers' performance in many fields, especially in the machine learning and binary-class problems (Ben-David 2008). The purpose of ROC analysis is similar to that of the Cohen's Kappa, which is mainly used for ranking classifiers. The ROC curve conveys more information than Cohen's Kappa in a sense that it can also visualize the performance of a classifier by a curve instead of generating just a scalar value. In this study, we propose a method to select the most suitable similarity measures in the context of classification based on False Positive Rates (FPRs) and True Positive Rates (TPRs) by using ROC curve analysis. We discuss the step-by-step development of this method by applying it to assess the similarity of herbal medicines in the context of their efficacies. Initially, we gathered 79 binary similarity and dissimilarity equations. Some identical equations were eliminated in the preliminary step. Subsequently, the capability of

binary measures to separate herbal medicine pairs into match and mismatch efficacy groups was assessed by using the ROC analysis.

### 3.2. Datasets

We used the same Jamu dataset, mentioned in Chapter 2. We excluded from our analysis efficacy groups that only consist of very a few Jamu formulas, i.e. ear, nose, and throat disease (E4) and disease of the eye (E5). Therefore, there are 14 disease classes used in this Jamu study, of which 12 classes are from NCBI (National Center for Biotechnology Information 1998). The list of disease classes are as follows: blood and lymph diseases (E1), cancers (E2), the digestive system (E3), female-specific diseases (E6), the heart and blood vessels (E8), diseases of the immune system (E9), male-specific diseases (E10), muscle and bone (E11), the nervous system (E13), nutritional and metabolic diseases (E14), respiratory diseases (E15), skin and connective tissue (E16), the urinary system (E17), and mental and behavioral disorders (E18). Thus, Jamu vs. plant relations were then organized as a  $3131 \times 465$  matrix. Corresponding to 3,131 Jamu formulas, there can be  $(3,131 \times 3,130) / 2 = 4,900,015$  Jamu pairs.

For the purpose of comparison, we created four random matrices as the same size as Jamu-plant relations by randomly inserting 1s and 0s. In three of the random datasets, the numbers of 1s are 1%, 5%, and 10% of 465 plants (called as random 1%, random 5%, and random 10%). In the case of the other dataset, we randomly inserted the equal number of 1s in every row as it is in the original Jamu formulas (called as random Jamu). We also applied our proposed method to Kampo dataset (Okada et al. 2016). This dataset is presented as a two-dimensional binary matrix with rows and columns representing Kampo formulas and crude drug ingredients, respectively. Kampo dataset is composed of 274 Kampo formulas, and each formula consists of 3 to 19 crude drugs, with average 8.923, standard deviation 3.885, and the set union of all formulas consists



of 227 crude drugs. Then, each Kampo formula is classified into deficiency or excess class, according to Kampo-specific diagnosis of patient's constitution.

### 3.3. Methods

The proposed method leads to the selection of a suitable equation such that when two herbal medicine formulas belong to the same efficacy group, their ingredient similarity measured by the equation becomes higher in the global context of a large set of formulas. Fig. 3.1 illustrates data representation and also the procedure of our experiment. The binary similarity (S) and dissimilarity (D) measure between herbal medicine pair is expressed by the Operational Taxonomic Units (OTUs as shown in Fig. 3.1a) (Clifford & Stephenson 1975; Warrens 2008). Concretely, let two Jamu formulas be described by two-row vectors  $J_i$  and  $J_{i'}$ , each comprised of  $M$  variables with value 1 (presence) or 0 (absence). The four quantities  $a$ ,  $b$ ,  $c$ ,  $d$  in the OTUs table are defined as follows:  $a$  is the number of features where the values for both  $j_i$  and  $j_{i'}$  are 1 (positive matches),  $b$  and  $c$  are the number of features where the value for  $j_i$  is 0 and  $j_{i'}$  is 1 and vice versa, respectively (absence mismatches), and  $d$  is the number of features where the values for both  $j_i$  and  $j_{i'}$  are 0 (negative matches). The sum of  $a$  and  $d$  represents the total number of matches between  $j_i$  and  $j_{i'}$ , the sum of  $b$  and  $c$  represents the total number of mismatches between  $j_i$  and  $j_{i'}$ . The total sum of the quantities in the OTUs table  $a+b+c+d$  is equal to  $M$ .

**a. Format of the input data representing Jamu-plant relations and the OTUs expression of a Jamu pair**

$J_i$	$P_j$						$J_i$		
	$P_1$	$P_2$	$P_3$	$P_4$	...	$P_M$	1 (presence)	0 (absence)	Sum
$J_1$	0	0	1	0	...	0	$a$	$b$	$a+b$
$J_2$	1	0	0	0	...	0			
$J_3$	0	1	0	1	...	0	$c$	$d$	$c+d$
...	...	...	...	...	...	...			
$J_N$	0	1	0	0	...	1	$a+c$	$b+d$	$M=a+b+c+d$



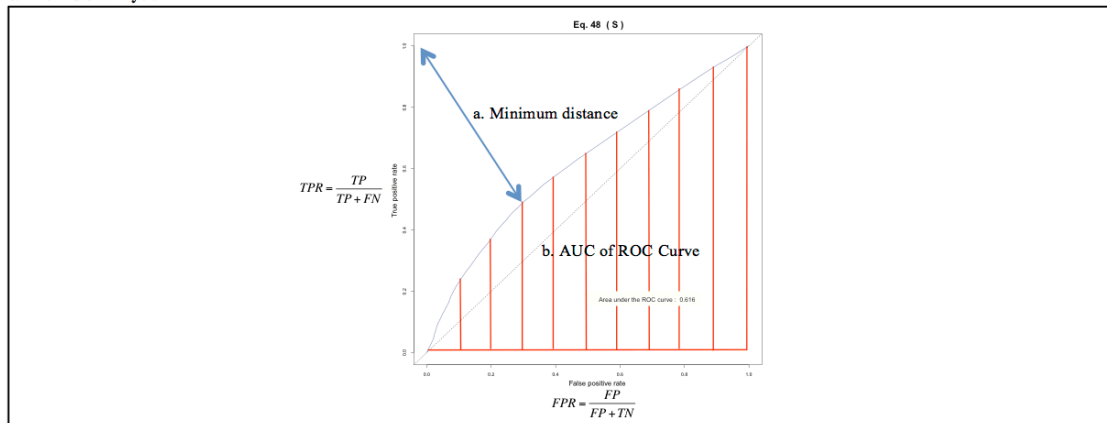
**b. Reducing the candidate equations**

Jamu Pairs		Efficacy		Efficacy's Match/Mismatch*		Binary Similarity/Dissimilarity Coefficients			
$J_i$	$J_{i'}$	$EJ_i$	$EJ_{i'}$	Match/Mismatch*	Eq. 1	...	Eq. $k$	...	Eq. 79
$J_1$	$J_2$	$EJ_1$	$EJ_2$	1/0	$s_{12}(1)$	...	$s_{12}(k)$	...	$s_{12}(79)$
...	...	...	...	...	...	...	...	...	...
$J_1$	$J_N$	$EJ_1$	$EJ_N$	1/0	$s_{1N}(1)$	...	$s_{1N}(k)$	...	$s_{1N}(79)$
$J_2$	$J_3$	$EJ_2$	$EJ_3$	1/0	$s_{23}(1)$	...	$s_{23}(k)$	...	$s_{23}(79)$
...	...	...	...	...	...	...	...	...	...
$J_2$	$J_N$	$EJ_2$	$EJ_N$	1/0	$s_{2N}(1)$	...	$s_{2N}(k)$	...	$s_{2N}(79)$
...	...	...	...	...	...	...	...	...	...
$J_i$	$J_{i'}$	$EJ_i$	$EJ_{i'}$	1/0	$s_{ii'}(1)$	...	$s_{ii'}(k)$	...	$s_{ii'}(79)$
...	...	...	...	...	...	...	...	...	...
$J_{N-1}$	$J_N$	$EJ_{N-1}$	$EJ_N$	1/0	$s_{N-1N}(1)$	...	$s_{N-1N}(k)$	...	$s_{N-1N}(79)$

\* '1' if the efficacy of Jamu pair is the same and '0' if the efficacy of Jamu pair is different



**c. The ROC analysis**



**Figure 3.1.** An illustration of the experimental flow. This figure also illustrates a representation of plant, herbal medicine formulas and efficacy relations as a two-dimensional matrix.

We collected equations to measure similarity or dissimilarity between binary vectors from published literatures (Consonni & Todeschini 2012; Batagelj & Bren 1995; Zhang & Sargur N. Srihari 2003; Zhang & Sargur N Srihari 2003; Choi et al. 2010; Todeschini et al. 2012; da Silva Meyer et al. 2004; Warrens 2008; Jackson et al. 1989; Dalirsefat et al. 2009; Jaccard 1912; Dice 1945; Hubalek 1982; Cheetham et al. 1969; Cha et al. 2009; Cha et al. 2005; Lourenco et al. 2004; Ojurongbe 2012; Johnson 1967; Michael 1920; Stiles 1961; Nei & Li 1979; Holliday et al. 2002; Boyce & Ellison 2001; Faith 1983; Gower & Legendre 1986; Chang et al. 2003; Lance & Williams 1966; Avcibaş et al. 2005; Baroni-urbani & Buser 1976), listed as Eqs. 1-79 in Table 3.1. The binary similarity and dissimilarity equations were represented by four quantities, i.e.  $a$ ,  $b$ ,  $c$  and  $d$ . We also implemented these 79 equations as an R package, called `bmeasures`. The `bmeasures` package is available on Github and can be installed by invoking these commands: `install.packages("devtools"), library("devtools"), install_github("shwijaya/bmeasures"), library("bmeasures")`. The installation of `bmeasures` package was tested on R release 3.2.4 and the `devtools` package ver. 1.11.0. Initially, we measure the similarity and dissimilarity coefficients between herbal medicine pairs by using 79 equations. Then, the resulted similarity/dissimilarity coefficients are used for further analysis. Our experimental procedure can be divided into two major steps, which we discuss in the following segments.

### 3.3.1. Reducing the candidate equations

The binary similarity and dissimilarity equations were evaluated to eliminate duplications. When two or more equations can be transformed into the same form by algebraic manipulations, only one of them is kept for further analysis. We also removed equations from our analysis that produce infinite/NaN values or indeterminate forms while applying to measure similarity and

dissimilarity using all datasets.

Hierarchical clustering of the equations was done with an aim to further narrow down the number of candidate equations and to evaluate the closeness between equations. After we obtained the similarity/dissimilarity coefficients between herbal medicine pairs for each remaining equation, we clustered those equations based on its similarity/dissimilarity coefficients using Agglomerative hierarchical clustering with Centroid linkage (Fig. 3.1b) (Ojurongbe 2012; Frigui & Krishnapuram 1997; Cimiano et al. 2004; Bolshakova & Azuaje 2003). The Euclidean distance (Eq. 80) was used to measure the distance between two equations,  $k$  and  $l$ , that is:

$$d_{k,l} = \sqrt{\sum_{m=1}^{N-1} \sum_{n=m+1}^N (s_{mn}(k) - s_{mn}(l))^2} \quad (80)$$

where  $s_{mn}(k)$  and  $s_{mn}(l)$  are the similarity/dissimilarity values between corresponding herbal medicine pair using equations  $k$  and  $l$  respectively,  $N$  is the total number of herbal medicine formulas, and  $d_{k,l}$  is the distance between equation  $k$  and  $l$ . The cluster centroid is the average values of the variables for the observations (in the present case equations) in that cluster. Let  $\bar{X}_G, \bar{X}_H$  denote group averages for clusters  $G$  and  $H$ . Then, the distance between cluster centroids is calculated using Eq. 81.

$$d_{centroid}(G, H) = \|\bar{X}_G - \bar{X}_H\|_2 \quad (81)$$

where  $\bar{X}_G$  is the centroid of  $G$  by arithmetic mean  $\bar{X}_G = \frac{1}{n_G} \sum_{i=1}^{n_G} X_{Gi}$  (Legendre & Legendre 1998; Hillenmeyer 2005; Bien & Tibshirani 2011). We implemented the clustering process using `hclust` function in R. At each step, the cluster centroid was calculated to represent a group of equations in the clusters. Furthermore, two equations or clusters are merged for which the distance between the centroids is the minimum until all equations are merged into one cluster.

We performed the hierarchical clustering process twice, first to reduce the candidate equations for which the distance between equations measured by Eq. 80

is zero or nearly zero and secondly to evaluate the combined characteristic of a group of equations. Mean centering and unit variance scaling were applied to the similarity/dissimilarity coefficients before the clustering process.

**Table 3.1.** *List of 79 binary similarity and dissimilarity measures.  $n$  is a constant ( $n = M = a + b + c + d$ ). According to Note column: \* means algebraically redundant, \*\* means produce infinite/NaN coefficients or indeterminate forms, \*\*\* means grouped in the same cluster with zero or nearly to zero distance.*

Eq. IDs	Equations	References	Note
1	$S_{Jaccard} = \frac{a}{a+b+c}$	(Consonni & Todeschini 2012; Zhang & Sargur N. Srihari 2003; Zhang & Sargur N Srihari 2003; Choi et al. 2010; da Silva Meyer et al. 2004; Warrens 2008; Jackson et al. 1989; Dalirsefat et al. 2009; Todeschini et al. 2012; Jaccard 1912; Hubalek 1982; Cheetham et al. 1969; Cha et al. 2009; Cha et al. 2005; Lourenco et al. 2004; Ojurongbe 2012; Holliday et al. 2002)	
2	$S_{Dice-2} = \frac{a}{2a+b+c}$	(Zhang & Sargur N. Srihari 2003; Zhang & Sargur N Srihari 2003; Cha et al. 2009; Cha et al. 2005)	
3	$S_{Dice-1/Czekanowski} = \frac{2a}{2a+b+c}$	(Choi et al. 2010; da Silva Meyer et al. 2004; Warrens 2008; Jackson et al. 1989; Dalirsefat et al. 2009; Todeschini et al. 2012; Dice 1945; Hubalek 1982; Cheetham et al. 1969; Cha et al. 2009; Lourenco et al. 2004; Ojurongbe 2012; Holliday et al. 2002; Batagelj & Bren 1995)	***
4	$S_{3W-Jaccard} = \frac{3a}{3a+b+c}$	(Choi et al. 2010; Todeschini et al. 2012; Jaccard 1912; Cha et al. 2009)	

Eq. IDs	Equations	References	Note
5	$S_{Nei\&Li} = \frac{2a}{(a+b) + (a+c)}$	(Choi et al. 2010; Warrens 2008; Nei & Li 1979)	*
6	$S_{Sokal\&Sneath-1} = \frac{a}{a+2b+2c}$	(Consonni & Todeschini 2012; Choi et al. 2010; Todeschini et al. 2012; Hubalek 1982; Cha et al. 2009; Warrens 2008; Holliday et al. 2002)	
7	$S_{Sokal\&Michener} = \frac{a+d}{a+b+c+d}$	(Consonni & Todeschini 2012; Zhang & Sargur N. Srihari 2003; Zhang & Sargur N Srihari 2003; Choi et al. 2010; da Silva Meyer et al. 2004; Jackson et al. 1989; Dalirsefat et al. 2009; Todeschini et al. 2012; Hubalek 1982; Cheetham et al. 1969; Cha et al. 2005; Lourenco et al. 2004; Warrens 2008; Ojurongbe 2012; Batagelj & Bren 1995)	
8	$S_{Sokal\&Sneath-2} = \frac{2(a+d)}{2a+b+c+2d}$	(Consonni & Todeschini 2012; Choi et al. 2010; Todeschini et al. 2012; Hubalek 1982; Lourenco et al. 2004; Warrens 2008; Ojurongbe 2012; Holliday et al. 2002)	
9	$S_{Roger\&Tanimoto} = \frac{a+d}{a+2(b+c)+d}$	(Zhang & Sargur N. Srihari 2003; Zhang & Sargur N Srihari 2003; Choi et al. 2010; da Silva Meyer et al. 2004; Jackson et al. 1989; Todeschini et al. 2012; Hubalek 1982; Cheetham et al. 1969; Cha et al. 2005; Lourenco et al. 2004; Warrens 2008; Ojurongbe 2012; Holliday et al. 2002; Boyce & Ellison 2001)	
10	$S_{Faith} = \frac{a+0.5d}{a+b+c+d}$	(Choi et al. 2010; Todeschini et al. 2012; Boyce & Ellison 2001; Faith	

Eq. IDs	Equations	References	Note
		1983)	
11	$S_{Gower\&Legendre} = \frac{a+d}{a+0.5(b+c)+d}$	(Choi et al. 2010; Todeschini et al. 2012; Gower & Legendre 1986)	*
12	$S_{Intersection} = a$	(Choi et al. 2010; Cha et al. 2009)	
13	$S_{Innerproduct} = a+d$	(Choi et al. 2010)	***
14	$S_{Russell\&Rao} = \frac{a}{a+b+c+d}$	(Consonni & Todeschini 2012; Zhang & Sargur N. Srihari 2003; Zhang & Sargur N Srihari 2003; Choi et al. 2010; da Silva Meyer et al. 2004; Jackson et al. 1989; Todeschini et al. 2012; Hubalek 1982; Cha et al. 2009; Cha et al. 2005; Lourenco et al. 2004; Warrens 2008; Ojuronbe 2012; Holliday et al. 2002; Boyce & Ellison 2001; Batagelj & Bren 1995)	***
15	$D_{Hamming} = b+c$	(Choi et al. 2010; Cha et al. 2005; Chang et al. 2003)	
16	$D_{Euclid} = \sqrt{b+c}$	(Choi et al. 2010)	
17	$D_{Squared-euclid} = \sqrt{(b+c)^2}$	(Choi et al. 2010; Lance & Williams 1966)	*
18	$D_{Canberra} = (b+c)^{\frac{2}{3}}$	(Choi et al. 2010)	*
19	$D_{Manhattan} = b+c$	(Choi et al. 2010)	*
20	$D_{Mean-Manhattan} = \frac{b+c}{a+b+c+d}$	(Choi et al. 2010; Holliday et al. 2002)	***
21	$D_{Cityblock} = b+c$	(Choi et al. 2010)	*
22	$D_{Minkowski} = (b+c)^{\frac{1}{2}}$	(Choi et al. 2010)	*
23	$D_{Vari} = \frac{b+c}{4(a+b+c+d)}$	(Choi et al. 2010; Avcibaş et al. 2005)	***
24	$D_{SizeDifference} = \frac{(b+c)^2}{(a+b+c+d)^2}$	(Choi et al. 2010)	
25	$D_{ShapeDifference} = \frac{n(b+c)-(b-c)^2}{(a+b+c+d)^2}$	(Choi et al. 2010)	
26	$D_{PatternDifference} = \frac{4bc}{(a+b+c+d)^2}$	(Choi et al. 2010)	
27	$D_{Lance\&Williams} = \frac{b+c}{2a+b+c}$	(Choi et al. 2010; Avcibaş et al. 2005)	
28	$D_{Bray\&Curtis} = \frac{b+c}{2a+b+c}$	(Choi et al. 2010)	*

Eq. IDs	Equations	References	Note
29	$D_{Hellinger} = 2 \sqrt{\left(1 - \frac{a}{\sqrt{(a+b)(a+c)}}\right)}$	(Choi et al. 2010)	
30	$D_{Chord} = \sqrt{2 \left(1 - \frac{a}{\sqrt{(a+b)(a+c)}}\right)}$	(Choi et al. 2010)	***
31	$S_{Cosine} = \frac{a}{\sqrt{(a+b)(a+c)}}$	(Todeschini et al. 2012; Holliday et al. 2002)	
32	$S_{Gilbert\&Wells} = \log a - \log n - \log\left(\frac{a+b}{n}\right) - \log\left(\frac{a+c}{n}\right)$	(Choi et al. 2010; Hubalek 1982)	**
33	$S_{Ochiai-1} = \frac{a}{\sqrt{(a+b)(a+c)}}$	(Choi et al. 2010; da Silva Meyer et al. 2004; Jackson et al. 1989; Todeschini et al. 2012; Lourenco et al. 2004; Warrens 2008; Holliday et al. 2002; Boyce & Ellison 2001)	*
34	$S_{Forbes-1} = \frac{na}{(a+b)(a+c)}$	(Choi et al. 2010; Todeschini et al. 2012; Hubalek 1982; Cha et al. 2009; Warrens 2008; Holliday et al. 2002)	
35	$S_{Fossum} = \frac{n(a-0.5)^2}{(a+b)(a+c)}$	(Choi et al. 2010; Todeschini et al. 2012; Holliday et al. 2002)	
36	$S_{Sorgenfrei} = \frac{a^2}{(a+b)(a+c)}$	(Choi et al. 2010; Hubalek 1982; Todeschini et al. 2012; Warrens 2008)	
37	$S_{Mountford} = \frac{a}{0.5(ab+ac)+bc}$	(Choi et al. 2010; Hubalek 1982; Todeschini et al. 2012; Warrens 2008)	**
38	$S_{Otsuka} = \frac{a}{((a+b)(a+c))^{0.5}}$	(Choi et al. 2010; Cheetham et al. 1969)	*
39	$S_{McConnaughey} = \frac{a^2 - bc}{(a+b)(a+c)}$	(Choi et al. 2010; Hubalek 1982; Warrens 2008; Holliday et al. 2002)	
40	$S_{Rarwid} = \frac{na - (a+b)(a+c)}{na + (a+b)(a+c)}$	(Choi et al. 2010; Hubalek 1982)	
41	$S_{Kulczynski-2} = \frac{\frac{a}{2}(2a+b+c)}{(a+b)(a+c)}$	(Choi et al. 2010; Hubalek 1982; Cheetham et al. 1969; Lourenco et al. 2004; Warrens 2008; Holliday et al. 2002)	***
42	$S_{Driver\&Kroeber} = \frac{a}{2} \left(\frac{1}{a+b} + \frac{1}{a+c}\right)$	(Choi et al. 2010; Hubalek 1982; Warrens 2008)	***



Eq. IDs	Equations	References	Note
43	$S_{Johnson} = \frac{a}{a+b} + \frac{a}{a+c}$	(Choi et al. 2010; Hubalek 1982; Todeschini et al. 2012; Warrens 2008; Johnson 1967)	***
44	$S_{Dennis} = \frac{ad - bc}{\sqrt{n(a+b)(a+c)}}$	(Choi et al. 2010; Todeschini et al. 2012; Holliday et al. 2002)	
45	$S_{Simpson} = \frac{a}{\min(a+b, a+c)}$	(Choi et al. 2010; Todeschini et al. 2012; Holliday et al. 2002; Hubalek 1982; Warrens 2008)	
46	$S_{Braun\&Banquet} = \frac{a}{\max(a+b, a+c)}$	(Choi et al. 2010; Hubalek 1982; Todeschini et al. 2012; Cha et al. 2009; Warrens 2008)	
47	$S_{Fager\&McGowan} = \frac{a}{\sqrt{(a+b)(a+c)}} - \frac{\max(a+b, a+c)}{2}$	(Choi et al. 2010; Hubalek 1982)	
48	$S_{Forbes-2} = \frac{na - (a+b)(a+c)}{n \min(a+b, a+c) - (a+b)(a+c)}$	(Choi et al. 2010; Hubalek 1982)	
49	$S_{Sokal\&Sneath-4} = \frac{\frac{a}{(a+b)} + \frac{a}{(a+c)} + \frac{d}{(b+d)} + \frac{d}{(c+d)}}{4}$	(Consonni & Todeschini 2012; Todeschini et al. 2012; Warrens 2008; Hubalek 1982)	
50	$S_{Gower} = \frac{a+d}{\sqrt{(a+b)(a+c)(b+d)(c+d)}}$	(Choi et al. 2010)	
51	$S_{Pearson-1} = \chi^2 = \frac{n(ad - bc)^2}{(a+b)(a+c)(c+d)(b+d)}$	(Choi et al. 2010; Hubalek 1982; Warrens 2008)	
52	$S_{Pearson-2} = \left( \frac{\chi^2}{n + \chi^2} \right)^{\frac{1}{2}}$	(Choi et al. 2010; Hubalek 1982)	
53	$S_{Pearson-3} = \left( \frac{\rho}{n + \rho} \right)^{\frac{1}{2}}$	(Choi et al. 2010)	**
	where $\rho = \frac{ad - bc}{\sqrt{(a+b)(a+c)(b+d)(c+d)}}$		
54	$S_{Pearson\&Heron-1} = \frac{ad - bc}{\sqrt{(a+b)(a+c)(b+d)(c+d)}}$	(Choi et al. 2010; Zhang & Sargur N. Srihari 2003; Zhang & Sargur N Srihari 2003; Todeschini et al. 2012; Hubalek 1982; Warrens 2008)	

Eq. IDs	Equations	References	Note
55	$S_{\text{Pearson\&Heron-2}} = \cos\left(\frac{\pi\sqrt{bc}}{\sqrt{ad} + \sqrt{bc}}\right)$	(Choi et al. 2010; Hubalek 1982)	
56	$S_{\text{Sokat\&Sneath-3}} = \frac{a+d}{b+c}$	(Choi et al. 2010; Hubalek 1982; Warrens 2008; Holliday et al. 2002)	**
57	$S_{\text{Sokat\&Sneath-5}} = \frac{ad}{(a+b)(a+c)(b+d)(c+d)^{0.5}}$	(Choi et al. 2010; Hubalek 1982; Consonni & Todeschini 2012; Todeschini et al. 2012; Warrens 2008)	
58	$S_{\text{Cote}} = \frac{\sqrt{2}(ad-bc)}{\sqrt{(ad-bc)^2 - (a+b)(a+c)(b+d)(c+d)}}$	(Choi et al. 2010; Hubalek 1982)	**
59	$S_{\text{Stiles}} = \log_{10} \frac{n( ad-bc  - \frac{n}{2})^2}{(a+b)(a+c)(b+d)(c+d)}$	(Choi et al. 2010; Warrens 2008; Stiles 1961; Holliday et al. 2002)	
60	$S_{\text{Ochiai-2}} = \frac{ad}{\sqrt{(a+b)(a+c)(b+d)(c+d)}}$	(Choi et al. 2010; da Silva Meyer et al. 2004; Lourenco et al. 2004)	*
61	$S_{\text{Yuleq}} = \frac{ad-bc}{ad+bc}$	(Zhang & Sargur N. Srihari 2003; Zhang & Sargur N Srihari 2003; Choi et al. 2010; Jackson et al. 1989; Todeschini et al. 2012; Hubalek 1982; Cheetham et al. 1969; Cha et al. 2005; Warrens 2008; Holliday et al. 2002)	
62	$D_{\text{Yuleq}} = \frac{2bc}{ad+bc}$	(Choi et al. 2010)	
63	$S_{\text{Yulew}} = \frac{\sqrt{ad} - \sqrt{bc}}{\sqrt{ad} + \sqrt{bc}}$	(Choi et al. 2010; Todeschini et al. 2012; Hubalek 1982; Warrens 2008; Batagelj & Bren 1995)	
64	$S_{\text{Kulczynski-1}} = \frac{a}{b+c}$	(Zhang & Sargur N. Srihari 2003; Zhang & Sargur N Srihari 2003; Choi et al. 2010; Hubalek 1982; Cheetham et al. 1969; Cha et al. 2005; Lourenco et al. 2004; Ojurongbe 2012; Holliday et al. 2002; Cha et al. 2009; Batagelj & Bren 1995)	**
65	$S_{\text{Tanimoto}} = \frac{a}{(a+b) + (a+c) - a}$	(Choi et al. 2010; Todeschini et al. 2012; Consonni & Todeschini 2012;	*

Eq. IDs	Equations	References	Note
		Holliday et al. 2002)	
66	$S_{Dispersion} = \frac{ad - bc}{(a + b + c + d)^2}$	(Choi et al. 2010; Todeschini et al. 2012)	
67	$S_{Hamann} = \frac{(a + d) - (b + c)}{a + b + c + d}$	(Choi et al. 2010; Hubalek 1982; Cheetham et al. 1969; Lourenco et al. 2004; Warrens 2008; Ojurongbe 2012; Holliday et al. 2002; Batagelj & Bren 1995)	***
68	$S_{Michael} = \frac{4(ad - bc)}{(a + d)^2 + (b + c)^2}$	(Choi et al. 2010; Todeschini et al. 2012; Hubalek 1982; Warrens 2008; Michael 1920)	
69	$S_{Goodman\&Kruskal} = \frac{\sigma - \sigma'}{2n - \sigma'}$ where $\sigma = \max(a, b) + \max(c, d) + \max(a, c) + \max(b, d)$ $\sigma' = \max(a + c, b + d) + \max(a + b, c + d)$	(Choi et al. 2010)	**
70	$S_{Anderberg} = \frac{\sigma - \sigma'}{2n}$	(Choi et al. 2010)	**
71	$S_{Baroni-Urbani\&Buser-1} = \frac{\sqrt{ad} + a}{\sqrt{ad} + a + b + c}$	(Choi et al. 2010; Todeschini et al. 2012; Hubalek 1982; Warrens 2008; Holliday et al. 2002; Boyce & Ellison 2001; Baroni-urbani & Buser 1976)	
72	$S_{Baroni-Urbani\&Buser-2} = \frac{\sqrt{ad} + a - (b + c)}{\sqrt{ad} + a + b + c}$	(Choi et al. 2010; Todeschini et al. 2012; Hubalek 1982; Warrens 2008; Baroni-urbani & Buser 1976)	***
73	$S_{Peirce} = \frac{ab + bc}{ab + 2bc + cd}$	(Choi et al. 2010; Hubalek 1982)	**
74	$S_{Eyrraud} = \frac{n^2(na - (a + b)(a + c))}{(a + b)(a + c)(b + d)(c + d)}$	(Choi et al. 2010)	
75	$S_{Tarantula} = \frac{a}{\frac{(a + b)}{c}} = \frac{a(c + d)}{c(a + b)}$	(Choi et al. 2010)	**
76	$S_{Ample} = \left  \frac{\frac{a}{(a + b)}}{\frac{c}{(c + d)}} \right  = \left  \frac{a(c + d)}{c(a + b)} \right $	(Choi et al. 2010)	**
77	$S_{Derived\_Russell-Rao} = \frac{\log(1 + a)}{\log(1 + n)}$	(Consonni & Todeschini 2012; Todeschini et al. 2012)	
78	$S_{Derived\_Jaccard} = \frac{\log(1 + a)}{\log(1 + a + b + c)}$	(Consonni & Todeschini 2012; Todeschini et al. 2012)	

Eq. IDs	Equations	References	Note
79	$S_{Var\_of\_Correlation} = \frac{\log(1+ad) - \log(1+bc)}{\log(1+n^2/4)}$	(Consonni & Todeschini 2012; Todeschini et al. 2012)	

### 3.3.2. An ROC analysis and Cohen's Kappa

The effectiveness of similarity/dissimilarity measuring the capability of the selected equations was evaluated by means of the ROC curve (Fig. 3.1c) (Sonego et al. 2008; Fawcett 2006). For ROC analysis, we divided all herbal medicine pairs into match and mismatch efficacy classes and used the corresponding distributions with respect to similarity scores to calculate *FPRs* and *TPRs*. The ROC curve was created by selecting a series of thresholds to generate *FPR* and *TPR*. *FPR* is the proportion of false positive predictions out of all the false data, and *TPR* is the proportion of true positive predictions out of all the true data, defined by Eq. 82 (Sonego et al. 2008; Li et al. 2008; Fawcett 2006):

$$FPR = FP/(FP + TN) \quad TPR = TP/(TP + FN) \quad (82)$$

where true positive (*TP*) is the number of herbal medicine pairs correctly classified as positive, true negative (*TN*) is the number of pairs correctly classified as negative, false positive (*FP*) is the number of pairs incorrectly classified as positive, and false negative (*FN*) is the number of pairs incorrectly classified as negative. We defined and compared the performance of good equations by using the minimum distance of the ROC curve to the theoretically optimum point and by using the Area Under the ROC curve (AUC) analysis (Gorunescu 2011). The minimum distance between the ROC curve and the optimum point was measured as the Euclidean distance. The minimum distance can also be computed by *TP*, *TN*, *FP*, and *FN* values corresponding to selected similarity thresholds *i* using the following formulation:

$$Min. dist = \min_{i \in thresholds} \sqrt{(FP_i/(TN_i + FP_i))^2 + (FN_i/(TP_i + FN_i))^2} \quad (83)$$

In order to give a wider perspective of equation performance, we also provided Cohen’s Kappa score. Cohen’s Kappa is defined as (Ben-David 2008):

$$K = \frac{P_0 - P_c}{1 - P_c} \quad (84)$$

where  $P_0$  is the total agreement probability (or the accuracy) and  $P_c$  is the agreement probability which is due to chance.

$$P_c = \sum_{i=1}^I P(x_i)P(x_i) \quad (85)$$

where  $I$  is the number of class values, and  $P(x_i)P(x_i)$  are the columns and rows marginal probabilities, respectively.

### 3.4. Results and discussion

#### 3.4.1. Preliminary verification of the equations

In the preliminary step, we removed 12 equations denoted by ‘\*’ in Table 3.1 because each of them can be recognized as identical to one or more other equations by only algebraic manipulations such as linear transformation. From the seven groups of redundant equations shown in Table 3.2, we included  $S_{\text{Jaccard}}$ ,  $S_{\text{Dice-1/Czekanowski}}$ ,  $S_{\text{Sokal\&Sneath-2}}$ ,  $D_{\text{Hamming}}$ ,  $D_{\text{Lance\&Williams}}$ ,  $S_{\text{Cosine}}$  and  $S_{\text{Sokal\&Sneath-5}}$  in our analysis and therefore, we were left with 67 equations at this stage. Next, we clustered the 67 equations to reduce the number of equations using Jamu and Kampo datasets. During the clustering process, we eliminated 11 equations indicated by ‘\*\*’ in Table 3.1 that produced infinite/NaN values or indeterminate forms while applied to all datasets. Such conditions can be reached when the denominator of an equation becomes equal to 0, i.e. the values of  $b$  and  $c$  in the Mountford and Peirce similarities (Eq. 37 and Eq. 73) are 0 if two formulas use exactly the same ingredients.

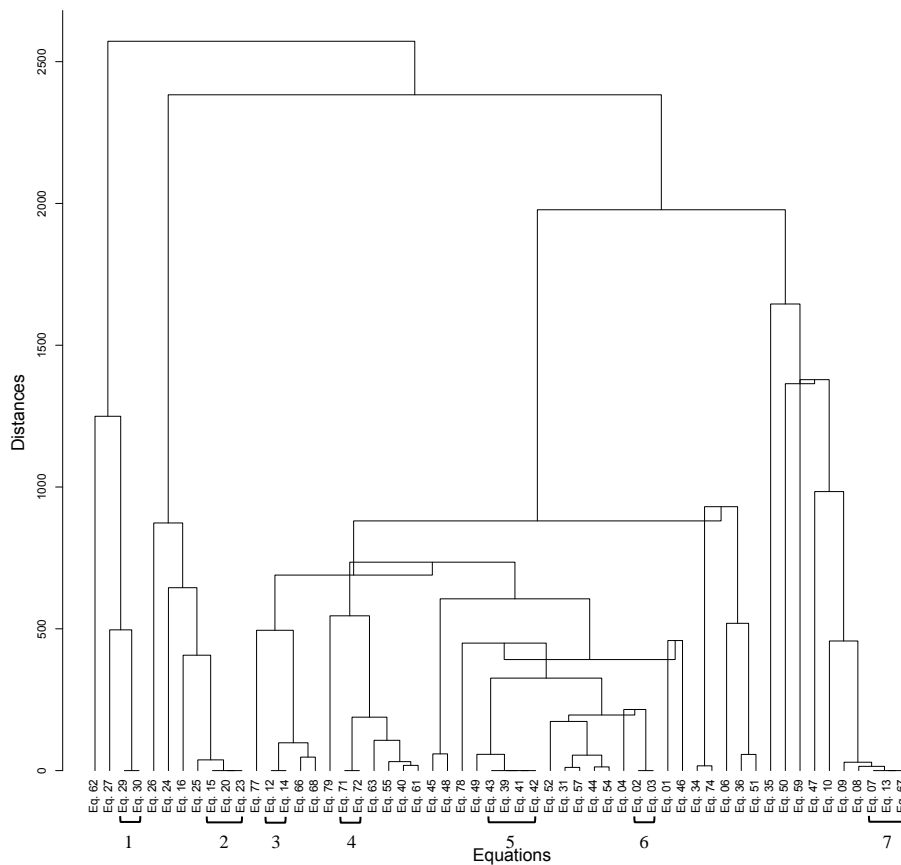
**Table 3.2.** *Groups of identical equations.*

Groups	Eliminated equations	Selected equations
1	$S_{\text{Nei\&Li}}(\text{Eq. 5}) = \frac{2a}{(a+b)+(a+c)}$	$S_{\text{Dice-1/Czekanowski}}(\text{Eq. 3}) = \frac{2a}{2a+b+c}$
2	$S_{\text{Gower\&Legendre}}(\text{Eq. 11}) = \frac{a+d}{a+0.5(b+c)+d}$	$S_{\text{Sokal\&Sneath-2}}(\text{Eq. 8}) = \frac{2(a+d)}{2a+b+c+2d}$

Groups	Eliminated equations	Selected equations
3	$D_{\text{Squared-euclid}}$ (Eq. 17) = $\sqrt{(b+c)^2}$ $D_{\text{Canberra}}$ (Eq. 18) = $(b+c)^{\frac{2}{3}}$ $D_{\text{Manhattan}}$ (Eq. 19) = $b+c$ $D_{\text{Cityblock}}$ (Eq. 21) = $b+c$ $D_{\text{Minkowski}}$ (Eq. 22) = $(b+c)^{\frac{1}{2}}$	$D_{\text{Hamming}}$ (Eq. 15) = $b+c$
4	$D_{\text{Bray\&Curtis}}$ (Eq. 28) = $\frac{b+c}{2a+b+c}$	$D_{\text{Lance\&Williams}}$ (Eq. 27) = $\frac{b+c}{2a+b+c}$
5	$S_{\text{Ochiai-1}}$ (Eq. 33) = $\frac{a}{\sqrt{(a+b)(a+c)}}$ $S_{\text{Otsuka}}$ (Eq. 38) = $\frac{a}{((a+b)(a+c))^{0.5}}$	$S_{\text{Cosine}}$ (Eq. 31) = $\frac{a}{\sqrt{(a+b)(a+c)}}$
6	$S_{\text{Ochiai-2}}$ (Eq. 60) = $\frac{ad}{\sqrt{(a+b)(a+c)(b+d)(c+d)}}$	$S_{\text{Sokal\&Sneath-5}}$ (Eq. 57) = $\frac{ad}{(a+b)(a+c)(b+d)(c+d)^{0.5}}$
7	$S_{\text{Tanimoto}}$ (Eq. 65) = $\frac{a}{(a+b)+(a+c)-a}$	$S_{\text{Jaccard}}$ (Eq. 1) = $\frac{a}{a+b+c}$

The clustering of 56 equations in the context of Jamu data is shown in Fig. 3.2. The distances among equations belonging to individual clusters indicated as 1 to 7 in Fig. 3.2 are equal or nearly equal to 0. In other words, those equations have similar characteristics when generating binary similarity/dissimilarity coefficients for Jamu data. By using the clustering result, we reduced 11 equations denoted by ‘\*\*\*’ in Table 3.1 because they were related to other equations in the same cluster e.g. we eliminated  $S_{\text{Baroni-Urbani\&Buser-2}}$  (Eq. 72) because it is similar to  $S_{\text{Baroni-Urbani\&Buser-1}}$  (Eq. 71). A careful observation of equations belonging to the same cluster in the group IDs 1 to 7 in Fig. 3.2 implies that one equation can be transformed to another just by adding or multiplying by constants (Table 3.3). For example, we can represent  $S_{\text{Baroni-Urbani\&Buser-2}}$  as  $[(2 \times S_{\text{Baroni-Urbani\&Buser-1}}) - 1]$ . The excluded equations based on the clustering process are as follows:  $S_{\text{Dice-1/Czekanowski}}$  (Eq. 3),  $S_{\text{Innerproduct}}$  (Eq. 13),  $S_{\text{Russell\&Rao}}$  (Eq. 14),  $D_{\text{Mean-Manhattan}}$  (Eq. 20),  $D_{\text{Vari}}$  (Eq. 23),  $D_{\text{Chord}}$  (Eq. 30),  $S_{\text{Kulczynski-2}}$  (Eq. 41),  $S_{\text{Driver\&Kroeber}}$  (Eq. 42),  $S_{\text{Johnson}}$  (Eq. 43),  $S_{\text{Hamann}}$  (Eq. 67), and  $S_{\text{Baroni-Urbani\&Buser-2}}$  (Eq. 72). In case of Kampo dataset, the clustering results also identified the same equations belong to the same cluster with zero or nearly to zero distance. Therefore, both datasets eliminated the same equations, indicated by ‘\*\*\*’ in

Table 3.1, and also obtained the same number of selected equations (45 binary similarity and dissimilarity measures) for further analysis. Hence, among the 79 binary similarity and dissimilarity measures used over the last century, there are only 45 unique equations that produce different coefficients by capturing different information. These binary measures satisfy the symmetry property (Carey et al. 2005), i.e. in case of such equations  $d(x, y) = d(y, x)$  or  $S(x, y) = S(y, x)$ .



**Figure 3.2.** Clustering of 56 binary similarity and dissimilarity measures in the context of Jamu data after removing algebraically redundant equations and equations that produce invalid coefficients. The distances between equations belonging to the same clusters are zero or nearly zero, and we select only one equation from each such cluster for the ROC analysis of the next step.

**Table 3.3.** Transformation of an equation into another by adding or multiplying by constants (Group IDs correspond to clusters in Fig. 3.2).

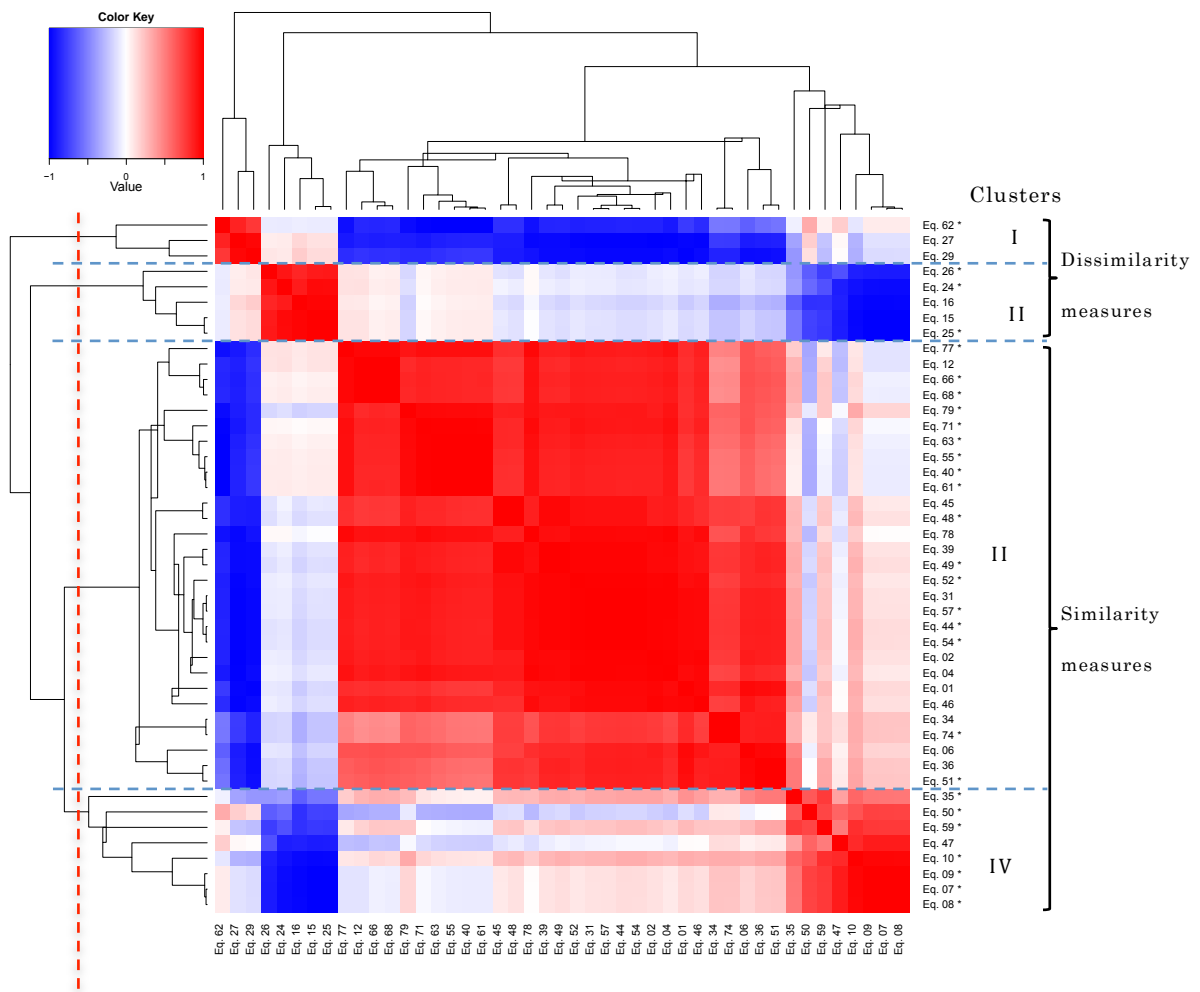
Group IDs	Eliminated equations	Selected equations *
1	$D_{Chord}(\text{Eq. 30}) = \sqrt{2 \left(1 - \frac{a}{\sqrt{(a+b)(a+c)}}\right)}$	$= \frac{1}{\sqrt{2}} 2 \sqrt{\left(1 - \frac{a}{\sqrt{(a+b)(a+c)}}\right)} = \frac{1}{\sqrt{2}} D_{Hellinger} \text{ (Eq. 29)}$
2	$D_{Mean-Manhattan}(\text{Eq. 20}) = \frac{b+c}{a+b+c+d}$	$= \frac{1}{M} (b+c) = \frac{1}{M} D_{Hamming} \text{ (Eq. 15)}$
	$D_{Vari}(\text{Eq. 23}) = \frac{b+c}{4(a+b+c+d)}$	$= \frac{1}{4M} (b+c) = \frac{1}{4M} D_{Hamming} \text{ (Eq. 15)}$
3	$S_{Russell\&Rao}(\text{Eq. 14}) = \frac{a}{a+b+c+d}$	$= \frac{1}{M} a = \frac{1}{M} S_{Intersection} \text{ (Eq. 12)}$
4	$S_{Baroni-Urbani\&Buser-2}(\text{Eq. 72}) = \frac{\sqrt{ad+a-(b+c)}}{\sqrt{ad+a+b+c}}$	$= 2 \frac{\sqrt{ad+a}}{\sqrt{ad+a+b+c}} - 1 = [2 \times S_{Baroni-Urbani\&Buser-1} \text{ (Eq. 71)}]$ $-1$
5	$S_{Kulczynski-2}(\text{Eq. 41}) = \frac{\frac{a}{2}(2a+b+c)}{(a+b)(a+c)}$	$= \frac{1}{2} \left( \frac{a}{a+b} + \frac{a}{a+c} \right) = \frac{1}{2} S_{Johnson} \text{ (Eq. 43)}$
	$S_{Driver\&Kroeber}(\text{Eq. 42}) = \frac{a}{2} \left( \frac{1}{a+b} + \frac{1}{a+c} \right)$	$= \frac{1}{2} \left( \frac{a}{a+b} + \frac{a}{a+c} \right) = \frac{1}{2} S_{Johnson} \text{ (Eq. 43)}$
	$S_{Johnson}(\text{Eq. 43}) = \frac{a}{a+b} + \frac{a}{a+c}$	$= 1 + \left( \frac{a^2-bc}{(a+b)(a+c)} \right) = 1 + S_{McConnaughey} \text{ (Eq. 39)}$
6	$S_{Dice-1/Czekanowski}(\text{Eq. 3}) = \frac{2a}{2a+b+c}$	$= 2 \frac{a}{2a+b+c} = 2 \times S_{Dice-2} \text{ (Eq. 2)}$
7	$S_{Innerproduct}(\text{Eq. 13}) = a + d$	$= M \frac{a+d}{a+b+c+d} = M \times S_{Sokal\&Michener} \text{ (Eq. 7)}$
	$S_{Hamann}(\text{Eq. 67}) = \frac{(a+d)-(b+c)}{a+b+c+d}$	$= 2 \left( \frac{a+d}{a+b+c+d} \right) - 1 = [2 \times S_{Sokal\&Michener} \text{ (Eq. 7)}] - 1$

\*  $M$  is a constant  $(a + b + c + d)$

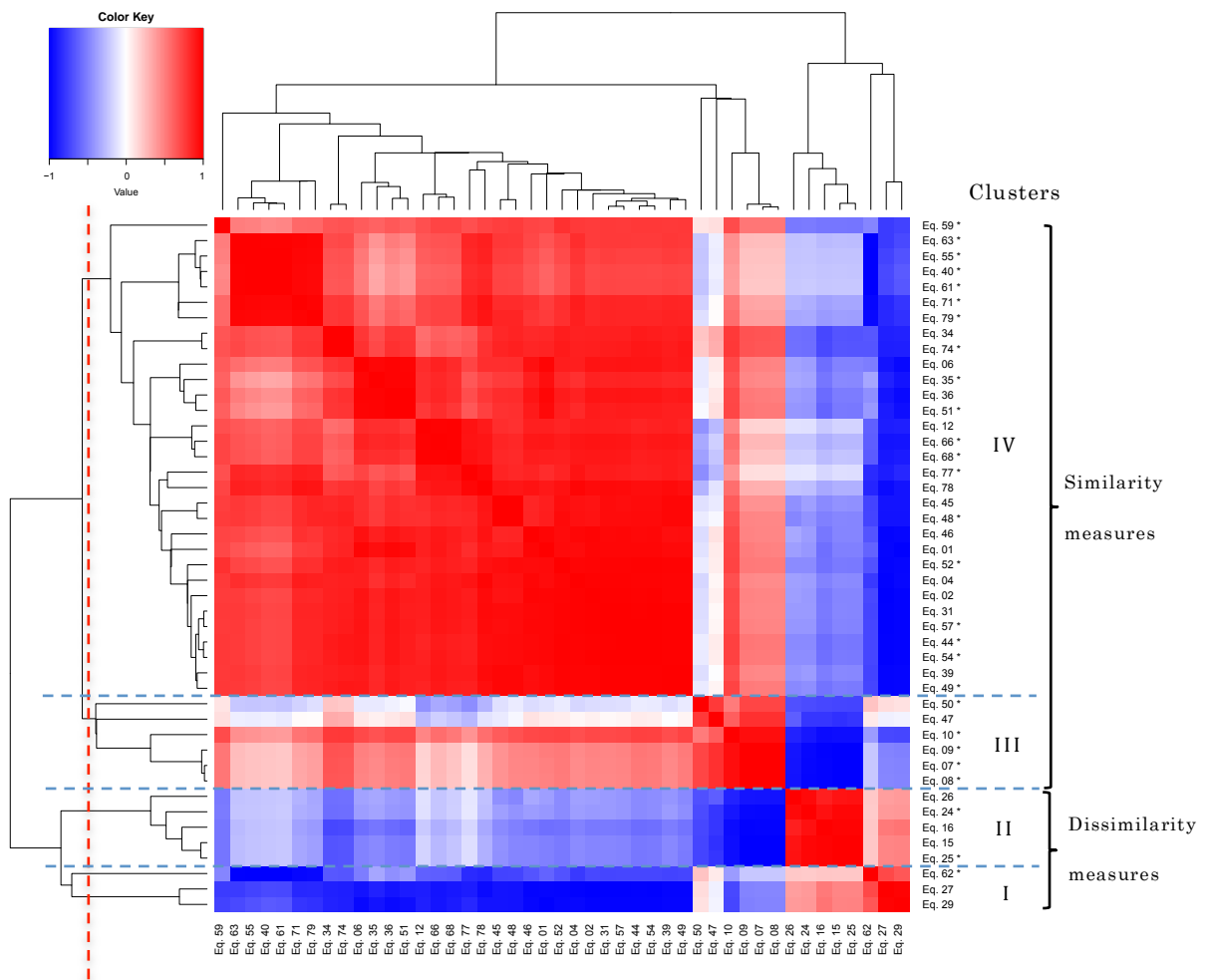
We applied hierarchical clustering again to these 45 equations to give a better understanding of relationships between selected equations. The resulted dendrogram together with the heatmap of Jamu and Kampo datasets are shown in Fig. 3.3 and Fig. 3.4. Both datasets generated more or less the same heatmap and list of equations in every cluster. We can roughly identify four main clusters (I, II, III, and IV). The hierarchical clustering clearly separated the equations on the basis whether they measure similarity or dissimilarity. Although both similarity/dissimilarity measures may produce the same coefficient range, they work in the opposite way. The higher the similarity between two herbal medicine formulas, the higher the similarity coefficients. On the other hand, the higher the similarity between two herbal medicine formulas the lower the dissimilarity coefficients. Therefore, the agglomerative clustering with centroid linkage performs well in the process to separate similarity and dissimilarity equations.



All the equations belonging to clusters I and II are for measuring dissimilarity whereas the equations belonging to clusters III and IV are for measuring similarity. Conversely, the equations that include negative match quantity  $d$  spread throughout all the clusters. This result indicates that the equations cannot be grouped based on the existence of negative match quantity  $d$ .



**Figure 3.3.** The heatmap and dendrogram of remaining binary similarity and dissimilarity measures using Jamu data. The asterisk (\*) indicates the negative match quantity  $d$  is used in the equation.



**Figure 3.4.** The heatmap and dendrogram of remaining binary similarity and dissimilarity measures using Kampo data. The asterisk (\*) indicates the negative match quantity  $d$  is used in the equation.

### 3.4.2. ROC analysis of selected equations

The ROC curves were created for each binary similarity/dissimilarity equation to compare their performance. Initially, we normalized the similarity and dissimilarity coefficients, such that their minimum becomes 0, and maximum becomes 1, before using them to create the ROC curves. In the case of equations

that measure dissimilarity, we transformed a normalized dissimilarity coefficient  $D$  to a similarity coefficient  $S$  for the sake of comparison by using the following equation  $S = 1 - D^2$  (Warrens 2008; Jackson et al. 1989).

Our objective is to assess the capability of the equations to separate the Jamu pairs into match and mismatch efficacy classes based on their similarity coefficients using ROC analysis. In order to create an ROC curve corresponding to an equation, we need the distributions of match class and mismatch class Jamu pairs with respect to their similarity values calculated by the equation. We divided the range of the similarity coefficient into 100 equal intervals, and the lower limit of each interval was considered as a threshold. Corresponding to every threshold,  $TP$  and  $FN$  were determined from the distribution of match class, and  $FP$  and  $TN$  were determined from the distribution of mismatch class. In our case,  $TP$  and  $FP$  are the numbers of Jamu pairs with the similarity value larger than or equal to a threshold, and  $FN$  and  $TN$  are the numbers of Jamu pairs with the similarity value smaller than a threshold.  $FPR$  and  $TPR$  were then calculated for every threshold using Eq. 82. We produced the ROC curve by plotting the resulting  $FPR$  on the x-axis and  $TPR$  on the y-axis. The goal of ROC curve is to be in the top left corner of ROC space. In perfect or ideal classification, the ROC curve follows the vertical line from (0,0) to (0,1) and then horizontal line up to (1,1). In the case of random data, the ROC curve follows the diagonal line from (0,0) to (1,1). In the case of real data, the ROC curve usually follows an above diagonal line. The (0,1) is the optimum classification point where  $FPR$  is zero and  $TPR$  is one and hence the (0,1) point will be referred to as ‘optimum point’. The performance of a classifier was assessed either by measuring the minimum distance from the optimum point to the curve or by measuring the AUC. In the case of the minimum distance, the lower is the value of the minimum distance the better is the performance of the classifier. In the case of the AUC, the bigger is the AUC value; the better is the performance of the classifier.

In order to assess the effectiveness of an equation using the minimum distance, the ROC curve was generated by using all of the Jamu pairs from match and mismatch efficacies. The Euclidean distance metric was used to measure the distance from the (0,1) point to the ( $FPR$ ,  $TPR$ ) points for all 45 selected equations. In addition, we created 20 ROC curves for each equation considering in each case the match class Jamu pairs and one of the 20 different mismatch class samples. Thus, we obtained 20 AUCs of the ROC curve for each equation and averaged those values to determine the overall AUCs corresponding to an equation. The ROCR package (Sing et al. 2005) was used to calculate the AUC values. Table 3.4 shows the results of ROC analysis and also Kappa scores for Jamu data. A value inside the bracket in the minimum distance and mean Kappa columns of Table 3.4 represents the ranking of an equation if we order based on the respective columns. Standard deviations from both metrics are relatively similar and small; those are  $2.4 \times 10^{-4}$  for mean AUCs and  $0.6 \times 10^{-4}$  for the mean of Kappa scores. The scatter plot of minimum distances and mean of AUCs corresponding to 45 equations for both datasets is shown in Fig. 3.5. Based on the scatter plot generated using Jamu data in Fig. 3.5a, the 45 equations are empirically divided into four groups (C1, C2, C3, and C4). C1 is represented by equations from cluster III, C2 is represented by equations from clusters I and III, C3 is represented by equations from clusters II and IV, and C4 is represented by equations from cluster II. Cluster I, II, III, and IV are associated with Fig. 3.3. The well-performing equations corresponding to both approaches were obtained in C1, which consists of Eqs. 48, 49, 54, 68, and 79. The Michael similarity (Eq. 68) produces the lowest minimum distance, and the highest AUC is obtained by the Forbes-2 similarity (Eq. 48). The ROC curves generated using Michael and Forbes-2 similarities for all datasets are shown in Fig. 3.6. As expected, the ROC curves corresponding to all random datasets follow the diagonal line and that corresponding to Jamu data follows the above diagonal line. Most equations with the highest AUC values are similarity-measuring equations, and these equations belong to cluster III in Fig. 3.3. Out of these

equations, the Lance & Williams distance (Eq. 27) produces the highest AUC value among dissimilarity-measuring equations.

**Table 3.4.** *The ROC analysis and Cohen's Kappa score of Jamu data.*

No	Eq. IDs	S/D	Incl. <i>d</i>	ROC analysis		Cohen's	
				Mean AUCs	Min.	Mean Kappa	
1	Eq. 48	S	Y	0.616	0.587 ( 3 )	0.088	( 13 )
2	Eq. 74	S	Y	0.613	0.599 (29)	0.024	( 28 )
3	Eq. 49	S	Y	0.613	0.588 ( 4 )	0.076	( 15 )
4	Eq. 54	S	Y	0.611	0.590 ( 5 )	0.074	( 19 )
5	Eq. 44	S	Y	0.611	0.599 (19)	0.073	( 21 )
6	Eq. 66	S	Y	0.611	0.599 (26)	0.023	( 31 )
7	Eq. 68	S	Y	0.610	0.583 ( 1 )	0.024	( 29 )
8	Eq. 79	S	Y	0.610	0.583 ( 2 )	0.090	( 11 )
9	Eq. 78	S		0.609	0.599 (28)	0.092	( 8 )
10	Eq. 46	S		0.609	0.599 (20)	0.065	( 23 )
11	Eq. 01	S		0.609	0.599 (10)	0.052	( 24 )
12	Eq. 04	S		0.609	0.599 (11)	0.089	( 12 )
13	Eq. 06	S		0.609	0.599 (12)	0.036	( 27 )
14	Eq. 27	D		0.609	0.599 (14)	0.109	( 7 )
15	Eq. 02	S		0.609	0.599 ( 8 )	0.074	( 20 )
16	Eq. 36	S		0.608	0.600 (31)	0.040	( 25 )
17	Eq. 29	D		0.608	0.599 (15)	0.076	( 16 )
18	Eq. 31	S		0.608	0.599 (16)	0.076	( 17 )
19	Eq. 57	S	Y	0.608	0.599 (22)	0.076	( 18 )
20	Eq. 71	S	Y	0.608	0.599 ( 9 )	0.152	( 6 )
21	Eq. 39	S		0.607	0.599 (17)	0.078	( 14 )
22	Eq. 62	D	Y	0.606	0.599 (24)	0.185	( 1 )
23	Eq. 63	S	Y	0.606	0.599 (25)	0.167	( 5 )
24	Eq. 55	S	Y	0.606	0.599 (21)	0.180	( 3 )
25	Eq. 61	S	Y	0.606	0.599 (23)	0.183	( 2 )
26	Eq. 40	S	Y	0.605	0.599 (18)	0.180	( 4 )
27	Eq. 34	S		0.605	0.600 (30)	0.024	( 30 )
28	Eq. 45	S		0.605	0.599 ( 7 )	0.091	( 10 )
29	Eq. 52	S	Y	0.604	0.597 ( 6 )	0.092	( 9 )
30	Eq. 77	S	Y	0.604	0.599 (27)	0.067	( 22 )

No	Eq. IDs	S/D	Incl. $d$	ROC analysis		Cohen's	
				Mean AUCs	Min.	Mean Kappa	
31	Eq. 51	S	Y	0.604	0.602 ( 32 )	0.039	( 26 )
32	Eq. 12	S		0.604	0.599 ( 13 )	0.022	( 32 )
33	Eq. 10	S	Y	0.556	0.656 ( 33 )	0.014	( 34 )
34	Eq. 35	S	Y	0.546	0.671 ( 34 )	0.018	( 33 )
35	Eq. 59	S	Y	0.545	0.671 ( 35 )	0.013	( 35 )
36	Eq. 24	D	Y	0.529	0.860 ( 44 )	0.000	( 43 )
37	Eq. 15	D		0.529	0.680 ( 39 )	0.004	( 42 )
38	Eq. 08	S	Y	0.529	0.680 ( 37 )	0.010	( 39 )
39	Eq. 09	S	Y	0.529	0.680 ( 38 )	0.010	( 36 )
40	Eq. 16	D		0.529	0.680 ( 40 )	0.010	( 38 )
41	Eq. 07	S	Y	0.529	0.680 ( 36 )	0.010	( 37 )
42	Eq. 25	D	Y	0.526	0.680 ( 41 )	0.004	( 41 )
43	Eq. 26	D	Y	0.517	0.895 ( 45 )	0.000	( 44 )
44	Eq. 47	S		0.515	0.684 ( 42 )	0.005	( 40 )
45	Eq. 50	S	Y	0.466	0.754 ( 43 )	-0.008	( 45 )

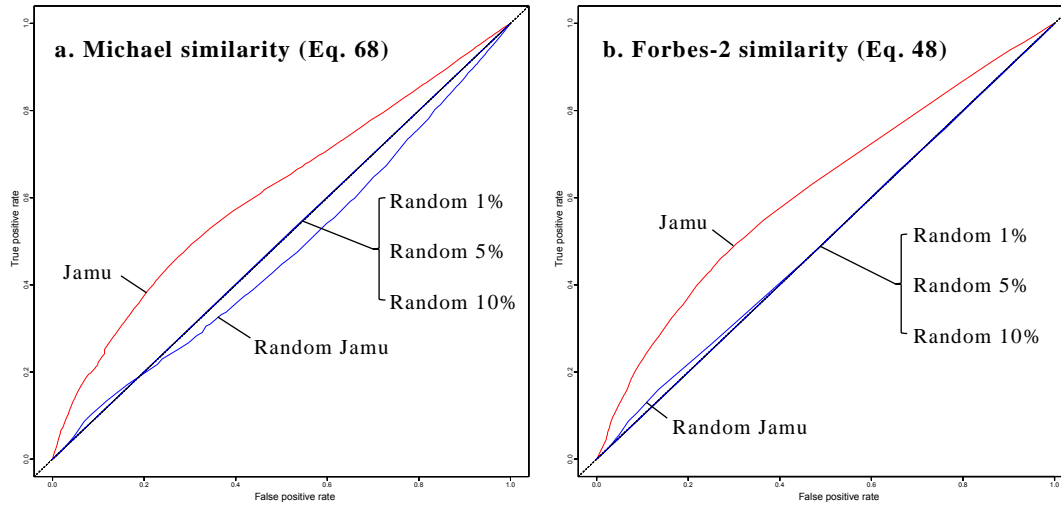
We repeated our experiments for Kampo data following the same procedures. The results of ROC analysis and also Cohen's Kappa are shown in Table 3.5. In addition, the plot between minimum distances and mean AUCs of Kampo data is shown in Fig. 3.5b. The remaining equations are clustered into three groups (C1, C2, and C3). The most suitable binary equations for classifying Kampo data were found in the cluster C1, with Tarwid Similarity (Eq. 40) and Variant of Correlation similarity (Eq. 79) producing the lowest minimum distance and the highest mean AUCs, respectively, which are different from the top ranking equations in case of Jamu data. Only 5 of top-10 well-performing equations corresponding to Jamu data matches with those corresponding to Kampo data with the different order. These results indicate different dataset produce a different ranking of equations, and there is no superior equation that can perform well for all datasets (Gelbard et al. 2007). Each binary similarity and dissimilarity equations has its own characteristics and fits for a specific problem.

Therefore, our proposed method can be used to choose the appropriate equations wisely, depending on the characteristics of the data to analyze.

In case of Jamu and Kampo pairs, the negative match quantity  $d$  is much higher compared to the positive match  $a$  and the absence mismatches  $b$  and  $c$ . One of our objectives is to understand the effect of  $d$  in calculating similarity/dissimilarity coefficients between herbal medicines. Among the equations that do not include  $d$ , the Simpson similarity (Eq. 45) and the Forbes-1 similarity (Eq. 34) produce the lowest minimum distance in Jamu and Kampo data, respectively. Furthermore, the Derived Jaccard similarity (Eq. 78) and the McConnaughey (Eq. 39) produce the highest AUC in Jamu data and Kampo data. Out of 79 equations in Table 3.1, 46 equations use  $d$  in their expressions. Interestingly, the equations that include  $d$  perform better in measuring similarity/dissimilarity in both datasets. The best performing equations corresponding to minimum distance and mean AUCs for Jamu data are Eqs. 68 and 48, which include negative match quantity  $d$ . Likewise, the best equations in the Kampo data (Eqs. 79 and 40) also include negative match quantity  $d$ . Then, the top-5 well-performing equations corresponding to both approaches in Jamu and Kampo datasets include  $d$ . If we also consider another metric to rank the classifier performance, i.e. Cohen's Kappa, we found a consistent result. That is top-5 equations with the largest Kappa score also include  $d$  (Table 3.4-3.5). It implies the similarity between Jamu pairs and Kampo pairs are influenced by the negative matches. This result supports the findings of Zhang *et al.* that all possible matches,  $S_{ij}$  where  $i, j \in \{0,1\}$ , should be considered for better classification results (Zhang & Sargur N. Srihari 2003). Moreover, the performance measurement of binary similarity/dissimilarity equations using the AUC of ROC curve is preferable to the minimum distance because this approach considers all ( $FPR$ ,  $TPR$ ) points, not only a single point with the minimum distance to the optimum point.







**Figure 3.6.** The ROC curves of Michael and Forbes-2 similarities for Jamu and random datasets.

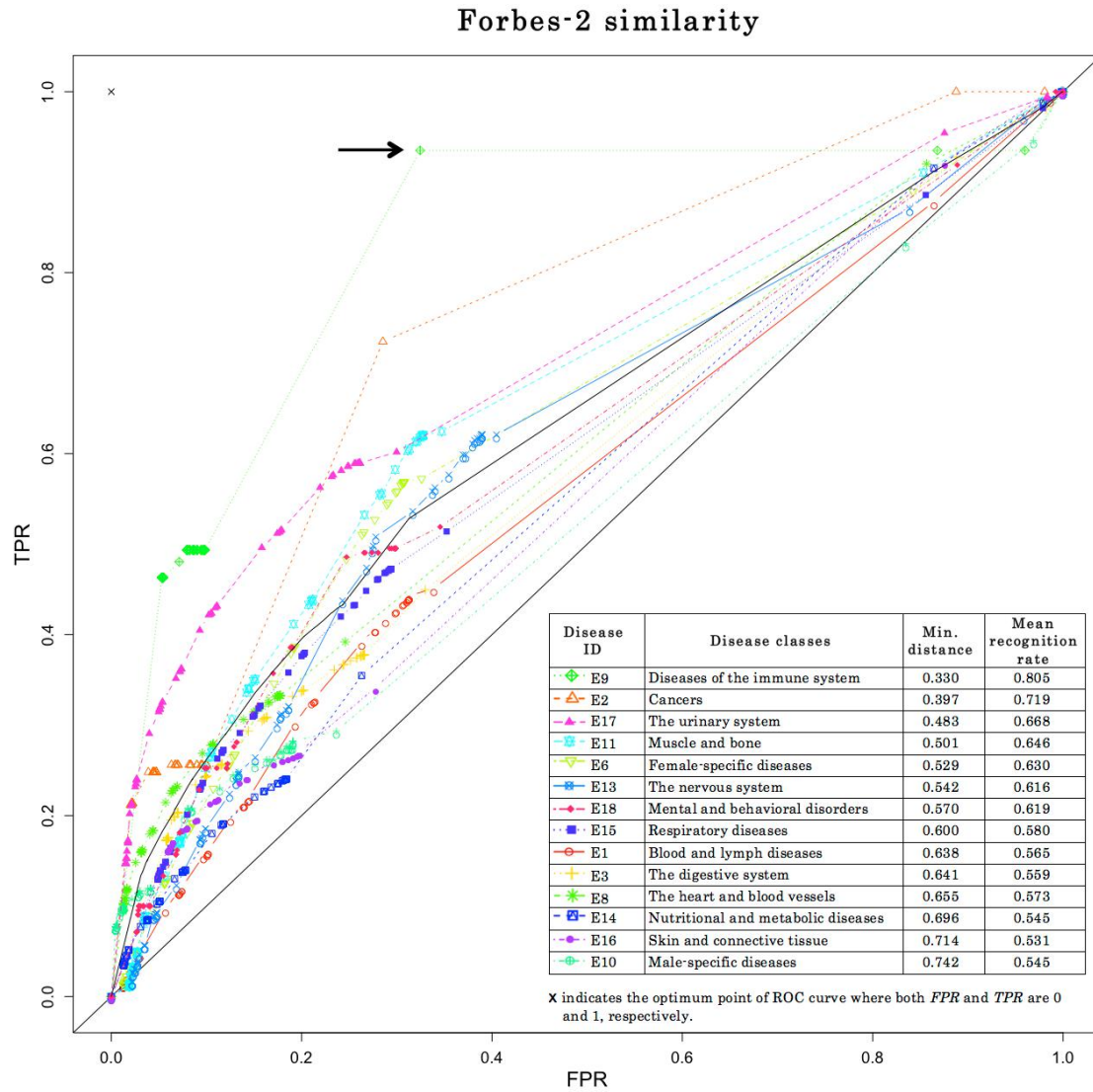
**Table 3.5.** The ROC analysis and Cohen's Kappa score of Kampo data.

No	Eq.	IDs	S/D	Incl. <i>d</i>	ROC analysis			Cohen's Kappa	
					Mean AUCs	SD mean AUCs	Min. distance	Mean Kappa	SD mean Kappa
1	Eq. 79	S	Y		0.610	0.001	0.607 ( 9)	0.069 (14)	0.001
2	Eq. 55	S	Y		0.609	0.001	0.604 ( 2)	0.106 ( 1)	0.001
3	Eq. 61	S	Y		0.609	0.001	0.606 ( 5)	0.106 ( 2)	0.001
4	Eq. 63	S	Y		0.609	0.001	0.606 ( 6)	0.099 ( 5)	0.001
5	Eq. 62	D	Y		0.609	0.001	0.610 (16)	0.101 ( 4)	0.001
6	Eq. 48	S	Y		0.608	0.001	0.608 (12)	0.084 ( 9)	0.001
7	Eq. 49	S	Y		0.608	0.001	0.607 (11)	0.069 (15)	0.001
8	Eq. 44	S	Y		0.608	0.001	0.610 (15)	0.065 (21)	0.001
9	Eq. 54	S	Y		0.607	0.001	0.607 ( 8)	0.066 (20)	0.001
10	Eq. 39	S			0.607	0.002	0.607 (10)	0.070 (13)	0.001
11	Eq. 57	S	Y		0.606	0.001	0.611 (17)	0.067 (18)	0.000
12	Eq. 71	S	Y		0.606	0.001	0.608 (14)	0.092 ( 6)	0.001
13	Eq. 51	S	Y		0.606	0.001	0.612 (18)	0.040 (27)	0.001
14	Eq. 31	S			0.606	0.001	0.612 (20)	0.068 (17)	0.001
15	Eq. 29	D			0.606	0.001	0.612 (19)	0.068 (16)	0.001

No	Eq.	IDs	S/D	Incl. <i>d</i>	ROC analysis			Cohen's Kappa	
					Mean	SD mean	Min.	Mean Kappa	SD mean
					AUCs	AUCs	distance		
16	Eq. 52	S	Y	0.606	0.001	0.608 (13)	0.078 (10)	0.001	
17	Eq. 36	S		0.606	0.001	0.612 (21)	0.042 (26)	0.001	
18	Eq. 74	S	Y	0.605	0.002	0.606 ( 4)	0.037 (29)	0.001	
19	Eq. 45	S		0.605	0.001	0.606 ( 7)	0.086 ( 8)	0.001	
20	Eq. 04	S		0.605	0.001	0.615 (29)	0.075 (12)	0.001	
21	Eq. 27	D		0.605	0.001	0.615 (30)	0.091 ( 7)	0.001	
22	Eq. 06	S		0.605	0.001	0.618 (32)	0.032 (40)	0.001	
23	Eq. 02	S		0.604	0.001	0.615 (28)	0.065 (22)	0.001	
24	Eq. 34	S		0.604	0.001	0.605 ( 3)	0.035 (36)	0.001	
25	Eq. 01	S		0.604	0.001	0.616 (31)	0.047 (24)	0.001	
26	Eq. 40	S	Y	0.604	0.001	0.604 ( 1)	0.102 ( 3)	0.002	
27	Eq. 78	S		0.602	0.001	0.614 (25)	0.075 (11)	0.001	
28	Eq. 46	S		0.600	0.001	0.613 (23)	0.055 (23)	0.001	
29	Eq. 68	S	Y	0.597	0.001	0.612 (22)	0.036 (32)	0.001	
30	Eq. 66	S	Y	0.597	0.001	0.614 (24)	0.035 (37)	0.001	
31	Eq. 59	S	Y	0.591	0.001	0.614 (26)	0.043 (25)	0.001	
32	Eq. 35	S	Y	0.590	0.001	0.615 (27)	0.036 (35)	0.001	
33	Eq. 12	S		0.590	0.001	0.621 (33)	0.034 (38)	0.000	
34	Eq. 77	S	Y	0.589	0.001	0.621 (34)	0.066 (19)	0.000	
35	Eq. 10	S	Y	0.584	0.001	0.630 (35)	0.036 (31)	0.001	
36	Eq. 26	D	Y	0.568	0.001	0.653 (43)	0.015 (43)	0.001	
37	Eq. 24	D	Y	0.564	0.001	0.651 (42)	0.017 (42)	0.001	
38	Eq. 25	D	Y	0.564	0.001	0.650 (36)	0.032 (41)	0.001	
39	Eq. 08	S	Y	0.564	0.001	0.651 (38)	0.036 (33)	0.001	
40	Eq. 16	D		0.564	0.001	0.651 (41)	0.037 (30)	0.001	
41	Eq. 15	D		0.563	0.001	0.651 (40)	0.032 (39)	0.001	
42	Eq. 07	S	Y	0.563	0.001	0.651 (37)	0.036 (34)	0.001	
43	Eq. 09	S	Y	0.563	0.001	0.651 (39)	0.037 (28)	0.001	
44	Eq. 47	S		0.518	0.001	0.683 (44)	0.010 (44)	0.001	
45	Eq. 50	S	Y	0.501	0.001	0.702 (45)	-0.004 (45)	0.000	

For further insight into the matter, we examined the performance of the equations for every disease class in Jamu data separately using the same approach. We created match and mismatch datasets for every disease class from all Jamu pairs. The match class consists of Jamu pairs with the same efficacy class and the mismatch class consists of Jamu pairs with different efficacy class, but one of the Jamu formulas in that pair has the same efficacy class as the match class. To measure the AUC of ROC curve, we created 20 mismatch classes each equal to the size of the match class by using the bootstrap method. Thus, we obtained 20 AUCs of the ROC curve for each disease class and each equation, and we averaged those 20 values to determine the overall AUCs corresponding to a disease class and an equation (Appendix C). Fig. 3.7 shows the ROC curves for every disease class using Forbes-2 similarity coefficients. The immune system disease class (E9) produces the highest AUC score and the highest average of AUCs (for all 45 equations). Moreover, the best classification is obtained in case of immune system class indicated by an arrow in Fig. 3.7, with the average of recognition rate of 0.805. The relatively high recognition rate of the E9 class corresponds to our knowledge that the disease of immune system class is a very specific disease, and utilization of the crude drug is restricted compared to other disease classes. The minimum distance of a ROC curve from the optimum point (expressed by Eq. 83) indicates the difficulty of classification i.e. the higher the minimum distance, the more difficult it is to achieve a successful classification. Therefore, when the minimum distance is close to zero, it implies that good classification of the data is possible. In case of classification of Jamu formulas concerning individual diseases, a relatively lower minimum distance was obtained for specific type of disease classes such as diseases related to E9 and urinary systems (E17), which indicates that very specific types of medicinal plants are used to make such Jamu formulas. On the other hand, the disease classes such as those related to digestive systems (E3) and nutritional and metabolic disease (E14) are caused by diverse factors and therefore the corresponding Jamu

formulas are made using diverse types of plants resulting in relatively higher minimum distance for these disease classes (Fig. 3.7).



**Figure 3.7.** The ROC curves for every disease class in Jamu data using Forbes-2 similarity coefficients. The average of recognition rate was calculated as  $\frac{1}{2} \left[ \frac{TP}{TP+FN} + \frac{TN}{TN+FP} \right]$  by using the *TP*, *FN*, *FP*, and *TN* values from (*FPR*, *TPR*) point with the shortest distance to the optimum point (0,1).

### 3.5. Summary

Different binary similarity and dissimilarity measures yield different similarity/dissimilarity coefficients, which in turn causes differences in downstream analysis e.g. clustering. Therefore, determining appropriate binary similarity and dissimilarity coefficients is an essential aspect of big data analysis in versatile areas of scientific research including chemometrics and bioinformatics. The experimental results show that the binary similarity and dissimilarity measures that include the negative match quantity  $d$  in their expressions have a better capability to separate herbal medicine pairs than those equations that exclude  $d$ . Moreover, we obtained a different ranking of binary equations and recommended equations for different datasets, i.e. Jamu and Kampo data. Thus, this result indicates the selection of binary similarity and dissimilarity measures is data dependent, and we should choose the binary similarity and dissimilarity measures wisely depending on the data to be processed. In case of Jamu data, the biggest AUC value is obtained by the Forbes-2 similarity (Eq. 48). Conversely, the Variant of Correlation similarity (Eq. 79) is recommended for classifying Kampo pairs into match and mismatch classes. The procedure followed in this work can also be used to find suitable binary similarity and dissimilarity measures under similar situations in other applications.



## Chapter 4

# Metabolomic studies of Jamu medicines

We extended our study in Chapter 2 by including metabolites exist in the plants used as main ingredients of Jamu medicines using machine-learning approaches. This study consists of two main activities, i.e. prediction of Jamu efficacy based on its active compounds and identification of important metabolites for each efficacy group. Support Vector Machine (SVM) and Random Forest (RF) were chosen as classifiers in the prediction of Jamu efficacy. Single Filtering (SF) algorithm and Regularized Random Forest (RRF) were applied to eliminate inconsistent Jamu formulas and to select the features, respectively. Then, the proposed methods for prediction of Jamu efficacy were evaluated by accuracy and Cohen's Kappa score from  $k$ -fold cross-validation. According to the best model generated by RF classifier, we extracted the rules using Interpretable Tress (inTrees) framework. The important metabolites for each efficacy group were then determined based on the distribution of error rate and frequency of the resulted rules.

### 4.1. Background

A metabolite is any organic molecule detectable with a molecular weight < 1500 Da. Then, a comprehensive quantitative and qualitative analysis of all metabolites present in a specific cell, tissue, or organism is known as metabolomic (Shyur & Yang 2008; Mahadevan et al. 2008). This concept was first reported in 1998 to examine the interplay between the global metabolite pool and specific environmental conditions in *Escherichia coli* (Tweeddale et al. 1998). Metabolomic data is mainly obtained from high throughput technologies by identifying and

characterizing all the small molecules or metabolites, i.e. nuclear magnetic resonance (NMR) spectroscopy and gas chromatography-mass spectrometry (GC-MS). In the medical research, metabolomic is important because 89% of all known drugs are small molecules and 50% of all drugs are derived from pre-existing metabolites. In addition, metabolomic has shown a great success for early diagnosis of diseases or preclinical screening of candidate drugs in the pharmaceutical industry (Lindon et al. 2003).

A remarkably diverse array of metabolites was produced by plants. It is estimated more than 200,000 secondary metabolites in the plant kingdom (Dixon & Strack 2003). Traditional medicines, which are mainly composed by a part of plants or a whole plant, hold great public and medical interest worldwide as sources of novel lead compounds for drug development. Most of real drugs were developed by utilizing effective compounds in the plants that emerge from drug discovery processes. In this case, drug discovery can be used to recognize the active ingredient from traditional remedies by compounds screening and isolation of active compounds, i.e. isolation of xanthorrhizol from *Curcuma xanthorrhiza*. Identification of effective compounds is very challenging for developing new drugs. If we can find the effective compounds from traditional remedies, this may lead to finding side-effect free modern medicines. Additionally, we can extract the same compounds from other plants as an alternative, i.e. substitute a compound from a tropical plant species with a compound from other sub-tropical plants.

Machine learning is a more recent technique of multivariate analysis. This method can be trained to learn rules and generate a model from the input data and subsequently be applied to analyze new data. Application of machine learning methods in the metabolomic studies has had recent successes, such as predicting target of compounds (Nidhi et al. 2006), identification of the efficacy target from phenotypic drug discovery (Schirle & Jenkins 2016) and so on. SVM and RF are machine-learning methods for supervised learning, and it can be used for



classification of data. Additionally, both methods have also been implemented to analyze metabolomic data with success (Mahadevan et al. 2008; Chen et al. 2013).

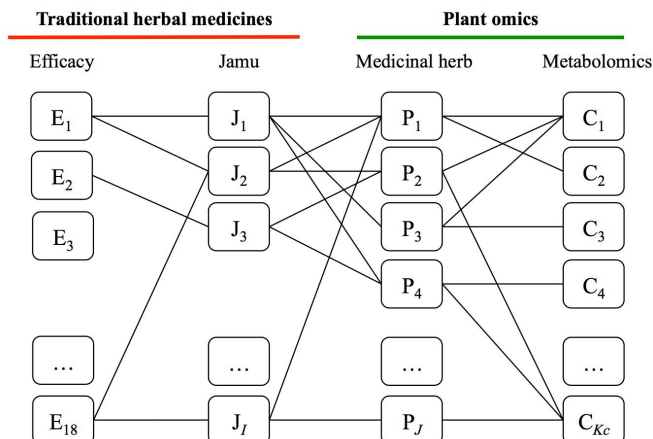
The extension of plant-efficacy relations of Jamu medicines with plant-metabolite relations may lead to a sparse and noisy matrix. Therefore, a proper preprocessing technique is needed to maintain the quality of Jamu data. Learning algorithms can be utilized as a noise filter to remove the outliers in the extended dataset (Brodley & Friedl 1999). This approach was motivated by removing outliers in regression analysis (Weisberg 2005). Filtering will significantly improve the prediction or classification accuracy by identifying and eliminating mislabeled or inconsistent Jamu formulas, such as Jamu formulas with the same composition but classified into different disease class. One of the filtering methods proposed by Brodley & Friedl (1999) is SF algorithm. This method uses one classifier with cross-validation over the training data to remove noises from samples.

In Chapter 2, we examined the relationship between efficacy and Jamu formulas based on plants used as Jamu ingredients. Hence, the systematization of Jamu formulas can be extended by considering plant's metabolites. For each plant used as Jamu ingredient, we can narrow down our analysis by including its metabolites from published literature or compound databases, i.e. KNApSAcK Family database (Afendi, Okada, et al. 2012; Weber & Kim 2016). Thus, it is interesting to model the compounds of Jamu and use this model to predict Jamu efficacy based on its active compounds. After the prediction of Jamu efficacy based on its metabolites is already established, we can elaborate our analysis by determining important metabolites for each efficacy group and also examining the pharmacological activity of effective compounds. These results will give a better understanding why some Jamu formulas can be used to remedy a disease in the molecular level, and also either to develop new Jamu formulas or to utilize alternative plants as Jamu ingredients.

## 4.2. Datasets

We used the same Jamu data from Chapter 2 (Table 2.2). There are 3,134 Jamu formulas, consist of 465 plants, and each formula is mapped into 16 efficacy groups. We also excluded from our analysis efficacy groups that only consist of very few Jamu formulas, i.e. Jamu formulas from ear, nose, and throat disease (E4) and disease of the eye (E5). In order to extend the relationship between efficacy-plant relations by including plant's metabolites, we determined plants as main ingredients from previous studies. Afendi et al. identified 231 of 465 plants were effective as main ingredients for at least one efficacy (Afendi, Darusman, et al. 2013). In Chapter 2, we also recognized another 17 plants as main ingredients. According to important plants identified as the main ingredient for each efficacy group, we collected metabolites information of 238 plants from published literature. We obtained 6,597 plant-metabolite relations, and it was composed by 3,490 unique metabolites and 26 parts of plants.

Fig. 4.1 depicts the relationship between efficacy, Jamu, plant, and plant's metabolites. Initially, we extended the efficacy-plant relations of Jamu data with plant-metabolite relations. Then, we eliminated Jamu-metabolite relations that classified into two or more efficacy groups and also Jamu formulas with no metabolite information from our analysis. Therefore, there were 2,992 Jamu-metabolite relations exist. In Fig. 4.1,  $I$  is equal to the number of Jamu formulas ( $I = 2,992$ ),  $J$  is the number of plants as main ingredients ( $J = 238$ ), and  $Kc$  is the number of metabolites found in the plants used as Jamu ingredients ( $Kc = 3,490$ ). Table 4.1 and Table 4.2 show an illustration of data structure relating efficacy-metabolite obtained from network in Fig. 4.1 and the distribution of Jamu formulas according to 14 efficacy groups, respectively. The relationship between Jamu and its effective compounds is represented by a Jamu-metabolite matrix.



**Figure 4.1.** The illustration of a network connecting efficacy, Jamu, plant, and metabolite.

**Table 4.1.** Representation of Jamu, metabolites and efficacy in Fig. 4.1 as a two-dimensional matrix.

Jamu	Compounds						Efficacy
	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	...	C <sub>Kc</sub>	
J <sub>1</sub>	1	1	1	1	...	0	EJ <sub>1</sub>
J <sub>2</sub>	1	1	0	0	...	1	EJ <sub>2</sub>
J <sub>3</sub>	1	0	0	1	...	1	EJ <sub>3</sub>
...	...	...	...	...	...	...	
J <sub>I</sub>	1	0	0	0	...	1	EJ <sub>I</sub>

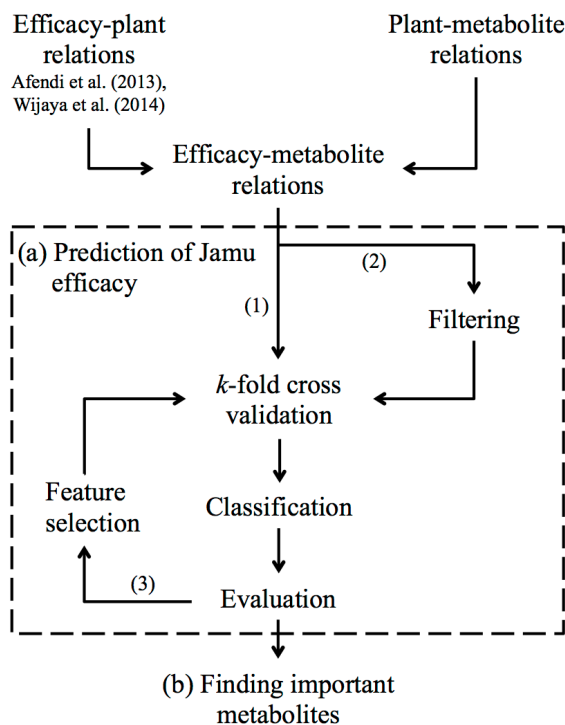
### 4.3. Methods

The schematic diagram of our method to predict Jamu efficacy based on its metabolites and also to extract important metabolites for each efficacy group in Jamu medicines is shown in Fig. 4.2. We started our study by expanding the efficacy-plant relations with plant-metabolite relations. Then, efficacy-metabolite relations were examined by utilizing three different approaches. First, we divided the efficacy-metabolite dataset into five groups using  $k$ -fold cross-validation (in this case,  $k$  is equal to 5), and it was directly used as an input for the classification process. Second, we filtered and eliminated inconsistent Jamu

formulas before dividing the dataset using 5-fold cross-validation. Then, we examined the quality of filtering results by predicting Jamu efficacy based on its metabolites. Third, we also examined the classifier performance by applying a combination of filtering and feature selection. Finally, the important metabolites were extracted for each efficacy group by utilizing the best approach. The accuracy and Cohen’s Kappa score were used as metrics to measure the quality of filtering and the classification results.

**Table 4.2.** *The distribution of Jamu formulas according to its metabolites taken from plants used as main ingredients.*

IDs	Class of diseases	Number of Jamu	Number of metabolites	
			Min	Max
E1	Blood and Lymph Diseases	195	2	440
E2	Cancers	30	9	252
E3	The Digestive System	455	2	398
E6	Female-Specific Diseases	382	5	376
E8	The Heart and Blood Vessels	56	3	377
E9	Diseases of the Immune System	22	11	208
E10	Male-Specific Diseases	17	2	378
E11	Muscle and Bone	647	3	439
E13	The Nervous System	32	8	310
E14	Nutritional and Metabolic Diseases	572	1	431
E15	Respiratory Diseases	311	8	383
E16	Skin and Connective Tissue	162	1	280
E17	The Urinary System	90	5	436
E18	Mental and Behavioral Disorders	21	16	398



**Figure 4.2.** *The schematic diagram of the prediction of Jamu efficacy and identification of important metabolites for each efficacy group.*

#### 4.3.1. Single Filtering algorithm

SF algorithm uses a classifier to remove noise from samples (Brodley & Friedl 1999). There are two approaches that can be implemented for SF algorithm. First, the same learning algorithm is used to construct both the filter and the final classifier. Second, we can use different algorithm between filtering and constructing classifier. In this case, some algorithm performs as a good filter for other algorithms such as some algorithms perform as good feature selection methods for others. The SF process mainly consists of two steps as follows:

- a. Identify candidate instance by using a single learning algorithm to tag instances as correctly or incorrectly classified.
- b. Form classifier using a filtered dataset for which misclassified instances are removed. The filtered dataset can be based on one classifier.

We implemented SF algorithm to eliminate inconsistent efficacy-metabolite relations from Jamu data. We followed Brodley & Friedl (1999) method by performing the filtering process using cross-validation over one iteration. SVM was chosen as a classifier for filtering because of its ability to construct optimal separator between classes (Gunn 1998; Schüldt et al. 2004).

#### 4.3.2. Support Vector Machine

Initially, SVM has been developed as a binary classifier by constructing optimal linear classifier (hyperplane), which has the largest margin between two classes. The hyperplane is constructed by simultaneous minimization of the empirical classification error and maximization of the geometric margin (Vapnik 1998). If we have  $n$  training data pairs,  $T = \{(x_i, y_i)\}$ ,  $i = 1, \dots, n$  where  $x_i (\in \mathbb{R}^p)$  is a feature vector representing Jamu  $i$  and  $y_i$  is the class label of  $x_i$ . In case of binary classification,  $y_i = \{-1, 1\}$ , while for multiple class classification problem ( $c > 2$ ),  $y_i = \{1, 2, \dots, c\}$ . The decision function of SVM is defined as  $f(x) = w^T x + b$  where  $w = [w_1, w_2, \dots, w_p]^T$  is the weight vector, and  $b$  is a scalar. The optimization problem that SVMs would like to minimize is as follows:

$$\min_{w \in \mathbb{R}^p, \xi_i \in \mathbb{R}^+} \frac{1}{2} \|w\|^2 + C \sum_i^n \xi_i$$

subject to  $y_i(w^T x_i + b) \geq 1 - \xi_i$  where  $C$  is a regulation parameter, as a trade-off between the width of the margin and the number of misclassification, and  $\xi_i$  is a slack variable that allows an example to be in the margin ( $0 \leq \xi_i \leq 1$ , also called a margin error) or to be misclassified ( $\xi_i > 1$ ). In addition, SVM can be extended to classify not linearity separable data by utilizing kernel trick. The original data will be mapped into higher dimensional feature space, and a linear classifier is constructed in this feature space. It is the same as constructing a non-linear classifier in the original input space. There are three kernel functions that

commonly used, i.e. linear kernel ( $K(x_i, x_j) = x_i^T x_j$ ), polynomial kernel ( $K(x_i, x_j) = (\gamma x_i^T x_j + 1)^d, \gamma > 0$  with  $d$  is the degree of the polynomial and  $\gamma$  is the inverse of the radius of influence of samples selected by the model as support vectors), and radial basis function (RBF) kernel ( $K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \gamma > 0$ ) (Mahadevan et al. 2008; Hussain et al. 2011).

Commonly, the problems in the real world are multiclass classification problems. One of SVM approaches to handle multi-class problems is multiclass SVMs One-Against-One (OAO) (Zhou & Tuck 2007). Multiclass SVMs OAO uses  $[c(c - 1)/2]$  binary SVM classifiers for all pairs of classes (Duan et al. 2007). Hence, for every pair of classes, a binary SVM problem is solved. A majority vote is used to assign an instance to a class that has the largest number of votes, called as Max Wins strategy (Duan & Keerthi 2005). In our study, we implemented SVM for multiclass-classification using OAO approach. We examined the performance of SVM as a filter and also as a classifier of efficacy-metabolite relations by using three different kernels, i.e. linear, polynomial and RBF kernels. We used R package named e1071 ver. 1.6-7 to implement the SVM method (Meyer et al. 2014).

#### 4.3.3. Random Forest

RF is an ensemble method that consists of many decision trees. Each of the classification trees is built using a bootstrap sample of the data, and at each split the candidate set of variables is a random subset of the variables (Breiman 2001; Díaz-Uriarte & De Andres 2006; Jiang et al. 2009). This method mainly includes four steps as follows: bootstrap resampling, random feature selection, full depth decision tree growing, and out-of-bag (OOB) error estimate.

Given a set of training samples  $L = \{(x_i, y_i)\}, i = 1, \dots, n$ , where  $x_i (\in \mathbb{R}^p)$  is a vector of predictor variables (in this case metabolites) and  $y_i$  the response

variable (class label), an RF targets on generating a number of  $n_{tree}$  decision trees from these samples. For each tree, the same number of  $n$  samples is randomly selected with replacement (bootstrap resampling) to form a new training set, and the samples not selected are called OOB samples. Using this new training set, a decision tree is grown to the largest extent possible without any pruning according to the classification and regression tree (CART) methodology (Duda et al. 2012). The Gini index is used during the development process of a decision tree. The Gini index at node  $v$ ,  $Gini(v)$ , is defined as:

$$Gini(v) = \sum_{c=1}^c \hat{p}_c^v (1 - \hat{p}_c^v)$$

where  $\hat{p}_c^v$  is the proportion of class  $c$  observations at node  $v$  (Deng & Runger 2013). Then, the Gini information gain of  $x_i$  for splitting node  $v$ ,  $Gain(x_i, v)$ , is the difference between the impurity at the node  $v$  and the weighted average of impurities at each child node of  $v$ . That is,

$$Gain(x_i, v) = Gini(x_i, v) - w_L Gini(x_i, v^L) - w_R Gini(x_i, v^R)$$

where  $v^L$  and  $v^R$  are the left and right child nodes of  $v$ , respectively, and  $w_L$  and  $w_R$  are the proportions of instances assigned to the left and right child nodes. At each node, a random set of  $m_{try}$  features out of  $P$  is evaluated (random feature selection), and the feature with the maximum  $Gain(x_i, v)$  is used for splitting the node  $v$ .

The random forest natively estimates an OOB error in the process of constructing the forest. With the construction of a decision tree, each OOB sample is tested, and its OOB classification result is collected. Upon the finish of constructing the entire forest, OOB classification results for each sample are used to determine a decision for this sample via a majority-voting rule. The fraction of decisions that disagree with the true class label is then the OOB error estimate.



We applied RF as a classifier in this study. We considered RF as one of selected method for classification because RF has many advantages and also outperforms other classification methods, i.e. SVM and  $k$ -Nearest Neighbor (Díaz-Uriarte & De Andres 2006). Therefore, we compared the performance of RF and SVM as classifiers to predict Jamu efficacy based on its active compounds.

#### 4.3.3.1. Regularized Random Forest

The Regularized Random Forest (RRF) is a feature selection method to select a compact variable or feature subset without loss of predictive information about  $y_i$  by building one ensemble. The features are evaluated on a part of the training data at each tree node. The RRF is built in a way similar to RF. The main difference is that the regularized information gain,  $Gain_R(x_i, v)$ , is used in the RRF (Deng & Runger 2013):

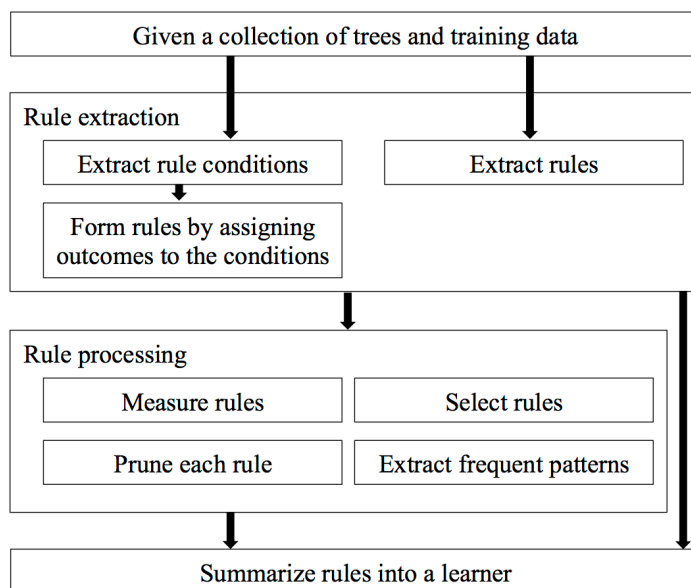
$$Gain_R(x_i, v) = \begin{cases} \lambda \cdot Gain(x_i, v), & i \notin F \\ Gain(x_i, v), & i \in F \end{cases}$$

where  $F$  is the set of indices of features used for splitting in previous nodes and also  $F$  is an empty set at the root node in the first tree. Here  $\lambda = (0,1]$  is called as penalty coefficient. When  $i \notin F$ , the coefficient penalized the  $i$ -th feature for splitting node  $v$ . A smaller  $\lambda$  leads to a larger penalty to the feature not belong to  $F$ . The RRF uses  $Gain_R(x_i, v)$  at each node, and adds the index of new feature to  $F$  if the feature adds enough predictive information to the selected features. RRF with  $\lambda = 1$  has the minimum regularization, and it is called as the least regularized subset. In this study, we also examined the performance of classifiers after selecting the features using RRF.

#### 4.3.3.2. Interpretable Trees

The Interpretable Trees (inTrees) is a new framework to interpret tree ensembles by extracting interpretable information from the resulted model, i.e. the model generated by RF and Boosted Trees (Deng 2014). Particularly, the inTrees framework can extract, measure, prune, select rules from a tree ensemble,

and calculates frequent variable interactions (Fig. 4.3). In addition, the rules from a tree ensemble can be used to build a rule-based learner for future predictions. The inTrees framework is independent from the tree assemble building process. Therefore, inTrees algorithm can be applied as long as each tree in a tree ensemble can be transformed to a specific format.



**Figure 4.3.** *Illustration of the inTrees framework.*

In this study, we implemented rule extraction, measurement and pruning to obtain important metabolites for each efficacy group from selected Jamu model. The rules were extracted from multiple decision trees generated by RF and Jamu data. The extraction process will start from a decision tree's root node to a leaf node. The rules extracted from a tree ensemble are a combination of rules extracted each decision tree in the tree ensemble (rule extraction in Appendix D). The quality of a rule is measured by frequency, error rate and length. Frequency is defined as the proportion of data instances satisfying the rule condition, whereas error of a rule is defined as the number of incorrectly classified instances determined by the rule divided by the number of instances satisfying the rule

condition for classification problems. Length is the number of predictor variable-value pairs in the condition, and it is used to measure the complexity of a rule. In case there are two rules with similar frequency and error, the rule with a smaller length is preferred because it is more interpretable. Then, the resulted rules will be pruned by leave-one-out approach to eliminate irrelevant or redundant variable-value pairs of a rule. The inTrees algorithm will calculate the relative increase of error after removing a variable-value pair  $i$  ( $decay_i$ , rule pruning in Appendix D). When  $decay_i$  is smaller than a threshold, e.g. 0.05, the  $i$ -th variable-value pair is considered as unimportant for the rule and thus can be removed. After we obtained a set of pruned rules, we defined thresholds based on error and frequency distributions to select the most important rules and also to identify the important metabolites.

#### 4.4. Results and discussion

We started our experiment by filtering the efficacy-metabolite data using multiclass SVMs with three different kernels. Subsequently, we compared the performance of classifiers to predict Jamu efficacy based on its active compounds using original Jamu data (without filtering), filtered data, and also a combination of filtered data with selected features. After we confirmed the prediction results, we extended our analysis by determining important metabolites in Jamu medicines.

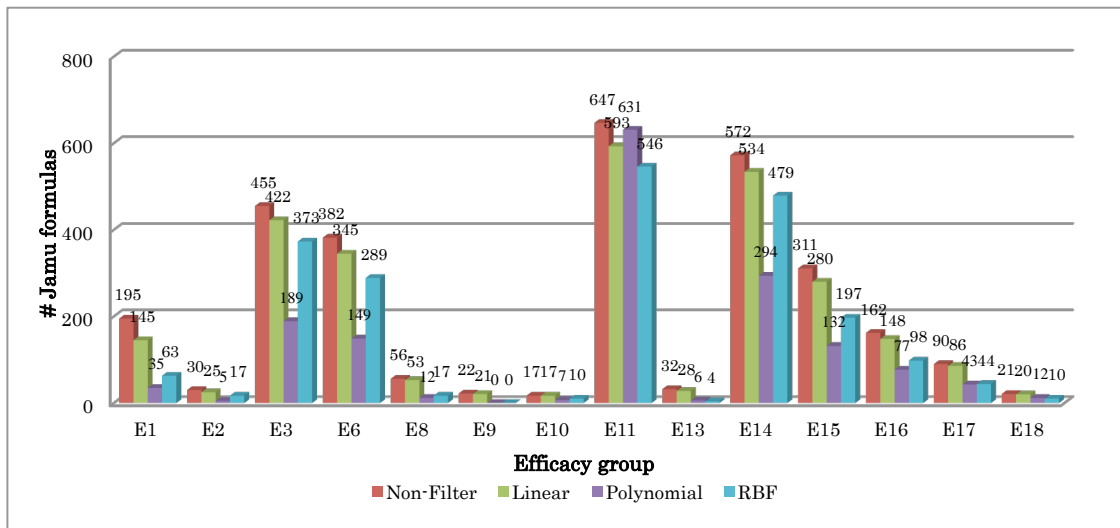
##### 4.4.1. Filtering efficacy-metabolite data

In order to create filtered datasets, 2,992 Jamu formulas were used as an input and multiclass SVMs OAO as a filter. We started our analysis by identifying the optimal parameter values for each SVM kernel using best.tune function. We obtained the same regulation parameter  $C = 1$  for all kernels. In case of polynomial and RBF kernels, we used  $d = 3$  and  $\gamma = 0.000286533$ , respectively. Then, we identified and eliminated inconsistent Jamu formulas by applying

multiclass SVMs OAO with three different kernels. All Jamu formulas were used as training and testing datasets. Here, we only selected Jamu formulas, which were correctly classified as an output of filtering process. Then, we obtained three new filtered datasets, i.e. JamuL, JamuP, and JamuR, which were generated by Jamu data and multiclass SVMs OAO with linear kernel, polynomial kernel and also RBF kernel filters, respectively (Table 4.3). Fig. 4.4 shows the distribution of Jamu formulas for each efficacy group after SF algorithm was applied.

**Table 4.3.** *The summary of filtered datasets.*

Parameters	Original (non-filter)	SVM Filter		
		Linear kernel (JamuL)	Polynomial kernel (JamuP)	RBF kernel (JamuR)
# Jamu formulas	2,992	2,717	1,592	2,147
# Efficacy groups	14	14	13	13
Cohen's Kappa		1	0	1



**Figure 4.4.** *The distribution of Jamu data. SVM classifier with three different kernels was used as Single Filtering algorithm.*

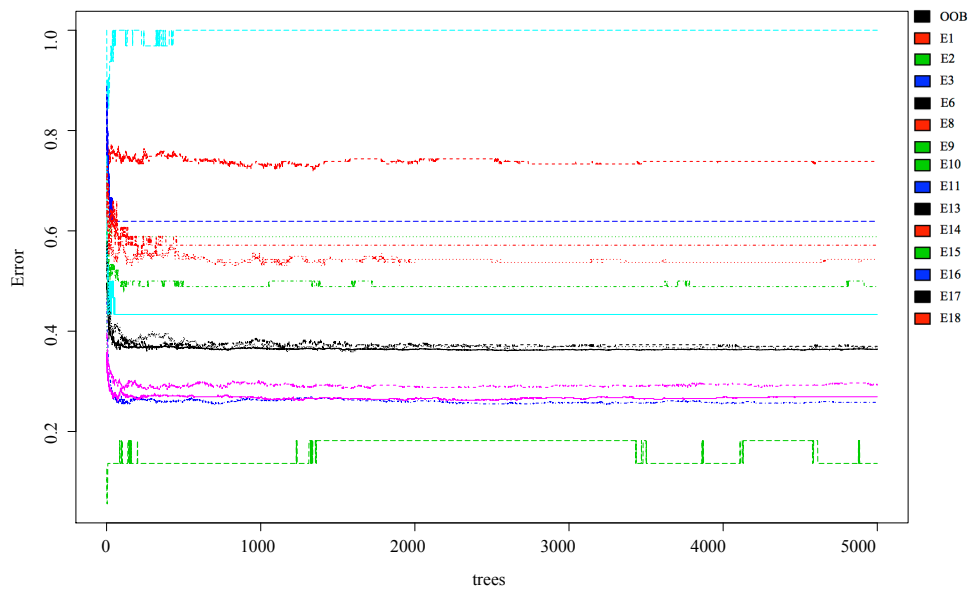
SF algorithm reduced the efficacy-metabolite relations by eliminating inconsistent Jamu data. The SVM filter with linear kernel reduced the original

efficacy-metabolite relations by 275 (9.19%), whereas filtering with RBF kernel reduced 845 (28.24%). SVM with polynomial kernel eliminated almost a half of original data, i.e. 1,400 (46.79%) Jamu formulas. In addition, the SVM polynomial and RBF filters also eliminated one efficacy group, i.e. disease of immune systems (E9). Therefore, there were only 13 classes left for JamuP and JamuR data. In case of JamuP data, most of Jamu formulas from E9 class were classified as muscle and bone class (E11), whereas it was classified as a nutritional and metabolic disease class (E14) in JamuR data. We also examined the quality of filtered datasets by utilizing Cohen's Kappa score (Ben-David 2008). We obtained the score of Cohen's Kappa from JamuP data was equal to 0. Hence, we eliminated this dataset for further analysis because JamuP data produced very poor classification result.

#### 4.4.2. Prediction of Jamu efficacy

We evaluated the effect of data filtering for predicting Jamu efficacy based on its active compounds. In this case, we compared the performance of classifiers by using non-filter dataset (original efficacy-metabolite relations) and filtered datasets (JamuL and JamuR). SVM and RF were chosen as classifiers because of their ability to analyze metabolomics data. Before we started the training process to generate a model for each classifier, we determined the best parameter values for each dataset. In case of SVM, we used the same parameter values as mentioned in Section 4.4.1. The `best.tune` function obtained the same results for each SVM kernel and dataset. In case of RF, we used `tuneRF` function from a `randomForest` package. We initially started defining the appropriate number of trees *n<sub>tree</sub>* in RF. Fig. 4.5 illustrates the relationship between error rate and the number of trees. As we can see from Fig. 4.5, the larger the number of *n<sub>tree</sub>*, the model tends to produce more consistent error rate. Consequently, it will also produce more computational time. For all datasets, we selected *n<sub>tree</sub>*=1000

because we obtained relatively constant OOB error rate for all disease classes with relatively low computational time. Next, we determined the optimal  $mtry$  value for each dataset by using the same tuning function. The  $mtry$  value for non-filter and filtered datasets (JamuL and JamuR) are 40 and 80, respectively. In term of 5-fold cross-validation, we obtained the classification accuracy by averaging accuracies produced by each subset of samples. For each classifier and dataset, we repeated the classification process 20 times before we obtained the mean accuracy over 20 iterations.



**Figure 4.5.** *The relationship between the number of trees and error rate in Random Forest tuneRF. The OOB error for all disease classes tends to be constant when  $ntree > 300$ .*

The prediction results for different datasets and also different classifiers are shown in Table 4.4. The combination of JamuR data and multiclass SVMs with linear kernel has the highest mean accuracy with 84.13%. This combination improves the accuracy 24.85% compared to the same classifier with non-filter data. In case of non-filter and JamuL datasets, RF classifier outperforms SVM with all

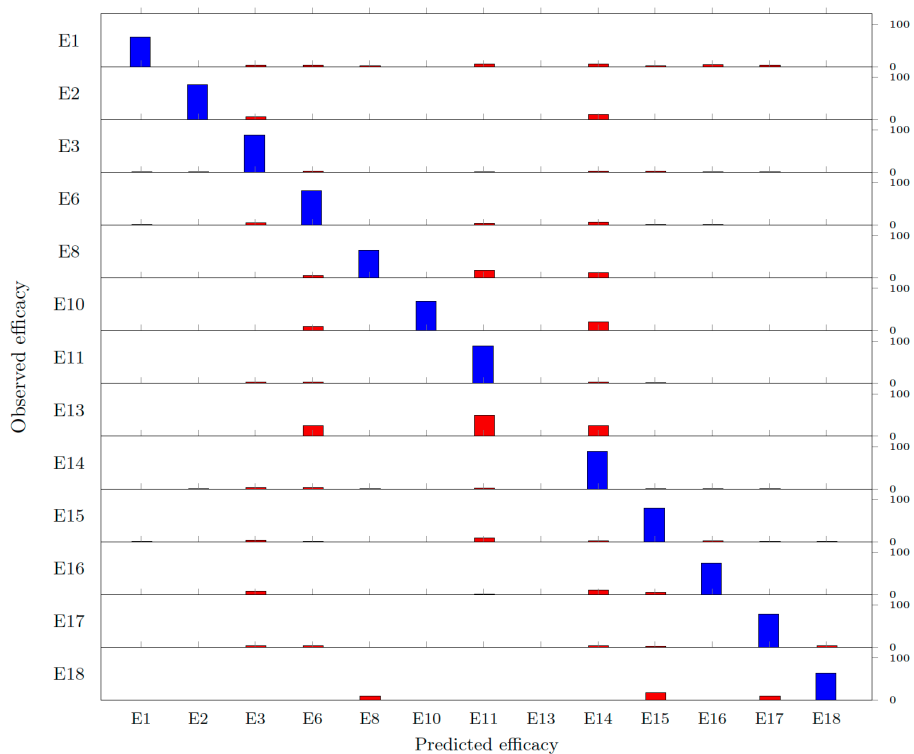
kernels. Overall, the utilization of SF algorithm improves the classification result by eliminating inconsistent Jamu formulas.

**Table 4.4.** *The classification results of non-filter and filtered Jamu data using SVM and Random Forest classifiers.*

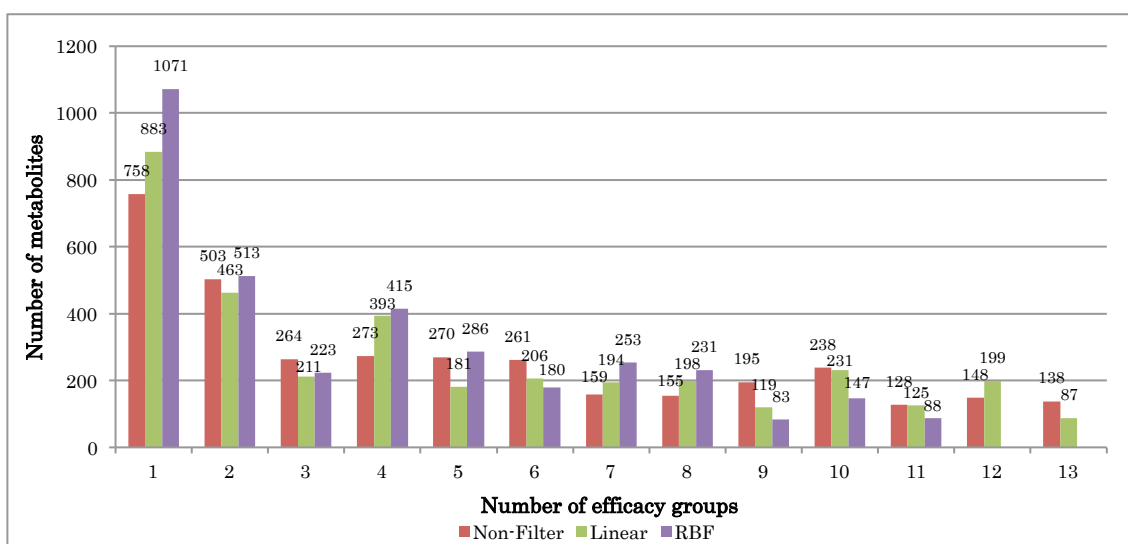
Datasets	SVM Linear		SVM Polynomial		SVM RBF		Random Forest	
	Accuracy	STD	Accuracy	STD	Accuracy	STD	Accuracy	STD
Non-filter	59.26	0.48	40.56	0.39	59.22	0.53	<b>62.04</b>	<b>0.33</b>
JamuL	68.01	0.64	43.02	0.50	64.71	0.46	<b>68.96</b>	<b>0.52</b>
JamuR	<b>84.13</b>	<b>0.29</b>	49.56	0.73	81.38	0.42	81.80	0.34

We re-ran the multiclass SVMs OAO with JamuR data to evaluate the classification results for each efficacy group. Among 2,147 Jamu formulas, the efficacy of 1,820 Jamu formulas (84.8%, Cohen’s Kappa score is 0.82) can be assigned to a correct efficacy class. Hence, the efficacy of most Jamu medicines can be predicted by utilizing its active compounds in the Jamu formula. Fig. 4.6 depicts the accuracy for each efficacy is varying from 0% for the nervous system (E13) to 89.35% for nutritional and metabolic disease (E14). The unsuccessful classification for E13 can be addressed due to a small number of Jamu for this efficacy, which is only 4 Jamu formulas (Fig. 4.4). Another source of error in the prediction of Jamu efficacy is that the active compound in the Jamu formula is not unique for a certain efficacy. As we can see in Fig. 4.7, many metabolites were effective for two or more efficacy groups.

We also applied RRF to reduce the number of features in JamuR data. As we can see from Fig. 4.8a, the number of selected features was different depending on the base coefficient of RRF ( $\lambda$ ). The number of selected features tends to increase as  $\lambda$  increases. When we set  $\lambda = 1$ , the RRF reached its minimum regularization, and it also reduced the number of features by 68.45%. Although the RRF has the least regularized subset, the selected features can obtain the same or better classification results. The average accuracy of SVM and RF classifiers for each  $\lambda$



**Figure 4.6.** Confusion matrix from the prediction of Jamu efficacy based on its metabolites using JamuR data and SVM with a linear kernel. The blue bars indicate correctly predicted Jamu and the red bars denote incorrectly predicted Jamu.



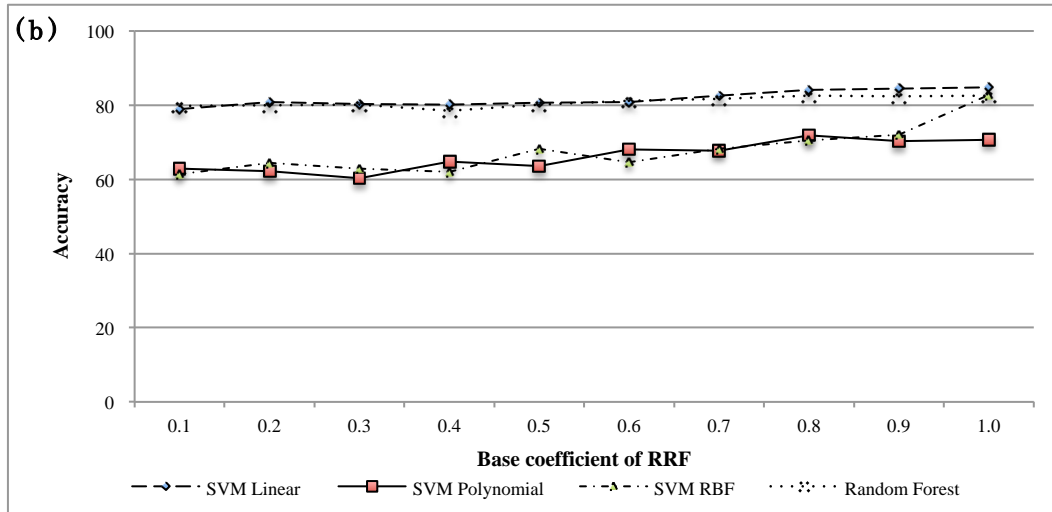
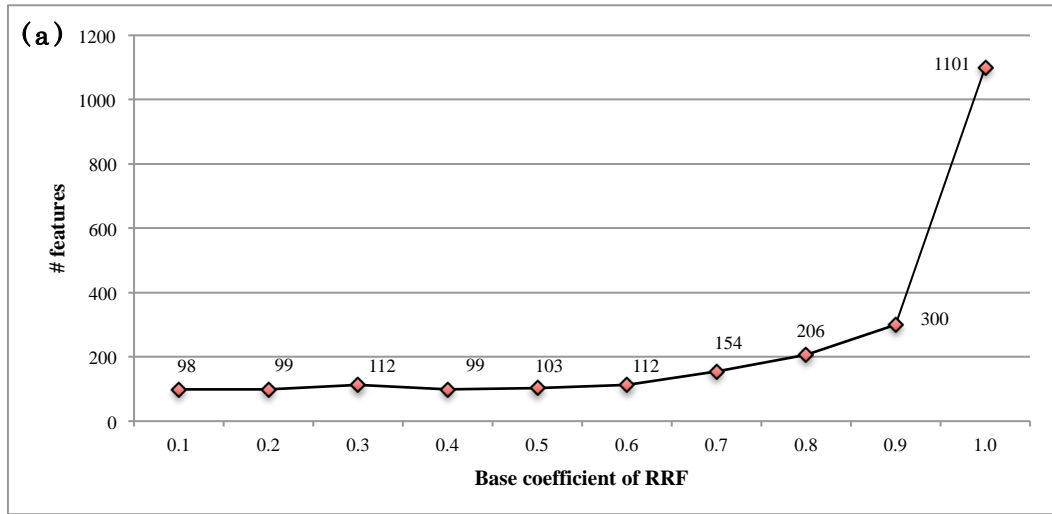
**Figure 4.7.** Distribution of 3,490 metabolites with respect to the number of efficacy groups they are assigned to.



over 20 iterations is shown in Fig. 4.8b and Table 4.5. The higher  $\lambda$ , the larger the number of selected features and also the higher accuracy obtained. In case of JamuR data and RRF, SVM with linear and RBF kernels obtained the best accuracy when  $\lambda = 1$ , whereas RF and SVM with polynomial kernel obtained the best accuracy when  $\lambda = 0.8$ . Among all combination of classifiers and datasets with RRF, SVM with a linear kernel is the best approach to predict Jamu efficacy based on its metabolites. Surprisingly, a significant result is achieved by SVM with polynomial kernel. The utilization of RRF improves the ability of SVM with polynomial kernel to classify JamuR by 22.37% compared to the approach without implementing feature selection. If we compare the performance of SVM (with linear kernel) and RF in the Jamu study, both classifiers perform well in the prediction of Jamu efficacy with the accuracy above 82%. SVM slightly outperforms RF by obtaining a higher accuracy with 2.19% difference ( $p=2.29 \times 10^{-15}$  for t-test). For all classifiers, RRF not only improves the classification accuracy but also improves interpretability by reducing the number of features.

#### 4.4.3. Identification of important metabolites

In the Section 4.4.2, we employed RF, SF algorithm, and RRF, to predict the utilization of metabolites in Jamu medicines for obtaining desired efficacy. Hence, the filtering and feature selection methods over the RF classifier can improve the classification result by eliminating inconsistent Jamu formulas and reducing the number of metabolites; that is 82.61% (Table 4.5). This result is slightly different compared to the best classification result obtained by multiclass SVMs with linear kernel. Nevertheless, one of the advantages when classifying using Random Forest classifier is its ability to interpret the rules from resulted tree ensembles. The inTrees framework can extract the rules and use the rules for building a rule-based learner or further analysis.



**Figure 4.8.** The performance of RRF for selected values of  $\lambda$ . (a) shows the number of features selected by RRF. A smaller  $\lambda$  leads to fewer features. (b) shows the mean accuracy of classifiers applied to the feature subsets selected by RRF for different  $\lambda$ . The accuracies tend to increase as  $\lambda$  increases.

**Table 4.5.** The performance of classifiers after feature selection with RRF.

Datasets	SVM Linear			SVM Polynomial			SVM RBF			Random Forest		
	$\lambda$	Accuracy	STD	$\lambda$	Accuracy	STD	$\lambda$	Accuracy	STD	$\lambda$	Accuracy	STD
JamuR + RRF	1	84.80	0.62	0.8	71.93	0.45	1	82.73	0.43	0.8	82.61	0.45

We extracted the rules from the model generated by the RF with SF algorithm and RRF using inTrees framework (Deng 2014). Initially, we re-ran our experiment using RF classifier, JamuR data and RRF with  $\lambda = 0.8$ . Then, the accuracy and Cohen's Kappa score obtained by RF model were 84.07% and 0.81, respectively. The number of extracted rules is 135,300 rules, produced by 1,000 assembled trees and 206 features (Appendix E). We reduced the resulted rules by only considering the unique rules. Therefore, the number of extracted rules was reduced from 135,000 rules to 86,670 rules. Table 4.6 indicates the distribution of extracted rules for each disease class. We examined the rules for each efficacy group by determining thresholds for error rate and frequency. In our study, a rule is categorized as a good rule if its error rate is less than or equal to a threshold and its frequency is greater than or equal to a threshold. Here, the lower error rate threshold and the higher rule's frequency indicate the better rule. The thresholds for error rate and frequency were determined by visually examined the plot between error rate and frequency and also the distribution of rule's frequency. Fig. 4.9 illustrates the thresholds for rules selection. The error thresholds for all disease classes varied from 0.01 to 0.03, whereas the frequency thresholds were in between  $1.1 \times 10^{-3}$  and  $1.89 \times 10^{-2}$ . In total, the selected rules for all disease classes are 596 rules and it is composed by 94 metabolites (Table 4.6 and Appendix F). Table 4.7 indicates the important metabolites for each disease class according to the selected rules. The length of selected rules differs from 1 to 9, and the average length is 4.64. In a rule, we determined a variable (in this case metabolite) as an important metabolite if it had a value equal to 1. Otherwise, we did not recognize this variable as an important metabolite. In the resulted rules, we can see the synergistic interaction between metabolites in the formulations of Jamu medicines (Williamson 2001). For instance, two or more metabolites should exist to obtain a desired efficacy or only a specific metabolite should be available in the Jamu-metabolite relation to attain a specific efficacy. In addition, the number of selected metabolites for each disease class also differs. For a specific type of

disease, such as cancer (E2) and heart and blood vessels (E8), the number of selected metabolites is relatively small. On the other hand, the disease classes caused by diverse factors such as a digestive system (E3), muscle and bones disease (11), and respiratory disease (E15) identify very diverse of metabolites. Further investigation to the resulted metabolites discovered that some of selected metabolites were efficacious for a disease where they were belong to, such as Germacrene B and  $\alpha$ -Humulene as anticancer (E2) (Quintans et al. 2013; Legault & Pichette 2007), catechin and epicatechin for urinary tract infections (E17) (Feliciano et al. 2015), and so on. These results can also be considered as a reduced number of compounds used in the early step of drug discovery cycle for compound screening in the process of finding a new drug against a chosen target for a particular disease. In addition, we can also utilize these results for finding alternative plants as Jamu ingredients, such as *Camellia sinensis* as an alternative for *Strobilanthes crispus* because it has the same efficacy for urinary tract infection disease (Reygaert & Jusufi 2013), and also *C. sinensis* contains catechin and epicatechin (Yang et al. 2007). Moreover, further exploration is also needed whether a combination of metabolites in a rule has a positive or negative interaction.

**Table 4.6.** *Distribution of extracted rules for each disease class.*

Class ID	Number of rules	Threshold		Number of selected rules	Number of metabolites
		Error ( $\leq$ th)	Freq. ( $\geq$ th)		
E1	6,108	0.02	0.0031	30	15
E2	419	0.03	0.0058	201	5
E3	11,094	0.02	0.0082	23	17
E6	19,231	0.01	0.0101	42	16
E8	1,074	0.03	0.0021	7	4
E10	19	0.03	0.0010	4	5
E11	16,690	0.01	0.0142	63	18
E13	570	-	-	-	-

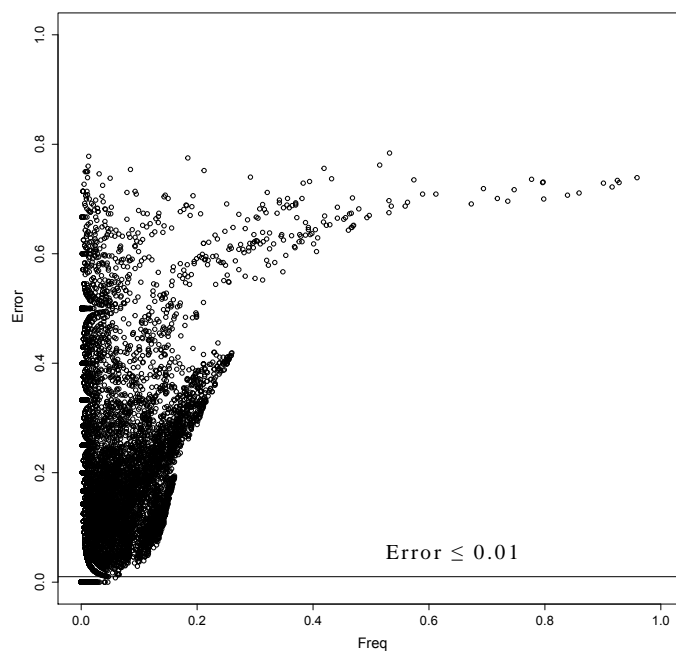
Class ID	Number of rules	Threshold		Number of selected rules	Number of metabolites
		Error ( $\leq$ th)	Freq. ( $\geq$ th)		
E14	10,095	0.01	0.0189	49	7
E15	12,896	0.01	0.0091	65	32
E16	4,904	0.02	0.0072	28	4
E17	2,973	0.02	0.0094	58	14
E18	597	0.03	0.0011	26	8
Total	86,670			596	

**Table 4.7.** List of important metabolites extracted from selected rules.

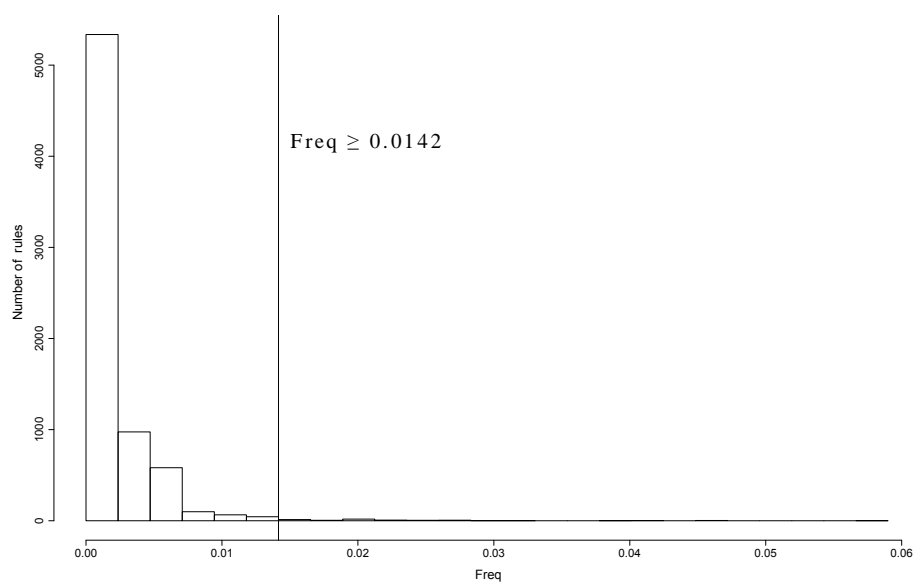
Class ID	Selected compounds (Compounds ID)
E1	(4Z)-1-(2,3,5-trihydroxy-4-methylphenyl)dec-4-en-1-one (C98), Ursolic acid (C3013), 10-Undecyn-1-ol (C326), Hentriacontane (C1852), Luteolin (C2166), beta-Caryophyllene (C3197), Tricin (C2949), beta-Amyrin (C3189), Quercitrin (C2703), Coumapherine (C1387), beta-Sitosterol (C3237), 17 <sup>^</sup> 3-Ethoxypheophorbide B (C373), Caffeic acid (C1227), Diacetoxy-[6]-gingerdiol (C1501), Gallic acid (C1734)
E2	Zingiberene (C3103), beta-Elementone (C3213), Elemol (C1585), Germacrene B (C1779), a-Humulene (C3117)
E3	(Z)-beta-Ocimene (C183), 3-Mercaptodecene (C653), Anisucumarin A (C1065), Benzaldehyde (C1126), Morin-3-O-lyxoside (C2341), Longifolene (C2158), Palmitic acid (C2496), alpha-Terpinyl acetate (C3170), alpha-Muurolene (C3154), Germacrene D (C1780), Bornyl acetate (C1189), gamma-Muurolene (C3353), Eugenol (C1672), Thujopsene (C2927), 1-epi-Cubenol (C316), alpha-Phellandrene (C3159), Geranyl acetate (C1773)
E6	Proanthocyanidin A-6 (C2619), (Z)-beta-Ocimene (C183), Curcumenol (C1418), 2-hydroxy-3',4'-dihydroxyacetophenone (C555), beta-Caryophyllene (C3197), Isocryptomerin (C1963), Fumaric acid (C1724), Geranyl acetate (C1773), Ellagic acid (C1589), Zingiberene (C3103), (-)-Epicatechin (C45), Stigmasterol (C2844), Gallic acid (C1734), alpha-Thujene (C3171), Palmitic acid (C2496), Allylpyrocatechol (C1047)
E8	Eugenol (C1672), Senkyunolide J (C2792), Diallyl sulfide (C1507), beta-Caryophyllene (C3197)

<b>Class ID</b>	<b>Selected compounds (Compounds ID)</b>
E10	1-Monolaurin (C281), Hydroxycinnamic acid (C1913), Kaempferol (C2037), Caffeic acid (C1227), Rutin (C2731)
E11	Germacrene B (C1779), [10]-Shogaol (C3106), 8-Hydroxy-9-methoxycanthin-6-one (C967), Coumapherine (C1387), beta-Caryophyllene (C3197), (Z)-beta-Ocimene (C183), Diacetoxy-[6]-gingerdiol (C1501), Dihydropiperlonguminine (C1531), 2-Phenylethanol (C539), Curzerenone (C1429), Geranyl acetate (C1773), beta-Cubebene (C3205), Ethyl-4E-octenoate (C1660), alpha-Phellandrene (C3159), Elemol (C1585), (Z)-beta-Farnesene (C182), Myristicin (C2356), delta-Cadinene (C3306)
E14	ent-Gallocatechin (C3323), Bornyl acetate (C1189), alpha-Bergamotene (C3125), Asaraldehyde (C1103), trans-Pinocarveol (C3473), Zingiberene (C3103), T-Muurolol (C2868)
E15	(Z)-beta-Ocimene (C183), Diacetoxy-[6]-gingerdiol (C1501), Nonanol (C2432), 3-Mercaptodecene (C653), Isopulegol (C1996), Elemol (C1585), a-Humulene (C3117), Benzaldehyde (C1126), [10]-Shogaol (C3106), 1-Tetradecene (C309), alpha-Terpinyol acetate (C3170), Menthol (C2237), Quercetin (C2655), Longifolene (C2158), alpha-Copaene (C3138), alpha-Thujene (C3171), Citronellal (C1347), Eugenol (C1672), Gallic acid (C1734), Linalool (C2143), Verbenone (C3031), (Z)-3-Hexenyl acetate (C166), (Z)-beta-Farnesene (C182), (E)-Anethole (C122), Bornyl acetate (C1189), Carvone (C1263), Germacrene D (C1780), Luteolin-7-methyl ether (C2185), Geraniol (C1771), Myristicin (C2356), beta-Bourbonene (C3192), Ellagic acid (C1589)
E16	Hentriacontane (C1852), alpha-Santalene (C3162), Tricin (C2949), beta-Sitosterol (C3237)
E17	Eugenol (C1672), Octacosane (C2451), (-)-Epicatechin (C45), Tetradecanal (C2906), Kaempferol (C2037), T-Muurolol (C2868), Phytol (C2568), Sinensetin (C2811), beta-Sitosterol (C3237), 2,3-Dihydrobenzofuran (C411), Caffeic acid (C1227), Apigenin (C1070), Stigmasterol (C2844), (+)-Catechin (C13)
E18	Epoxysesquithujene (C1609), Thymol (C2933), a-Humulene (C3117), beta-Bourbonene (C3192), Linalool (C2143), Myristicin (C2356), Geraniol (C1771), gamma-Terpinene (C3360)

(a)



(b)



**Figure 4.9.** Illustration of rule selection for muscle and bone class (E11). (a) shows the plot of error and frequency and (b) shows the distribution of rules based on its frequency.

#### 4.5. Summary

In order to obtain a better understanding why some Jamu formulas can be used to treat a specific disease, we performed metabolomic studies of Jamu by considering active compounds (metabolites) exist in plants as main ingredients. A thorough integration of information from omics, such as metabolomics, proteomics, transcriptomics, and genomics, is expected to provide solid evidence-based scientific rationales for the development of modern phytomedicines. This study mainly included two activities, i.e. prediction of Jamu efficacy based on its metabolites and identification of important metabolites for each efficacy group. We compared the performance of two classifiers, Support Vector Machines and Random Forest, to predict the Jamu efficacy with three different data pre-processing approaches, such as no filtering, Single Filtering algorithm, and a combination of Single Filtering algorithm and feature selection using Regularized Random Forest. Both classifiers obtained good classification results. The mean accuracy of 5-fold cross-validation obtained by SVM with linear kernel was slightly better than Random Forest, i.e. 84.60% and 82.61% respectively. Then, we extended our analysis by identifying important metabolites from the Random Forest model. The inTrees framework was used to extract the rules and to select important metabolites for each efficacy group. Overall, we identified 94 significant metabolites associated to 12 efficacy groups, which some of them were validated by published literature.



## Chapter 5

# Conclusions

In order to obtain a better understanding why some Jamu formulas can be used for curing a specific disease, we utilized data-intensive science for discovering and identifying interesting patterns of Jamu medicines. This study has been started by accumulating Jamu data from previous Jamu studies and also existing databases, i.e. KNAPSAcK Family Databases. Jamu data was obtained as a part of research collaborations between Indonesia and Japan, which is not only focus on herbal medicines but also systems agronomy. In case of herbal medicine, both countries have their own traditional medicines called as Indonesian Jamu and Japanese Kampo. The formulation of both herbal medicines is relatively the same and it is generally prepared by a combination of plants/crude drugs. In case of systems agronomy, we have collaborated in the construction of autopoietic systems for sustainable agronomy based on time scale and background culture. The data collected from this collaboration will be used to increase the formulations of herbal medicines and also the ability of conjugated enzyme concentrate (CEC) by considering ecosystems where CEC will be applied.

We explored the relationship between plant and disease in Jamu medicines by integrating network clustering and supervised learning. Jamu networks were constructed based on correlation similarity between Jamu formulas. Hence, we generated three Jamu networks with a different number of Jamu pairs for further analysis. For each Jamu formula in the Jamu network, we applied DPCLUSO to determine the potential clusters. We evaluated the clustering results using matching score to determine dominant disease for every cluster and also success rate with respect to the threshold of matching score. Here, most of the DPCLUSO

generated cluster can be confidently related to a dominant disease. Then, we empirically decided 0.6 as a matching score threshold to determine 135 predicted plants that can be assigned to diseases. The plant-disease relations predicted by our proposed method were evaluated in the context of previously published results. For all three Jamu networks, the TPR corresponding to each disease is roughly 90% or more. If we compared the list of predicted plants with previously published results, we found another 17 new plants identified as main ingredients.

In the Jamu studies, the relationships between plant, Jamu, and disease (efficacy) are represented as binary feature vectors, indicate whether a particular plant is used or not as Jamu ingredients. Therefore, a suitable binary similarity or dissimilarity equations to measure the similarity/dissimilarity between Jamu pairs is required to obtain better classification results. We proposed a method to select binary similarity and dissimilarity measures based on ROC analysis. We started our study by collecting 79 equations to measure similarity or dissimilarity between binary vectors from published literature. We eliminated 23 equations in the preliminary study because these equations were identical with other equations and also produced invalid scores while applied to all datasets. In addition, we also reduced 11 equations because they were related to other equations in the same cluster. Hence, among 79 equations used over the last century, there are only 45 equations that produce different coefficients by capturing different information. Utilization of selected binary similarity and dissimilarity measures to Jamu data found the Forbes-2 similarity was recommended in the Jamu studies. Additionally, among 14 disease classes, the equations produced good classification results on the immune system disease class. This result corresponds to our knowledge that the disease and immune system class is a very specific disease, and utilization of plant is restricted compared to other disease classes.

In addition, we extended Jamu study by including metabolites information from the plants used as Jamu ingredients for predicting Jamu efficacy and identifying important metabolites. We compared the performance of SVM and RF as classifiers, along with data pre-processing techniques such as filtering and feature selection. The SF algorithm can eliminate inconsistent Jamu formulas, whereas Regularized RF can reduce the number of features (in this case metabolites). The SVM with linear kernel and RF produced good classification results if we combined these classifiers with SF algorithm and Regularized RF, i.e. its accuracies are 84.60% and 82.61% for SVM and RF, respectively. Then, we also identified important metabolites for each efficacy group by utilizing the RF model. The inTrees framework was used to extract the rules and to select important metabolites. In total, we recognized 94 significant metabolites associated to 12 efficacy groups, which some of them were validated by published literature.

A mobile application, called as Herbal Medicine Systems, has been developed as a tool to publish our findings, to integrate and to provide information about various herbal medicines. During the development processes, the Waterfall method was chosen as a framework to build the HMS application and data warehouse pre-processing technique was used to develop HMS database. The resulted application has been published on Google Play Store and can be downloaded freely. Currently, the HMS is preloaded with Jamu and Kampo medicines. The users can update the HMS by adding new formulas or new herbal medicines, and also predict the Jamu efficacy based on plants used as Jamu ingredients.



## Bibliography

- Abe, H. et al., 1990. Systemization of semantic descriptions of odors. *Analytica Chimica Acta*, 239, pp.73–85.
- Afendi, F.M., Darusman, L.K., et al., 2012. A Bootstrapping Approach for Investigating the Consistency of Assignment of Plants to Jamu Efficacy by PLS-DA Model. *Malaysian Journal of Mathematical Sciences*, 6(2), pp.147–164.
- Afendi, F.M., Ono, N., et al., 2013. Data Mining Methods for Omics and Knowledge of Crude Medicinal Plants toward Big Data Biology Abstract : Molecular biological data has rapidly increased with the recent progress of the Omics fields , e . g . , genomics , transcriptomics , proteomics and me. *Computational and structural biotechnology journal*, 4(5).
- Afendi, F.M., Darusman, L.K., et al., 2013. Efficacy prediction of jamu formulations by PLS modeling. *Current computer-aided drug design*, 9(1), pp.46–59.
- Afendi, F.M., Okada, T., et al., 2012. KNApSACk family databases: Integrated metabolite-plant species databases for multifaceted plant research. *Plant and Cell Physiology*, 53(2), pp.e1(1–12).
- Afendi, F.M. et al., 2010. System biology approach for elucidating the relationship between Indonesian herbal plants and the efficacy of Jamu. In *Proceedings - IEEE International Conference on Data Mining, ICDM*. IEEE, pp. 661–668.
- Agrafiotis, D.K., Rassokhin, D.N. & Lobanov, V.S., 2001. Multidimensional scaling and visualization of large molecular similarity tables. *Journal of Computational Chemistry*, 22(5), pp.488–500.
- Altaf-Ul-Amin, M. et al., 2006. DPCLus : A density-periphery based graph clustering software mainly focused on detection of protein complexes in interaction networks. *Journal of Computer Aided Chemistry*, 7, pp.150–156.
- Altaf-Ul-Amin, M., Wada, M. & Kanaya, S., 2012. Partitioning a PPI network into overlapping modules constrained by high-density and periphery tracking. *ISRN Biomathematics*, 2012, pp.1–11.
- Auer, J. & Bajorath, J., 2008. Molecular similarity concepts and search calculations. In J. M. Keith, ed. *Bioinformatics volume II: Structure, function and*

- applications (Methods in molecular biology)*. Totowa, NJ: Humana Press, pp. 327–347.
- Avcibaş, I. et al., 2005. Image steganalysis with binary similarity measures. *EURASIP Journal on Applied Signal Processing*, 17, pp.2749–2757.
- Bachmaier, C., Brandes, U. & Schreiber, F., 2013. Biological networks. In R. Tamassia, ed. *Handbook of Graph Drawing and Visualization*. CRC Press, pp. 621–651.
- Barabási, A.-L. & Albert, R., 1999. Emergence of scaling in random networks. *Science*, 286(5498), pp.509–512.
- Baroni-urbani, C. & Buser, M.W., 1976. Similarity of binary data. *Systematic biology*, 25(3), pp.251–259.
- Batagelj, V. & Bren, M., 1995. Comparing resemblance measures. *Journal of Classification*, 12(1), pp.73–90.
- Ben-David, A., 2007. A lot of randomness is hiding in accuracy. *Engineering Applications of Artificial Intelligence*, 20(7), pp.875–885.
- Ben-David, A., 2008. About the relationship between ROC curves and Cohen’s kappa. *Engineering Applications of Artificial Intelligence*, 21(6), pp.874–882.
- Bien, J. & Tibshirani, R., 2011. Hierarchical clustering with prototypes via minimax linkage. *Journal of the American Statistical Association*, 106(495), pp.1075–1084.
- Bolshakova, N. & Azuaje, F., 2003. Cluster validation techniques for genome expression data. *Signal Processing*, 83(4), pp.825–833.
- Boyce, R.L. & Ellison, P.C., 2001. Choosing the best similarity index when performing fuzzy set ordination on binary data. *Journal of Vegetation Science*, 12(5), pp.711–720.
- Breiman, L., 2001. Random forests. *Machine learning*, 45(1), pp.5–32.
- Brodley, C.E. & Friedl, M.A., 1999. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11, pp.131–167.
- Burd, S.D., Jackson, R.B. & Satzinger, J.W., 2004. *Systems Analysis and Design in a Changing World*.
- Carey, V.J. et al., 2005. *Bioinformatics and computational biology solutions using R and Bioconductor* R. Gentleman, ed., New York: Springer.

- Cha, S., Choi, S. & Tappert, C., 2009. Anomaly between Jaccard and Tanimoto coefficients. In *Proceedings of Student-Faculty Research Day, CSIS, Pace University*. pp. 1–8.
- Cha, S.-H., Tappert, C.C. & Yoon, S., 2005. *Enhancing binary feature vector similarity measures*,
- Chang, J., Chen, R. & Tsai, S., 2003. Distance-preserving mappings from binary vectors to permutations. *IEEE Transactions on Information Theory*, 49(4), pp.1054–1059.
- Cheetham, A.H. et al., 1969. Binary (presence-absence) similarity coefficients. *Journal of Paleontology*, 43(5), pp.1130–1136.
- Chen, T. et al., 2013. Random forest in clinical metabolomics for phenotypic discrimination and biomarker selection. *Evidence-based Complementary and Alternative Medicine*, 2013, pp.1–11.
- Chen, X., Chen, M. & Ning, K., 2006. BNArray: An R package for constructing gene regulatory networks from microarray data by using Bayesian network. *Bioinformatics*, 22(23), pp.2952–2954.
- Choi, S.-S., Cha, S.-H. & Tappert, C.C., 2010. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics & Informatics*, 8, pp.43–48.
- Cimiano, P., Hotho, A. & Staab, S., 2004. Comparing conceptual, divisive and agglomerative clustering for learning taxonomies from text. In *Ecai 2004: Proceedings of the 16th European Conference on Artificial Intelligence*. pp. 435–439.
- Clifford, H.T. & Stephenson, W., 1975. *An Introduction to numerical classification*, Academic Press Inc., New York.
- Cohen, J., 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), pp.37–46.
- Connolly, T.M. & Begg, C.E., 2005. *Database systems: a practical approach to design, implementation, and management*, Pearson Education.
- Consonni, V. & Todeschini, R., 2012. New similarity coefficients for binary data. *Match-Communications in Mathematical and Computer Chemistry*, 68, pp.581–592.

- Dalirsefat, S.B., da Silva Meyer, A. & Mirhoseini, S.Z., 2009. Comparison of similarity coefficients used for cluster analysis with amplified fragment length polymorphism markers in the silkworm, *Bombyx mori*. *Journal of insect science (Online)*, 9(71), pp.1–8.
- Davis, J. & Goadrich, M., 2006. The relationship between Precision-Recall and ROC curves. *Proceedings of the 23rd International Conference on Machine learning -- ICML'06*, pp.233–240.
- Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7, pp.1–30.
- Deng, H., 2014. Interpreting tree ensembles with inTrees. *arXiv preprint arXiv:1408.5456*, pp.1–18.
- Deng, H. & Runger, G., 2013. Gene selection with guided regularized random forest. *Pattern Recognition*, 46(12), pp.3483–3489.
- Díaz-Uriarte, R. & De Andres, S.A., 2006. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(1), p.3.
- Dice, L.R., 1945. Measures of the amount of ecologic association between species. *Ecology*, 26(3), pp.297–302.
- Dixon, R.A. & Strack, D., 2003. Phytochemistry meets genome analysis, and beyond..... *Phytochemistry*, 62(6), pp.815–816.
- Duan, K., Rajapakse, J.C. & Nguyen, M.N., 2007. One-Versus-One and One-Versus-All Multiclass SVM-RFE for Gene Selection in Cancer Classification. In *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. pp. 47–56.
- Duan, K.-B. & Keerthi, S.S., 2005. Which Is the Best Multiclass SVM Method? An Empirical Study. *Multiple Classifier Systems*, 3541, pp.278–285.
- Duda, R.O., Hart, P.E. & Stork, D.G., 2012. *Pattern classification*, John Wiley & Sons.
- Erdős, P. & Rényi, A., 1959. On the evolution of random graphs. *Publ Math Inst Hung Acad Sci*, 5, pp.17–60.
- Ernst, E., 2003. Herbal medicines put into context: their use entails risks, but probably fewer than with synthetic drugs. *BMJ: British Medical Journal*, 327(7420), p.881.



- Faith, D.P., 1983. Asymmetric binary similarity measures. *Oecologia*, 57, pp.287–290.
- Fawcett, T., 2006. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), pp.861–874.
- Feliciano, R.P., Krueger, C.G. & Reed, J.D., 2015. Methods to determine effects of cranberry proanthocyanidins on extraintestinal infections: Relevance for urinary tract health. *Molecular Nutrition & Food Research*, 59(7), pp.1292–1306.
- Fligner, M. a, Verducci, J.S. & Blower, P.E., 2002. A modification of the Jaccard–Tanimoto similarity index for diverse selection of chemical compounds using binary strings. *Technometrics*, 44(2), pp.110–119.
- Flower, D.R., 1998. On the properties of bit string-based measures of chemical similarity. *Journal of Chemical Information and Modeling*, 38(3), pp.379–386.
- Frigui, H. & Krishnapuram, R., 1997. Clustering by competitive agglomeration. *Pattern Recognition*, 30(7), pp.1109–1119.
- Furnham, A., 1996. Why do people choose and use complementary therapies. *Complementary medicine an objective Appraisal. Edited by: Ernst E. Oxford: Butterworth-Heinemann.*
- Gelbard, R., Goldman, O. & Spiegler, I., 2007. Investigating diversity of clustering methods: an empirical comparison. *Data & Knowledge Engineering*, 63(1), pp.155–166.
- Godden, J.W., Xue, L. & Bajorath, J., 2000. Combinatorial preferences affect molecular similarity/diversity calculations using binary fingerprints and Tanimoto coefficients. *Journal of Chemical Information and Modeling*, 40(1), pp.163–166.
- Gorunescu, F., 2011. *Data Mining: Concepts, models and techniques*, Springer.
- Gower, J.C. & Legendre, P., 1986. Metric and Euclidean properties of dissimilarity coefficients. *Journal of Classification*, 3(1), pp.5–48.
- Gunn, S.R., 1998. *Support Vector Machines for classification and regression*,
- Haldar, S., 2015. *SQLite Database System Design and Implementation*, Sibsankar Haldar.
- Han, J., Kamber, M. & Pei, J., 2011. *Data mining: concepts and techniques: concepts and techniques*, Elsevier.

- Hanafi, M. et al., 2006. *Indonesian country report on traditional medicine*, New Delhi, India.
- Hillenmeyer, M., 2005. Machine Learning. Available at: <http://www.stanford.edu/~maureenh/quals/html/ml/> [Accessed January 14, 2014].
- Holliday, J.D., Hu, C.-Y. & Willett, P., 2002. Grouping of coefficients for the calculation of inter-molecular similarity and dissimilarity using 2D fragment bit-strings. *Combinatorial chemistry & high throughput screening*, 5, pp.155–166.
- Hubalek, Z., 1982. Coefficients of association and similarity, based on binary (presence-absence) data: An evaluation. *Biological Reviews*, 57, pp.669–689.
- Hussain, M. et al., 2011. A comparison of SVM kernel functions for breast cancer detection. In *Proceedings - 2011 8th International Conference on Computer Graphics, Imaging and Visualization, CGIV 2011*. pp. 145–150.
- Indonesian Food Technologist, C., 2014. Seminar nasional dan pameran industri Jamu. Available at: <http://seminar.ift.or.id/seminar-jamu-brand-indonesia/> [Accessed August 19, 2014].
- Jaccard, P., 1912. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2), pp.37–50.
- Jackson, D.A., Somers, K.M. & Harvey, H.H., 1989. Similarity coefficients: Measures of co-occurrence and association or simply measures of occurrence? *The American Naturalist*, 133(3), pp.436–453.
- Jiang, R. et al., 2009. A random forest approach to the detection of epistatic interactions in case-control studies. *BMC bioinformatics*, 10(Suppl 1), p.S65.
- Johnson, S.C., 1967. Hierarchical clustering schemes. *Psychometrika*, 32(3), pp.241–254.
- Kangas, J.D., Naik, A.W. & Murphy, R.F., 2014. Efficient discovery of responses of proteins to compounds using active learning. *BMC Bioinformatics*, 15(143), pp.1–11.
- Kedariseti, P. et al., 2014. Prediction and characterization of cyclic proteins from sequences in three domains of life. *Biochimica et Biophysica Acta - Proteins and Proteomics*, 1844(1 PART B), pp.181–190.

- Kelling, S. et al., 2009. Data-intensive science: A new paradigm for biodiversity studies. *BioScience*, 59(7), pp.613–620.
- Khalifa, M. & Verner, J.M., 2000. Drivers for software development method usage. *IEEE Transactions on Engineering Management*, 47(3), pp.360–369.
- Kosman, E. & Leonard, K.J., 2005. Similarity coefficients for molecular markers in studies of genetic relationships between individuals for haploid, diploid, and polyploid species. *Molecular Ecology*, 14(2), pp.415–424.
- Lance, G.N. & Williams, W.T., 1966. Computer Programs for Hierarchical Polythetic Classification ('Similarity Analyses'). *The Computer Journal*, 9(1), pp.60–64.
- Langfelder, P. & Horvath, S., 2008. WGCNA: an R package for weighted correlation network analysis. *BMC bioinformatics*, 9, p.559.
- Legault, J. & Pichette, A., 2007. Potentiating effect of beta-caryophyllene on anticancer activity of alpha-humulene, isocaryophyllene and paclitaxel. *The Journal of pharmacy and pharmacology*, 59(12), pp.1643–7.
- Legendre, P. & Legendre, L., 1998. *Numerical ecology: Developments in environmental modelling*, Elsevier.
- Li, M. et al., 2008. Modifying the DPCLUS algorithm for identifying protein complexes based on new topological structures. *BMC Bioinformatics*, 9(398), pp.1–16.
- Lim, T., Loh, W. & Shih, Y., 2000. A comparison of prediction accuracy, complexity, and training time of thirty three old and new classification algorithms. *Machine Learning*, 40(3), pp.203–229.
- Lindon, J.C. et al., 2003. Contemporary issues in toxicology: The role of metabonomics in toxicology and its evaluation by the COMET project. *Toxicology and Applied Pharmacology*, 187(3), pp.137–146.
- Lourenco, F., Lobo, V. & Bacao, F., 2004. *Binary-based similarity measures for categorical data and their application in Self-Organizing Maps*,
- Mahadevan, S. et al., 2008. Analysis of metabolomic data using support vector machines. *Anal Chem*, 80(19), pp.7562–7570.
- Manning, C.D. & Schütze, H., 1999. *Foundations of statistical natural language processing*, MIT press.
- Martin, A. et al., 2010. BisoGenet: a new tool for gene network building, visualization and analysis. *BMC bioinformatics*, 11(1), p.91.

- Max Planck Institut Informatik, 2013. NetworkAnalyzer. Available at: <http://med.bioinf.mpi-inf.mpg.de/netanalyzer/index.php> [Accessed May 20, 2016].
- Metz, C.E., 1978. Basic principles of ROC analysis. *Seminars in nuclear medicine*, 8(4), pp.283–298.
- Meyer, D. et al., 2014. e1071: Misc Functions of the Department of Statistics (e1071), TU Wien. , pp.1–62.
- Michael, E.L., 1920. Marine ecology and the coefficient of association: A plea in behalf of quantitative biology. *Journal of Ecology*, 8(1), pp.54–59.
- National Center for Biotechnology Information, 1998. Genes and diseases. *National Center for Biotechnology Information (US), Bethesda (MD)*. Available at: <http://www.ncbi.nlm.nih.gov/books/NBK22185/> [Accessed May 20, 2016].
- Nei, M. & Li, W.-H., 1979. Mathematical model for studying genetic variation in terms of restriction endonucleases. *Proceedings of the National Academy of Sciences of the United States of America*, 76(10), pp.5269–5273.
- Nidhi et al., 2006. Prediction of biological targets for compounds using multiple-category bayesian models trained on chemogenomics databases. *Journal of Chemical Information and Modeling*, 46(3), pp.1124–1133.
- Ohtana, Y. et al., 2014. Clustering of 3D-structure similarity based network of secondary metabolites reveals their relationships with biological activities. *Molecular Informatics*, 33, pp.790–801.
- Ojurongbe, T.A., 2012. *Comparison of different proximity measures and classification methods for binary data*. Justus Liebig University Gießen.
- Okada, T. et al., 2016. Informatics framework of traditional Sino-Japanese medicine (Kampo) unveiled by factor analysis. *Journal of Natural Medicines*, 70(1), pp.107–114.
- Pinoli, P., Chicco, D. & Masseroli, M., 2015. Computational algorithms to predict Gene Ontology annotations. *BMC Bioinformatics*, 16(Suppl 6), pp.1–15.
- Pramono, S., 2007. *Jamu in Indonesian daily life and industry*, Toyama: Institute of Natural Medicine, University of Toyama.
- Quintans, J.S.S. et al., 2013. Chemical constituents and anticancer effects of the essential oil from leaves of *Xylocopa laevigata*. *Planta Medica*, 79(2), pp.123–130.

- Reygaert, W. & Jusufi, I., 2013. Green tea as an effective antimicrobial for urinary tract infections caused by *Escherichia coli*. *Frontiers in Microbiology*, 4(JUN), pp.1–4.
- Rodgers, J.L. & Nicewander, W.A., 1988. Thirteen Ways to Look at the Correlation Coefficient. *The American Statistician*, 42(1), pp.59 – 66.
- Rojas-Cherto, M. et al., 2012. Metabolite identification using automated comparison of high-resolution multistage mass spectral trees. *Analytical chemistry*, 84(13), pp.5524–5534.
- Romero, R. et al., 2006. The use of high-dimensional biology (genomics, transcriptomics, proteomics, and metabolomics) to understand the preterm parturition syndrome. *BJOG: An International Journal of Obstetrics and Gynaecology*, 113(SUPPL. 3), pp.118–135.
- Sam Kash Kachigan, 1991. *Multivariate Statistical Analysis: A Conceptual Introduction*, New York, NY, USA: Radius Press.
- Schippmann, U., Leaman, D. & Cunningham, A., 2006. A comparison of cultivation and wild collection of medicinal and aromatic plants under sustainability aspects. In *Medicinal and aromatic plants*. pp. 75–95.
- Schippmann, U., Leaman, D.J. & Cunningham, A.B., 2002. Impact of cultivation and gathering of medicinal plants on biodiversity: global trends and issues. *Biodiversity and the ecosystem approach in agriculture, forestry and fisheries*, (January), pp.1–21.
- Schirle, M. & Jenkins, J.L., 2016. Identifying compound efficacy targets in phenotypic drug discovery. *Drug Discovery Today*, 21(1), pp.82–89.
- Schüldt, C., Laptev, I. & Caputo, B., 2004. Recognizing human actions: A local SVM approach. In *Proceedings of the 17th International Conference on Pattern Recognition*. pp. 32–36.
- Shannon, P. et al., 2003. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11), pp.2498–2504.
- Shyur, L.F. & Yang, N.S., 2008. Metabolomics for phytomedicine research and drug development. *Current Opinion in Chemical Biology*, 12(1), pp.66–71.
- da Silva Meyer, A. et al., 2004. Comparison of similarity coefficients used for cluster analysis with dominant markers in maize (*Zea mays* L). *Genetics and Molecular Biology*, 27(1), pp.83–91.

- Sing, T. et al., 2005. ROCR: Visualizing classifier performance in R. *Bioinformatics*, 21(20), pp.3940–3941.
- Sommerville, I., 2010. *Software Engineering* 9th ed., Pearson.
- Sonego, P., Kocsor, A. & Pongor, S., 2008. ROC analysis: Applications to the classification of biological sequences and 3D structures. *Briefings in Bioinformatics*, 9(3), pp.198–209.
- Stiles, H.E., 1961. The association factor in information retrieval. *Journal of the ACM (JACM)*, 8(2), pp.271–279.
- Tibshirani, R. et al., 2004. Sample classification from protein mass spectrometry, by “peak probability contrasts.” *Bioinformatics*, 20(17), pp.3034–3044.
- Todeschini, R. et al., 2012. Similarity coefficients for binary chemoinformatics data: Overview and extended comparison using simulated and real data sets. *Journal of Chemical Information and Modeling*, 52(11), pp.2884–2901.
- Tweeddale, H., Notley-McRobb, L. & Ferenci, T., 1998. Effect of slow growth on metabolism of *Escherichia coli*, as revealed by global metabolite pool (“metabolome”) analysis. *Journal of bacteriology*, 180(19), pp.5109–5116.
- Vapnik, V., 1998. Statistical Learning Theory. *Adaptive and learning Systems for Signal Processing, Communications and Control*, pp.1–740.
- Varpoorte, R., Kim, H. & Choi, Y., 2006. Plants as source of medicines: new perspectives. *Medicinal and aromatic plants. Springer, Netherlands*, pp.261–273.
- Vázquez, A., 2003. Growing network with local rules: preferential attachment, clustering hierarchy, and degree correlations. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 67(5 Pt 2), p.056104.
- Warrens, M.J., 2008. *Similarity coefficients for binary data: properties of coefficients, coefficient matrices, multi-way metrics and multivariate coefficients*. Leiden University.
- Weber, T. & Kim, H.U., 2016. The secondary metabolite bioinformatics portal: Computational tools to facilitate synthetic biology of secondary metabolite production. *Synthetic and Systems Biotechnology*, 1(2), pp.69–79.
- Weisberg, S., 2005. *Applied linear regression*,

- Wijaya, S.H. et al., 2014. Supervised clustering based on DPCLUSO: Prediction of plant-disease relations using Jamu formulas of KNApSAcK database. *BioMed Research International*, 2014, pp.1–15.
- Willett, P., Barnard, J.M. & Downs, G.M., 1998. Chemical similarity searching. *Journal of Chemical Information and Modeling*, 38(6), pp.983–996.
- Williamson, E.M., 2001. Synergy and other interactions in phytomedicines. *Phytomedicine*, 8(5), pp.401–409.
- Winterbach, W. et al., 2013. Topology of molecular interaction networks. *BMC systems biology*.
- World Health Organization, 2010. International Classification of Diseases (ICD) 10. Available at: <http://www.who.int/classifications/icd/en/> [Accessed May 20, 2016].
- Yang, X.R. et al., 2007. Simultaneous analysis of purine alkaloids and catechins in *Camellia sinensis*, *Camellia ptilophylla* and *Camellia assamica* var. *kucha* by HPLC. *Food Chemistry*, 100(3), pp.1132–1136.
- Zapata, B.C., 2013. *Android Studio Application Development*, Packt Publishing Ltd.
- Zapata, B.C., 2015. *Android Studio Essentials*, Packt Publishing Ltd.
- Zhang, B. & Srihari, S.N., 2003. Binary vector dissimilarity measures for handwriting identification. In *Proceedings of SPIE-IS&T Electronic Imaging Vol. 5010*. pp. 28–38.
- Zhang, B. & Srihari, S.N., 2003. Properties of binary vector dissimilarity measures. In *Proc. JCIS Int'l Conf. Computer Vision, Pattern Recognition, and Image Processing*. pp. 1–4.
- Zhou, T. et al., 2015. Quantitative modeling of transcription factor binding specificities using DNA shape. *Proceedings of the National Academy of Sciences*, 112(15), pp.4654–4659.
- Zhou, X. & Tuck, D.P., 2007. MSVM-RFE: Extensions of SVM-RFE for multiclass gene selection on DNA microarray data. *Bioinformatics*, 23(9), pp.1106–1114.





## Achievements

### Reviewed publications

1. **Sony Hartono Wijaya**, Husnawati Husnawati, Farit Mochamad Afendi, Irmanida Batubara, Latifah K. Darusman, Md. Altaf-Ul-Amin, Tetsuo Sato, Naoaki Ono, Tadao Sugiura, and Shigehiko Kanaya, 2014, Supervised clustering based on DPCLUSO: Prediction of plant-disease relations using Jamu Formulas of KNAPSAcK Database, *BioMed Research International* 2014:1-15 doi:10.1155/2014/831751 (in Chapter 2).
2. **Sony Hartono Wijaya**, Yuki Tanaka, Md. Altaf-Ul-Amin, Aki Hirai Morita, Farit Mochamad Afendi, Irmanida Batubara, Naoaki Ono, Latifah K. Darusman, and Shigehiko Kanaya, 2016, Utilization of KNAPSAcK family databases for developing Herbal Medicine Systems, *Journal of Computer Aided Chemistry* 17:1-7, ISSN 1345-8647 (in Chapter 2).
3. **Sony Hartono Wijaya**, Diky Indrawibawa, Sony Suharsono, Kenzo Nakamura, Keichi Nishijima, Hideshige Takagishi, Keiichi Takeuchi, Ryohei Eguchi, and Shigehiko Kanaya, 2016, Systems agronomy collaboration between Japanese and Indonesian agriculture industries focus on potato cultivation, *CICSJ Bulletin* Vol. 34 No 2:53-58, ISSN 1347-2283 (in Chapter 5).

### International conferences

1. **Sony Hartono Wijaya**, Shinya Muraoka, Lidwina Andarini, Farit Mochamad Afendi, Aki Hirai Morita, Latifah K. Darusman, Md. Altaf-Ul-Amin, Tetsuo Sato, Naoaki Ono, Tadao Sugiura, and Shigehiko Kanaya, 2014, HerbsMed: Herbal medicine application using integrated Jamu and Kampo formulas, The

- 10<sup>th</sup> Annual International Conference of the Metabolomics Society (June 23-26 2014, Daiichi Hotel Tsuruoka, Tsuruoka, Japan).
2. Yuki Tanaka, **Sony Hartono Wijaya**, Naoaki Ono, Md. Altaf-Ul-Amin, and Shigehiko Kanaya, 2015, Prediction of Jamu efficacies based on its ingredients using Naive Bayes and Random Forest classifiers, The 3<sup>rd</sup> International Symposium on Temulawak (September 2-5, 2015, Bogor Agricultural University, Bogor, Indonesia).
  3. Md. Altaf-Ul-Amin, Dodi Fitra Chandra, **Sony Hartono Wijaya**, and Shigehiko Kanaya, 2015, Relation of essentiality and functionality of Yeast protein with their centrality values in a PPI network, The 27<sup>th</sup> International Conference on Yeast Genetics and Molecular Biology, ICYGMB 2015 (September 6-12 2015, PalaLevico Valsugana Fiere, Levico Terme, Italy).

## Appendices



## Appendix A

### List of diseases from ICD-10

The list of diseases that comply Jamu efficacy. Disease class corresponds to class disease IDs in Table 2.2.

ID	Disease	Disease class	ID	Disease	Disease class
1	Abdominal Pain	3	20	Common Cold	15
2	Abdominal Pain, Diarrhea	3	21	Common Cold, Dyspepsia, Insect Bites	15, 3, 16
3	Acne	16	22	Common Cold, Influenza	15
4	Acne, Skin Problems (Cosmetics)	16	23	Cough	15
5	Amenorrhoea, Dysmenorrhea	6	24	Degenerative Disease	14
6	Amenorrhoea, Irregular Menstruation	6	25	Dermatitis, Urticaria, Erythema	16
7	Anaemia	1	26	Diabetes	14
8	Appendicitis, Urinary Tract Infection, Tonsillitis	3	27	Diarrhea	3
9	Arthralgia	11	28	Diarrhea, Abdominal Pain	3
10	Arthralgia, Arthritis	11	29	Diseases Of The Eye	5
11	Asthma	15	30	Disorders In Pregnancy	6
12	Benign Prostatic Hyperplasia (Bph)	10	31	Dysmenorrhea	6
13	Breast Disorder	6	32	Dysmenorrhea, Irregular Menstruation	6
14	Bromhidrosis	16	33	Dysmenorrhea, Menstrual Syndrome	6
15	Bronchitis	15	34	Dyspepsia	3
16	Cancer	2	35	Dyspnoea	15
17	Cancer Pain	2	36	Dyspnoea, Cough, Orthopnoea	15
18	Cancer, Inflammation	2	37	Fatigue	11
19	Colic Abdomen, Bloating (In Infant)	3	38	Fatigue, Anaemia, Loss Appetite	1
			39	Fatigue, Lack Of Sexual Function	6

<b>ID</b>	<b>Disease</b>	<b>Disease class</b>	<b>ID</b>	<b>Disease</b>	<b>Disease class</b>
40	Fatigue, Low Back Pain	11	67	Low Back Pain, Myalgia,	11
41	Fatigue, Myalgia, Arthralgia	11		Arthralgia	
42	Fatigue, Osteoarthritis	11	68	Low Back Pain, Myalgia,	11
43	Fertility Problem	6, 10		Constipation	
44	Fever	0	69	Low Back Pain, Urinary Tract	17
45	Gastritis, Gastric Ulcer	3		Infection	
46	Haemorrhoids	1	70	Lung Diseases	15
47	Headache	13	71	Malaise & Fatigue	11
48	Heart Diseases	8	72	Malaise & Fatigue, Constipation	11
49	Heartburn	3, 8	73	Malaise & Fatigue, Fertility	10, 11
50	Hepatitis, Other Diseases Of Liver	3		Problems	
51	Hypercholesterolaemia	14	74	Malaise & Fatigue, Low Back Pain	11
52	Hypertension	8	75	Malaise & Fatigue, Sexual	11, 6, 10
53	Hypertension, Diabetes	14		Dysfunction	
54	Hypertension, Hypercholesterolaemia	14	76	Malaise & Fatigue, Skin Problems (Cosmetics)	16
55	Hyperuricemia	1	77	Malaria, Anaemia	1
56	Immunodeficiency	9	78	Meno-Metrorrhagia	6
57	Indigestion (K.30)	3	79	Menopausal Syndrome	6
58	Indigestion, Lose Appetite	3	80	Menopause/Menstrual	6
59	Infertility	6, 10		Syndrome, Leukorrhoea	
60	Irregular Menstruation, Menstruation Syndrome	6		(Vaginalis)	
61	Kidney Diseases	17	81	Menstrual Syndrome	6
62	Lactation Problems	6	82	Menstrual Syndrome, Fatigue	6
63	Leukorrhoea (Vaginalis)	6	83	Migraine	13
64	Leukorrhoea (Vaginalis), Dysmenorrhoea	6	84	Mood Disorder	18
65	Lose Appetite	3	85	Myalgia, Arthralgia	11
66	Lose Appetite, Underweight	14	86	Nausea/Vomiting Of Pregnancy	6
			87	Osteoarthritis	11
			88	Osteoarthritis, Fatigue	11

<b>ID</b>	<b>Disease</b>	<b>Disease class</b>	<b>ID</b>	<b>Disease</b>	<b>Disease class</b>
89	Overweight, Obesity	14	104	Stomatitis, Gingivitis, Tonsilitis	3
90	Paralysis	13	105	Stone In Kidney (N20.0)	17
91	Post Partum Syndrome	6	106	Stone In Kidney (N20.0), Urinary Bladder Stone (N21.0)	17
92	Prevent From Overweight	14	107	Tonsilitis	4
93	Respiratory Infection Due To Smoking	15	108	Tonsilofaringitis	4
94	Respiratory Tract Infection	15	109	Toothache	13
95	Rheumatoid Arthritis, Gout	11	110	Typhoid, Dyspepsia	3
96	Secondary Amenorrhea	6	111	Ulcer Of Anus And Rectum	3
97	Secondary Amenorrhea, Irregular Menstruation	6	112	Ulkus Diabetikum, Diabetic Gangrene	16
98	Sexual Dysfunction, Fatigue	6, 10	113	Underweight, Lose Appetite	3
99	Skin Diseases	16	114	Urinary Tract Infection (Urethritis)	17
100	Skin Problems (Cosmetics)	16	115	Vaginal Discharges	6
101	Sleeping & Mood Disorders	18	116	Vaginal Diseases	6
102	Sleeping Disorders	18			
103	Stomatitis	3			





## Appendix B

### List of plants assigned to each disease

The relationship between plant and disease. The asterisk (\*) indicates that plant will not be assigned if we use matching score  $> 0.7$ .

No	Plants Name	Hit / Miss Status	No	Plants Name	Hit / Miss Status
<b>A. Disease: Blood and Lymph Diseases</b>			22	<i>Sida rhombifolia</i>	Miss
1	<i>Tamarindus indica</i>	Hit *	23	<i>Cyperus rotundus</i>	Hit
2	<i>Allium sativum</i>	Hit *	24	<i>Sonchus arvensis</i>	Miss
3	<i>Tinospora tuberculata</i>	Hit *	25	<i>Curcuma aeruginosa</i>	Hit *
4	<i>Piper retrofractum</i>	Hit	26	<i>Curcuma xanthorrhiza</i>	Hit
5	<i>Syzygium aromaticum</i>	Hit *			
6	<i>Bupleurum falcatum</i>	Hit	<b>B. Disease: Cancers</b>		
7	<i>Graptophyllum pictum</i>	Hit	1	<i>Catharanthus roseus</i>	Hit
8	<i>Plantago major</i>	Hit	<b>C. Disease: The Digestive System</b>		
9	<i>Zingiber officinale</i>	Hit *	1	<i>Foeniculum vulgare</i>	Hit
10	<i>Cinnamomum burmani</i>	Hit *	2	<i>Glycyrrhiza uralensis</i>	Hit *
11	<i>Soya max</i>	Miss *	3	<i>Imperata cylindrica</i>	Hit
12	<i>Kaempferia galanga</i>	Hit	4	<i>Zingiber purpureum</i>	Hit *
13	<i>Curcuma longa</i>	Hit *	5	<i>Physalis peruviana</i>	Hit
14	<i>Piper nigrum</i>	Hit	6	<i>Punica granatum</i>	Hit *
15	<i>Zingiber aromaticum</i>	Hit *	7	<i>Echinacea purpurea</i>	Hit
16	<i>Phyllanthus urinaria</i>	Hit *	8	<i>Zingiber officinale</i>	Hit *
17	<i>Oryza sativa</i>	Hit	9	<i>Psidium guajava</i>	Hit
18	<i>Myristica fragrans</i>	Hit *	10	<i>Baeckea frutescens</i>	Hit *
19	<i>Alstonia scholaris</i>	Hit *	11	<i>Amomum compactum</i>	Hit
20	<i>Syzygium polyanthum</i>	Miss	12	<i>Cinnamomum burmani</i>	Hit *
21	<i>Andrographis paniculata</i>	Hit *			

No	Plants Name	Hit / Miss Status	No	Plants Name	Hit / Miss Status
13	<i>Melaleuca leucadendra</i>	Hit	41	<i>Matricaria chamomilla</i>	Hit *
14	<i>Caesalpinia sappan</i>	Hit *	42	<i>Cymbopogon nardus</i>	Hit *
15	<i>Parkia roxburghii</i>	Hit	<b>D. Disease: Female-Specific Diseases</b>		
16	<i>Rheum tanguticum</i>	Hit	1	<i>Foeniculum vulgare</i>	Hit
17	<i>Kaempferia galanga</i>	Hit	2	<i>Imperata cylindrica</i>	Hit
18	<i>Coriandrum sativum</i>	Hit	3	<i>Tamarindus indica</i>	Hit
19	<i>Curcuma longa</i>	Hit	4	<i>Pluchea indica</i>	Hit *
20	<i>Zingiber aromaticum</i>	Hit	5	<i>Piper retrofractum</i>	Hit
21	<i>Phyllanthus urinaria</i>	Hit	6	<i>Punica granatum</i>	Hit
22	<i>Myristica fragrans</i>	Hit	7	<i>Uncaria rhynchophylla</i>	Hit
23	<i>Hydrocotyle asiatica</i>	Hit *	8	<i>Zingiber officinale</i>	Hit
24	<i>Carica papaya</i>	Hit	9	<i>Guazuma ulmifolia</i>	Hit *
25	<i>Mentha arvensis</i>	Hit	10	<i>Nigella sativa</i>	Hit
26	<i>Lepiniopsis ternatensis</i>	Hit	11	<i>Terminalia bellirica</i>	Hit
27	<i>Helicteres isora</i>	Hit	12	<i>Baeckea frutescens</i>	Hit
28	<i>Andrographis paniculata</i>	Hit	13	<i>Phaseolus radiatus</i>	Hit
29	<i>Symplocos odoratissima</i>	Hit	14	<i>Amomum compactum</i>	Hit *
30	<i>Schisandra chinensis</i>	Hit	15	<i>Sauropus androgynus</i>	Hit
31	<i>Blumea balsamifera</i>	Hit	16	<i>Usnea misaminensis</i>	Hit
32	<i>Silybum marianum</i>	Hit *	17	<i>Cinnamomum burmani</i>	Hit
33	<i>Cinnamomum sintok</i>	Hit	18	<i>Melaleuca leucadendra</i>	Hit
34	<i>Elephantopus scaber</i>	Hit	19	<i>Parameria laevigata</i>	Hit
35	<i>Curcuma aeruginosa</i>	Hit	20	<i>Parkia roxburghii</i>	Hit
36	<i>Kaempferia pandurata</i>	Hit	21	<i>Piper cubeba</i>	Hit
37	<i>Curcuma xanthorrhiza</i>	Hit	22	<i>Kaempferia galanga</i>	Hit
38	<i>Curcuma mangga</i>	Hit *	23	<i>Coriandrum sativum</i>	Hit
39	<i>Curcuma zedoaria</i>	Hit	24	<i>Kaempferia angustifolia</i>	Hit
40	<i>Daucus carota</i>	Hit *			

No	Plants Name	Hit / Miss Status	No	Plants Name	Hit / Miss Status
25	<i>Curcuma longa</i>	Hit	<b>F. Disease: Male-Specific Diseases</b>		
26	<i>Zingiber aromaticum</i>	Hit	1	<i>Cucurbita pepo</i>	Miss
27	<i>Languas galanga</i>	Hit	2	<i>Serenoa repens</i>	Miss
28	<i>Galla lusitania</i>	Hit	3	<i>Baeckea frutescens</i>	Hit
29	<i>Quercus lusitanica</i>	Hit	4	<i>Phaseolus radiatus</i>	Hit
30	<i>Hydrocotyle asiatica</i>	Hit	5	<i>Curcuma longa</i>	Hit
31	<i>Areca catechu</i>	Hit	6	<i>Elephantopus scaber</i>	Hit
32	<i>Lepiniopsis ternatensis</i>	Hit	<b>G. Disease: Muscle and Bone</b>		
33	<i>Helicteres isora</i>	Hit *	1	<i>Foeniculum vulgare</i>	Hit
34	<i>Piper betle</i>	Hit	2	<i>Clausena anisum-olens</i>	Hit *
35	<i>Elephantopus scaber</i>	Hit *	3	<i>Zingiber purpureum</i>	Hit
36	<i>Kaempferia pandurata</i>	Hit	4	<i>Allium sativum</i>	Hit
37	<i>Curcuma xanthorrhiza</i>	Hit	5	<i>Strychnos ligustrina</i>	Hit
38	<i>Sesbania grandiflora</i>	Hit	6	<i>Tinospora tuberculata</i>	Hit *
<b>E. Disease: The Heart and Blood Vessels</b>			7	<i>Piper retrofractum</i>	Hit
1	<i>Allium sativum</i>	Hit	8	<i>Syzygium aromaticum</i>	Hit
2	<i>Curcuma longa</i>	Hit *	9	<i>Cola nitida</i>	Hit *
3	<i>Morinda citrifolia</i>	Hit *	10	<i>Ginkgo biloba</i>	Hit *
4	<i>Homalomena occulta</i>	Hit *	11	<i>Panax ginseng</i>	Hit
5	<i>Hydrocotyle asiatica</i>	Hit	12	<i>Equisetum debile</i>	Hit *
6	<i>Alstonia scholaris</i>	Hit *	13	<i>Zingiber officinale</i>	Hit
7	<i>Syzygium polyanthum</i>	Miss *	14	<i>Ganoderma lucidum</i>	Hit
8	<i>Andrographis paniculata</i>	Hit *	15	<i>Nigella sativa</i>	Hit
9	<i>Apium graveolens</i>	Miss	16	<i>Terminalia bellirica</i>	Hit *
10	<i>Imperata cylindrica</i>	Hit	17	<i>Baeckea frutescens</i>	Hit *
			18	<i>Amomum compactum</i>	Hit
			19	<i>Cinnamomum burmani</i>	Hit

No	Plants Name	Hit / Miss Status	No	Plants Name	Hit / Miss Status
20	<i>Melaleuca leucadendra</i>	Hit	48	<i>Mentha arvensis</i>	Hit *
21	<i>Parameria laevigata</i>	Hit *	49	<i>Lepiniopsis ternatensis</i>	Hit
22	<i>Psophocarpus tetragonolobus</i>	Hit *	50	<i>Pimpinella pruatjan</i>	Hit
23	<i>Parkia roxburghii</i>	Hit	51	<i>Andrographis paniculata</i>	Hit
24	<i>Piper cubeba</i>	Hit *	52	<i>Blumea balsamifera</i>	Hit
25	<i>Kaempferia galanga</i>	Hit	53	<i>Cymbopogon nardus</i>	Hit
26	<i>Coriandrum sativum</i>	Hit	54	<i>Sida rhombifolia</i>	Hit
27	<i>Cola acuminata</i>	Hit	55	<i>Cinnamomum sintok</i>	Hit
28	<i>Coffea arabica</i>	Hit	56	<i>Piper betle</i>	Hit *
29	<i>Orthosiphon stamineus</i>	Hit	57	<i>Talinum paniculatum</i>	Hit
30	<i>Curcuma longa</i>	Hit	58	<i>Elephantopus scaber</i>	Hit
31	<i>Piper nigrum</i>	Hit	59	<i>Cyperus rotundus</i>	Hit
32	<i>Alpinia galanga</i>	Hit	60	<i>Curcuma aeruginosa</i>	Hit
33	<i>Vitex trifolia</i>	Hit	61	<i>Kaempferia pandurata</i>	Hit *
34	<i>Zingiber amaricans</i>	Hit *	62	<i>Curcuma xanthorrhiza</i>	Hit
35	<i>Zingiber zerumbet</i>	Hit	63	<i>Tribulus terrestris</i>	Hit
36	<i>Zingiber aromaticum</i>	Hit	64	<i>Corydalis yanhusuo</i>	Hit
37	<i>Languas galanga</i>	Hit	65	<i>Pausinystalia yohimbe</i>	Hit
38	<i>Massoia aromatica</i>	Hit			
39	<i>Morinda citrifolia</i>	Hit	<b>H. Disease: Nutritional and Metabolic</b>		
40	<i>Carum copticum</i>	Hit *	<b>Diseases</b>		
41	<i>Panax pseudoginseng</i>	Hit *	1	<i>Foeniculum vulgare</i>	Hit
42	<i>Oryza sativa</i>	Hit	2	<i>Glycyrrhiza uralensis</i>	Hit
43	<i>Myristica fragrans</i>	Hit	3	<i>Zingiber purpureum</i>	Hit
44	<i>Pandanus amaryllifolius</i>	Hit	4	<i>Allium sativum</i>	Hit
45	<i>Eurycoma longifolia</i>	Hit	5	<i>Tinospora tuberculata</i>	Hit
46	<i>Hydrocotyle asiatica</i>	Hit	6	<i>Pandanus conoideus</i>	Hit
47	<i>Areca catechu</i>	Hit *	7	<i>Syzygium aromaticum</i>	Hit

No	Plants Name	Hit / Miss Status	No	Plants Name	Hit / Miss Status
8	<i>Punica granatum</i>	Hit	36	<i>Alstonia scholaris</i>	Hit
9	<i>Zingiber officinale</i>	Hit	37	<i>Hibiscus sabdariffa</i>	Hit
10	<i>Guazuma ulmifolia</i>	Hit	38	<i>Laminaria japonica</i>	Hit
11	<i>Nigella sativa</i>	Hit	39	<i>Syzygium polyanthum</i>	Hit
12	<i>Amomum compactum</i>	Hit *	40	<i>Andrographis paniculata</i>	Hit
13	<i>Cinnamomum burmani</i>	Hit	41	<i>Sindora sumatrana</i>	Hit *
14	<i>Parameria laevigata</i>	Hit	42	<i>Cassia angustifolia</i>	Hit
15	<i>Caesalpinia sappan</i>	Hit	43	<i>Woodfordia floribunda</i>	Hit
16	<i>Soya max</i>	Hit *	44	<i>Piper betle</i>	Hit
17	<i>Cocos nucifera</i>	Hit	45	<i>Spirulina</i>	Hit
18	<i>Rheum tanguticum</i>	Hit	46	<i>Stevia rebaudiana</i>	Hit
19	<i>Piper cubeba</i>	Hit *	47	<i>Theae sinensis</i>	Hit
20	<i>Murraya paniculata</i>	Hit	48	<i>Sonchus arvensis</i>	Hit
21	<i>Kaempferia galanga</i>	Hit *	49	<i>Curcuma heyneana</i>	Hit
22	<i>Coffea arabica</i>	Hit *	50	<i>Curcuma aeruginosa</i>	Hit
23	<i>Orthosiphon stamineus</i>	Hit	51	<i>Kaempferia pandurata</i>	Hit *
24	<i>Curcuma longa</i>	Hit	52	<i>Curcuma xanthorrhiza</i>	Hit
25	<i>Piper nigrum</i>	Hit *	53	<i>Curcuma zedoaria</i>	Hit *
26	<i>Zingiber aromaticum</i>	Hit	54	<i>Olea europaea</i>	Hit
27	<i>Aloe vera</i>	Hit			
28	<i>Phaleria papuana</i>	Hit	<b>I. Disease Respiratory Diseases</b>		
29	<i>Galla lusitania</i>	Hit	1	<i>Foeniculum vulgare</i>	Hit
30	<i>Quercus lusitanica</i>	Hit	2	<i>Clausena anisum-olens</i>	Hit
31	<i>Morinda citrifolia</i>	Hit	3	<i>Glycyrrhiza uralensis</i>	Hit
32	<i>Myristica fragrans</i>	Hit *	4	<i>Zingiber purpureum</i>	Hit
33	<i>Momordica charantia</i>	Hit	5	<i>Piper retrofractum</i>	Hit *
34	<i>Areca catechu</i>	Hit	6	<i>Syzygium aromaticum</i>	Hit
35	<i>Lepiniopsis ternatensis</i>	Hit	7	<i>Gaultheria punctata</i>	Hit

No	Plants Name	Hit / Miss Status	No	Plants Name	Hit / Miss Status
8	<i>Panax ginseng</i>	Hit	36	<i>Curcuma xanthorrhiza</i>	Hit
9	<i>Equisetum debile</i>	Hit *	37	<i>Salix alba</i>	Hit *
10	<i>Zingiber officinale</i>	Hit	38	<i>Matricaria chamomilla</i>	Miss *
11	<i>Citrus aurantium</i>	Hit *	<b>J. Disease: Skin and Connective Tissue</b>		
12	<i>Nigella sativa</i>	Hit *	1	<i>Strychnos ligustrina</i>	Hit
13	<i>Amomum compactum</i>	Hit	2	<i>Merremia mammosa</i>	Hit *
14	<i>Cinnamomum burmani</i>	Hit	3	<i>Piper retrofractum</i>	Hit *
15	<i>Melaleuca leucadendra</i>	Hit	4	<i>Santalum album</i>	Hit
16	<i>Parkia roxburghii</i>	Hit	5	<i>Zingiber officinale</i>	Hit *
17	<i>Cocos nucifera</i>	Hit	6	<i>Citrus aurantium</i>	Hit
18	<i>Piper cubeba</i>	Hit	7	<i>Citrus hystrix</i>	Hit
19	<i>Kaempferia galanga</i>	Hit	8	<i>Cassia siamea</i>	Hit
20	<i>Coriandrum sativum</i>	Hit	9	<i>Cocos nucifera</i>	Hit
21	<i>Curcuma longa</i>	Hit	10	<i>Trigonella foenum-graecum</i>	Hit
22	<i>Piper nigrum</i>	Hit	11	<i>Orthosiphon stamineus</i>	Hit
23	<i>Zingiber aromaticum</i>	Hit	12	<i>Curcuma longa</i>	Hit
24	<i>Languas galanga</i>	Hit	13	<i>Vetiveria zizanioides</i>	Hit
25	<i>Mentha piperita</i>	Hit	14	<i>Aloe vera</i>	Hit
26	<i>Oryza sativa</i>	Hit *	15	<i>Rosa chinensis</i>	Hit
27	<i>Myristica fragrans</i>	Hit	16	<i>Jasminum sambac</i>	Hit
28	<i>Pandanus amaryllifolius</i>	Hit *	17	<i>Phyllanthus urinaria</i>	Hit
29	<i>Hydrocotyle asiatica</i>	Hit *	18	<i>Mentha piperita</i>	Hit
30	<i>Mentha arvensis</i>	Hit	19	<i>Oryza sativa</i>	Hit
31	<i>Lepiniopsis ternatensis</i>	Hit	20	<i>Myristica fragrans</i>	Hit *
32	<i>Helicteres isora</i>	Hit	21	<i>Hydrocotyle asiatica</i>	Hit
33	<i>Blumea balsamifera</i>	Hit	22	<i>Lepiniopsis ternatensis</i>	Hit
34	<i>Cymbopogon nardus</i>	Hit	23	<i>Alstonia scholaris</i>	Hit
35	<i>Piper betle</i>	Hit			

No	Plants Name	Hit / Miss Status
24	<i>Andrographis paniculata</i>	Hit
25	<i>Cymbopogon nardus</i>	Hit
26	<i>Piper betle</i>	Hit
27	<i>Theae sinensis</i>	Hit
28	<i>Curcuma heyneana</i>	Hit
29	<i>Kaempferia pandurata</i>	Hit *
30	<i>Curcuma xanthorrhiza</i>	Hit
31	<i>Melaleuca leucadendra</i>	Hit
32	<i>Matricaria chamomilla</i>	Miss *

#### K. Disease: The Urinary System

1	<i>Foeniculum vulgare</i>	Hit *
2	<i>Imperata cylindrica</i>	Hit *
3	<i>Strychnos ligustrina</i>	Hit *
4	<i>Plantago major</i>	Hit
5	<i>Zingiber officinale</i>	Hit *
6	<i>Cinnamomum burmani</i>	Hit *
7	<i>Strobilanthes crispus</i>	Hit
8	<i>Kaempferia galanga</i>	Hit *
9	<i>Orthosiphon stamineus</i>	Hit
10	<i>Phyllanthus urinaria</i>	Hit
11	<i>Blumea balsamifera</i>	Hit *
12	<i>Sonchus arvensis</i>	Hit
13	<i>Curcuma xanthorrhiza</i>	Hit





## Appendix C

### The mean AUCs for each disease class

The mean of AUCs between equations and disease classes in Jamu data.  $S$  is similarity measure,  $D$  is dissimilarity measure, \* Incl.  $d$  indicates the availability of negative match quantity  $d$  in the equation (Yes/No).

No	Eq.	S/Incl.		Disease classes														Overall
		IDs	D	$d^*$	E1	E2	E3	E6	E8	E9	E10	E11	E13	E14	E15	E16	E17	E18
1	Eq. 48	S	Y	0.55	0.73	0.60	0.62	0.60	0.81	0.57	0.64	0.61	0.57	0.59	0.57	0.70	0.61	0.632
2	Eq. 49	S	Y	0.53	0.74	0.60	0.60	0.61	0.82	0.54	0.63	0.60	0.59	0.61	0.58	0.69	0.60	0.630
3	Eq. 74	S	Y	0.54	0.73	0.60	0.61	0.61	0.83	0.55	0.63	0.60	0.58	0.60	0.58	0.69	0.60	0.629
4	Eq. 44	S	Y	0.53	0.74	0.60	0.60	0.60	0.83	0.54	0.63	0.61	0.59	0.62	0.58	0.68	0.61	0.629
5	Eq. 54	S	Y	0.53	0.74	0.60	0.60	0.60	0.83	0.54	0.63	0.61	0.59	0.62	0.58	0.68	0.61	0.629
6	Eq. 79	S	Y	0.53	0.73	0.60	0.60	0.60	0.83	0.54	0.63	0.60	0.59	0.62	0.58	0.68	0.60	0.627
7	Eq. 66	S	Y	0.53	0.73	0.59	0.60	0.60	0.82	0.54	0.63	0.61	0.59	0.61	0.58	0.68	0.61	0.626
8	Eq. 68	S	Y	0.53	0.72	0.59	0.60	0.60	0.79	0.54	0.63	0.61	0.59	0.61	0.57	0.68	0.61	0.624
9	Eq. 04	S	N	0.57	0.59	0.57	0.64	0.58	0.71	0.55	0.66	0.63	0.53	0.60	0.54	0.69	0.61	0.610
10	Eq. 36	S	N	0.57	0.58	0.57	0.63	0.59	0.72	0.55	0.66	0.63	0.53	0.60	0.54	0.69	0.61	0.610
11	Eq. 06	S	N	0.57	0.58	0.57	0.64	0.58	0.71	0.55	0.66	0.63	0.53	0.60	0.54	0.69	0.61	0.610
12	Eq. 39	S	N	0.57	0.58	0.57	0.63	0.59	0.72	0.55	0.66	0.63	0.53	0.60	0.54	0.69	0.61	0.610
13	Eq. 29	D	N	0.57	0.58	0.57	0.63	0.58	0.72	0.55	0.66	0.63	0.53	0.60	0.54	0.69	0.61	0.610
14	Eq. 57	S	Y	0.57	0.59	0.57	0.63	0.58	0.71	0.55	0.66	0.62	0.53	0.60	0.54	0.69	0.61	0.610
15	Eq. 78	S	N	0.57	0.58	0.57	0.64	0.58	0.71	0.55	0.66	0.63	0.53	0.60	0.54	0.69	0.61	0.610
16	Eq. 27	D	N	0.57	0.58	0.57	0.64	0.58	0.72	0.55	0.66	0.62	0.53	0.60	0.54	0.69	0.61	0.609
17	Eq. 01	S	N	0.57	0.58	0.57	0.64	0.58	0.71	0.55	0.66	0.63	0.53	0.60	0.54	0.69	0.61	0.609
18	Eq. 02	S	N	0.57	0.59	0.57	0.64	0.58	0.71	0.55	0.66	0.63	0.53	0.60	0.54	0.69	0.61	0.609
19	Eq. 31	S	N	0.57	0.59	0.57	0.63	0.58	0.72	0.55	0.66	0.62	0.53	0.60	0.54	0.69	0.61	0.609
20	Eq. 46	S	N	0.57	0.58	0.57	0.64	0.58	0.72	0.55	0.66	0.63	0.53	0.60	0.54	0.68	0.61	0.609
21	Eq. 71	S	Y	0.57	0.58	0.57	0.63	0.58	0.71	0.55	0.66	0.62	0.53	0.60	0.54	0.69	0.61	0.609
22	Eq. 34	S	N	0.57	0.59	0.57	0.63	0.59	0.71	0.56	0.65	0.62	0.53	0.60	0.54	0.69	0.61	0.609
23	Eq. 40	S	Y	0.57	0.59	0.57	0.63	0.58	0.71	0.55	0.65	0.61	0.53	0.60	0.54	0.69	0.61	0.608
24	Eq. 61	S	Y	0.57	0.58	0.57	0.63	0.59	0.70	0.56	0.65	0.61	0.53	0.60	0.54	0.69	0.61	0.607
25	Eq. 45	S	N	0.57	0.58	0.57	0.63	0.59	0.71	0.55	0.65	0.62	0.53	0.60	0.54	0.69	0.60	0.607
26	Eq. 62	D	Y	0.57	0.58	0.57	0.63	0.58	0.70	0.55	0.65	0.62	0.53	0.60	0.54	0.69	0.61	0.607

No	Eq. S/Incl.		Disease classes															Overall
	IDs	D	d*	E1	E2	E3	E6	E8	E9	E10	E11	E13	E14	E15	E16	E17	E18	mean
27	Eq. 55	S	Y	0.57	0.58	0.57	0.63	0.59	0.70	0.55	0.65	0.62	0.53	0.60	0.54	0.69	0.61	0.607
28	Eq. 63	S	Y	0.57	0.58	0.57	0.63	0.59	0.70	0.55	0.65	0.62	0.53	0.60	0.54	0.69	0.61	0.607
29	Eq. 77	S	Y	0.56	0.57	0.55	0.63	0.57	0.69	0.54	0.66	0.63	0.53	0.59	0.53	0.67	0.60	0.600
30	Eq. 12	S	N	0.56	0.57	0.55	0.63	0.57	0.69	0.54	0.66	0.63	0.53	0.59	0.53	0.67	0.60	0.600
31	Eq. 10	S	Y	0.49	0.78	0.62	0.49	0.57	0.84	0.53	0.51	0.50	0.60	0.55	0.61	0.61	0.53	0.593
32	Eq. 52	S	Y	0.61	0.43	0.54	0.67	0.57	0.60	0.57	0.69	0.65	0.47	0.58	0.51	0.71	0.61	0.591
33	Eq. 51	S	Y	0.61	0.43	0.54	0.67	0.57	0.60	0.56	0.69	0.65	0.47	0.58	0.51	0.71	0.61	0.590
34	Eq. 35	S	Y	0.46	0.75	0.59	0.46	0.56	0.80	0.52	0.51	0.51	0.59	0.56	0.61	0.59	0.53	0.578
35	Eq. 59	S	Y	0.46	0.74	0.59	0.46	0.56	0.79	0.52	0.51	0.51	0.59	0.56	0.61	0.59	0.53	0.577
36	Eq. 08	S	Y	0.47	0.77	0.61	0.46	0.56	0.85	0.54	0.46	0.45	0.59	0.53	0.61	0.57	0.51	0.575
37	Eq. 07	S	Y	0.47	0.77	0.61	0.46	0.56	0.84	0.54	0.46	0.45	0.59	0.53	0.61	0.57	0.51	0.575
38	Eq. 15	D	N	0.47	0.77	0.61	0.46	0.56	0.85	0.54	0.46	0.45	0.59	0.53	0.61	0.57	0.50	0.575
39	Eq. 24	D	Y	0.47	0.77	0.61	0.46	0.56	0.84	0.53	0.46	0.45	0.59	0.53	0.61	0.57	0.52	0.575
40	Eq. 09	S	Y	0.47	0.77	0.61	0.46	0.56	0.85	0.53	0.46	0.45	0.59	0.53	0.61	0.57	0.51	0.575
41	Eq. 16	D	N	0.47	0.77	0.61	0.46	0.56	0.84	0.53	0.46	0.45	0.59	0.53	0.61	0.57	0.51	0.574
42	Eq. 25	D	Y	0.47	0.78	0.61	0.46	0.56	0.84	0.53	0.46	0.45	0.59	0.53	0.62	0.57	0.52	0.574
43	Eq. 26	D	Y	0.45	0.77	0.61	0.44	0.57	0.83	0.53	0.44	0.44	0.59	0.55	0.61	0.58	0.52	0.570
44	Eq. 47	S	N	0.48	0.76	0.60	0.47	0.53	0.84	0.55	0.45	0.43	0.57	0.48	0.60	0.53	0.49	0.562
45	Eq. 50	S	Y	0.42	0.75	0.58	0.38	0.53	0.83	0.51	0.35	0.37	0.57	0.50	0.61	0.47	0.46	0.529
Overall mean				0.53	0.66	0.58	0.58	0.58	0.76	0.54	0.60	0.57	0.55	0.58	0.57	0.65	0.58	

## Appendix D

### The inTrees framework

**Rule Extraction** (Deng 2014)

*condExtract(condSet, node, C, maxDepth, currentDepth)* is a function to extract conditions *condSet* from a tree ensemble. In the rule extraction algorithm, let *C* represent the conjunction of variable-value pairs aggregated in the path from the root node to the current node, *C<sub>node</sub>* indicate the variable-value pair used to split the current node, *leafNode* represent the flag whether the current node is a leaf node. We can also define a maximum depth, *maxDepth*, of the tree where the conditions are extracted from. In a decision tree, most useful splits occur in the top levels of the trees. Then, setting a maximum depth not only reduces the computations but may also avoid extracting overfitting rules. When we set *maxDepth*=-1, it means there is no limitation on the depth.

```
input  : condSet←null, node←rootNode, C←null, maxDepth←-1,
         curretDepth←0
output : condSet
         curretDepth = curretDepth+1
if leafNode = true or curretDepth = maxDepth then
    condSet←{condSet, currentCond}
    return condSet
end
for childi = every child of node do
    C←C ∧ Cnode
    condSet←condExtract(condSet, childi, C, maxDepth, curretDepth)
end
return condSet
```

### Rule Pruning (Deng 2014)

A rule can be expressed by  $\{C \Rightarrow T\}$ , where  $C$  indicates the condition of the rule as a conjunction of variable-value pairs, and  $T$  indicates the outcome of the rule. The condition of a rule is represented as variable-value pairs,  $a_1 = v_1, \dots, a_i = v_i, \dots, a_K = v_K$ , where  $a_i$  and  $v_i$  is the  $i$ -th variable-value pair, and  $K$  ( $K \geq 1$ ) is the total number of variable-value pairs. The extracted rule from trees may include irrelevant variable-value pairs.

Let  $E$  indicate a metric measuring the quality of a rule, and a smaller value of  $E$  implies a better rule. Let  $E_0$  denote the  $E$  of the original rule  $\{C \Rightarrow T\}$ , and let  $E_{-i}$  denote the  $E$  of the rule that leaves the  $i$ -th variable-value pair out. We use  $decay_i$  to evaluate the effect of removing the  $i$ -th pair:

$$decay_i = \frac{E_{-i} - E_0}{\max(E_0, s)}$$

where  $s$  is a positive number that bound the value of  $decay_i$  when  $E_0$  is 0 or very small. In the inTree R package, they currently set  $s = 10^{-6}$ . Alternatively, we can use  $decay_i$  as follows:

$$decay_i = E_{-i} - E_0$$

## Appendix E

# Feature selection using Regularized RF

Selected features using RRF with  $\lambda=0.8$ .

Compound ID	Compound name	Compound ID	Compound name
C8	(+)-7beta-Acetoxy-3,13-clerodadien-16,15-olid-18-oic acid		3,8,22<d0>trihydroxy-cholest-5,14,16,23-tetraene-1beta-yl-6-O-(3,4,5-trimethoxybenzoyl)-beta-D-glycopyranoside
C13	(+)-Catechin		
C45	(-)-Epicatechin		
C61	(-)-panduratin A	C373	17^3-Ethoxypheophorbide B
C98	(4Z)-1-(2,3,5-trihydroxy-4-methylphenyl)dec-4-en-1-one	C395	2'-Hydroxy-4,4',6'-trimethoxychalcone
C109	(E)-2-Decenal	C411	2,3-Dihydrobenzofuran
C122	(E)-Anethole	C429	2,6,10,14,18,22 Tetracosahexaene
C162	(S)-6-Gingerol	C436	2,6-Dimethyl bicyclo [3,2,1]octane
C166	(Z)-3-Hexenyl acetate	C443	2-(1-Oxypentyl)-benzoic acid methyl ester
C182	(Z)-beta-Farnesene	C458	2-Cycloprophylthiophene
C183	(Z)-beta-Ocimene	C482	2-Hydroxy-4,5-methylenedioxypropio phenone
C200	1,2,3,4,6-penta-O-galloyl-beta-D-glucose	C533	2-Octenal
C211	1,2-di-O-alpha-Linolenoyl-3-O-beta-galactopyranosyl-sn-glycerol	C539	2-Phenylethanol
C278	1-Methyl-3-propyl benzene	C553	2-epi-ziza-6(13)-en-3alpha-ol
C281	1-Monolaurin	C555	2-hydroxy-3',4'-dihydroxyacetophenone
C309	1-Tetradecene		
C316	1-epi-Cubenol	C558	20(S)-ginsenoside Rh1 (Rh1)
C326	10-Undecyn-1-ol	C600	3,4,5-trimethoxy geraniin
C338	12-Oxo-10E-dodecenoic acid	C601	3,4,6-trihydroxyphenanthrene-3-O-beta-D-glucopyranoside
C342	13-Methylpalmatine		
C365	16-beta-(beta-D glucopyranosyl,	C653	3-Mercaptodecene

<b>Compound ID</b>	<b>Compound name</b>	<b>Compound ID</b>	<b>Compound name</b>
C666	3-Methylbutanol	C1298	Chicoric acid
C670	3-Methylquercetin	C1304	Cholesterol
C714	3-Pentadecylphenol	C1331	Cinnamic acid
C744	3beta-p-Hydroxybenzoyldehydrotumulosic acid	C1347	Citronnellal
C833	5-Allyl-3-(4-allyl-2-methoxyphenoxy methyl)-2-(4-hydroxy-3-methoxyphenyl)-7-methoxy-2,3-dihydrobenzofuran	C1377	Constictic acid
C942	7-O-(beta-D-Glucosyl)apigenin	C1387	Coumaperine
C966	8-Heptadecene	C1389	Coumarin
C967	8-Hydroxy-9-methoxycanthin-6-one	C1418	Curcumenol
C1022	Aciphyllene	C1429	Curzerenone
C1047	Allylpyrocatechol	C1455	Cytidine
C1065	Anisucumarin A	C1456	D-Catechin
C1070	Apigenin	C1462	Daidzein
C1084	Apigenin-7-O-beta-D-glucuronic acid 6" methyl ester	C1501	Diacetoxy-[6]-gingerdiol
C1087	Apigenin-7-O-neohesperidoside	C1507	Diallyl sulfide
C1103	Asaraldehyde	C1510	Dibutyl phthalate
C1108	Asperulosidic acid	C1531	Dihydropiperlonguminine
C1126	Benzaldehyde	C1585	Elemol
C1148	Benzylidenemalonaldehyde	C1589	Ellagic acid
C1189	Bornyl acetate	C1600	Epifriedelinol
C1191	Bornylmagnolol	C1604	Epimedine B
C1197	Brucine N-Oxide	C1608	Epoxy-mollugin
C1219	C-12 Massoia lactone	C1609	Epoxy-sesquithujene
C1227	Caffeic acid	C1621	Eriocitrin
C1263	Carvone	C1636	Ethyl 3-methylbutyrate
C1280	Castillicetin	C1638	Ethyl acetate
C1283	Catechin	C1660	Ethyl-4E-octenoate
C1287	Cedrene	C1672	Eugenol
		C1721	Fructose
		C1724	Fumaric acid
		C1731	Galactose
		C1734	Gallic acid

<b>Compound ID</b>	<b>Compound name</b>	<b>Compound ID</b>	<b>Compound name</b>
C1748	Ganoderic acid S	C2153	Liquiritigenin
C1756	Garbogiol	C2158	Longifolene
C1759	Geniposide	C2164	Lupeol
C1771	Geraniol	C2166	Luteolin
C1773	Geranyl acetate	C2185	Luteolin-7-methyl ether
C1779	Germacrene B	C2237	Menthol
C1780	Germacrene D	C2247	Mesuagin
C1820	Glycoside D	C2311	Methyl salicylate
C1841	Guanoside	C2341	Morin-3-O-lyxoside
C1847	Hederagenin methyl ester	C2356	Myristicin
C1852	Hentriacontane	C2385	N-trans-Feruloyltyramine
C1866	Hexacosane	C2412	Nerol
C1870	Hexadecanoic acid, ethyl ester	C2432	Nonanol
C1883	Hexyl butyrate	C2451	Octacosane
C1890	Heyneanone C	C2479	Oleuropein aglycone
C1892	Hibiscetin	C2496	Palmitic acid
C1913	Hydroxycinnamic acid	C2529	Pentatriacontane
C1915	Hydroxygenkwanin	C2533	Petunidin
C1932	Icariside F2	C2566	Physcion
C1963	Isocryptomerin	C2568	Phytol
C1996	Isopulegol	C2619	Proanthocyanidin A-6
C2003	Isorhamnetin 3-O-(6-O-rhamnosyl-galactoside)	C2629	Protocatechuic acid
C2025	Isoscutellarein 4'-methyl ether 8-O-beta-D-glucuronide 6	C2631	Protosappanin A
C2037	Kaempferol	C2655	Quercetin
C2071	Kaempferol-3-O-(2,4-di-O-acetyl-alpha-L-rhamnopyranoside)	C2657	Quercetin 3,7-O-diglucoside
C2105	Kizutasaponin K12	C2659	Quercetin 3-O-(2'',6''-digalloyl)-beta-D-glucoside
C2134	Leucoanthocyanidins	C2699	Quercetin-7-O-rhamnoside
C2143	Linalool	C2703	Quercitrin
		C2731	Rutin
		C2740	Saffloquinoside B

<b>Compound ID</b>	<b>Compound name</b>	<b>Compound ID</b>	<b>Compound name</b>
C2758	Salicin	C3109	[21alpha-methylmelianol (21R,23R) epoxy-24-hydroxy-21alpha-Methoxyl]
C2764	Sammangaoside A		triucalla-7,25-dien-3-one
C2773	Scandoside methyl ester (6beta-Hydroxygeniposide)	C3117	a-Humulene
C2785	Seguinose K 4-methylether	C3125	alpha-Bergamotene
C2792	Senkyunolide J	C3138	alpha-Copaene
C2798	Sesbanimide	C3154	alpha-Murolene
C2806	Silychristin	C3159	alpha-Phellandrene
C2811	Sinensetin	C3162	alpha-Santalene
C2817	Skullcapflavone I 2'-glucoside	C3170	alpha-Terpinyl acetate
C2836	Stigmast-22-en-3-ol	C3171	alpha-Thujene
C2844	Stigmasterol	C3172	alpha-Thujone
C2847	Strictosamine	C3189	beta-Amyrin
C2851	Styrene	C3192	beta-Bourbonene
C2868	T-Murolol	C3197	beta-Caryophyllene
C2874	Tannic acid	C3205	beta-Cubebene
C2880	Taxifolin	C3211	beta-E-caryophyllene
C2906	Tetradecanal	C3213	beta-Elemenone
C2927	Thujopsene	C3232	beta-Pregnane
C2933	Thymol	C3237	beta-Sitosterol
C2947	Tribulusterine	C3249	beta-ekasantalic acid
C2949	Tricin	C3306	delta-Cadinene
C2998	Tyramine	C3311	delta-Guaiene
C3013	Ursolic acid	C3323	ent-Gallocatechin
C3031	Verbenone	C3353	gamma-Murolene
C3101	Ziganein	C3360	gamma-Terpinene
C3103	Zingiberene	C3458	trans-Cinnamaldehyde
C3106	[10]-Shogaol	C3473	trans-Pinocarveol



## Appendix F

### List of compounds and rules extracted from Jamu medicines

List of compounds identified as important metabolites for each disease class. The red color indicates the utilization of a metabolite for prediction of Jamu efficacy. #rules means the number of selected rules that uses this compound to predict Jamu efficacy.

No	Compound ID	Metabolites	#Rules	Efficacy groups													
				1	2	3	6	8	10	11	14	15	16	17	18		
1	C98	(4Z)-1-(2,3,5-trihydroxy-4-methylphenyl)dec-4-en-1-one	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	C326	10-Undecyn-1-ol	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	C373	17 <sup>^</sup> 3-Ethoxypheophorbide B	18	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	C1227	Caffeic acid	24	1	0	0	0	0	1	0	0	0	0	0	1	0	0
5	C1387	Coumapherine	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0
6	C1501	Diacetoxy-[6]-gingerdiol	20	1	0	0	0	0	0	1	0	1	0	0	0	0	0
7	C1734	Gallic acid	12	1	0	0	1	0	0	0	0	1	0	0	0	0	0
8	C1852	Hentriacontane	2	1	0	0	0	0	0	0	0	0	0	1	0	0	0
9	C2166	Luteolin	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
10	C2703	Quercitrin	29	1	0	0	0	0	0	0	0	0	0	0	0	0	0
11	C2949	Tricin	6	1	0	0	0	0	0	0	0	0	0	1	0	0	0
12	C3013	Ursolic acid	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0
13	C3189	beta-Amyrin	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0
14	C3197	beta-Caryophyllene	29	1	0	0	1	1	0	1	0	0	0	0	0	0	0
15	C3237	beta-Sitosterol	3	1	0	0	0	0	0	0	0	0	0	1	1	0	0
16	C1585	Elemol	74	0	1	0	0	0	0	1	0	1	0	0	0	0	0
17	C1779	Germacrene B	100	0	1	0	0	0	0	1	0	0	0	0	0	0	0
18	C3103	Zingiberene	35	0	1	0	1	0	0	0	1	0	0	0	0	0	0
19	C3117	a-Humulene	27	0	1	0	0	0	0	0	0	1	0	0	0	1	0
20	C3213	beta-Elementone	13	0	1	0	0	0	0	0	0	0	0	0	0	0	0

No	Compound ID	Metabolites	#Rules	Efficacy groups													
				1	2	3	6	8	10	11	14	15	16	17	18		
21	C183	(Z)-beta-Ocimene	47	0	0	1	1	0	0	1	0	1	0	0	0		
22	C316	1-epi-Cubenol	16	0	0	1	0	0	0	0	0	0	0	0	0		
23	C653	3-Mercaptodecene	9	0	0	1	0	0	0	0	0	1	0	0	0		
24	C1065	Anisucumarin A	12	0	0	1	0	0	0	0	0	0	0	0	0		
25	C1126	Benzaldehyde	2	0	0	1	0	0	0	0	0	1	0	0	0		
26	C1189	Bornyl acetate	2	0	0	1	0	0	0	0	1	1	0	0	0		
27	C1672	Eugenol	2	0	0	1	0	1	0	0	0	1	0	1	0		
28	C1773	Geranyl acetate	4	0	0	1	1	0	0	1	0	0	0	0	0		
29	C1780	Germacrene D	2	0	0	1	0	0	0	0	0	1	0	0	0		
30	C2158	Longifolene	3	0	0	1	0	0	0	0	0	1	0	0	0		
31	C2341	Morin-3-O-lyxoside	8	0	0	1	0	0	0	0	0	0	0	0	0		
32	C2496	Palmitic acid	1	0	0	1	1	0	0	0	0	0	0	0	0		
33	C2927	Thujopsene	6	0	0	1	0	0	0	0	0	0	0	0	0		
34	C3154	alpha-Muurolene	1	0	0	1	0	0	0	0	0	0	0	0	0		
35	C3159	alpha-Phellandrene	1	0	0	1	0	0	0	1	0	0	0	0	0		
36	C3170	alpha-Terpinyl acetate	3	0	0	1	0	0	0	0	0	1	0	0	0		
37	C3353	gamma-Muurolene	4	0	0	1	0	0	0	0	0	0	0	0	0		
38	C45	(-)-Epicatechin	31	0	0	0	1	0	0	0	0	0	0	1	0		
39	C555	2-hydroxy-3',4'-dihydroxyacetophenone	5	0	0	0	1	0	0	0	0	0	0	0	0		
40	C1047	Allylpyrocatechol	3	0	0	0	1	0	0	0	0	0	0	0	0		
41	C1418	Curcumenol	1	0	0	0	1	0	0	0	0	0	0	0	0		
42	C1589	Ellagic acid	1	0	0	0	1	0	0	0	0	1	0	0	0		
43	C1724	Fumaric acid	5	0	0	0	1	0	0	0	0	0	0	0	0		
44	C1963	Isocryptomerin	24	0	0	0	1	0	0	0	0	0	0	0	0		
45	C2619	Proanthocyanidin A-6	7	0	0	0	1	0	0	0	0	0	0	0	0		
46	C2844	Stigmasterol	2	0	0	0	1	0	0	0	0	0	0	1	0		
47	C3171	alpha-Thujene	2	0	0	0	1	0	0	0	0	1	0	0	0		
48	C1507	Diallyl sulfide	7	0	0	0	0	1	0	0	0	0	0	0	0		
49	C2792	Senkyunolide J	1	0	0	0	0	1	0	0	0	0	0	0	0		
50	C281	1-Monolaurin	1	0	0	0	0	0	1	0	0	0	0	0	0		

No	Compound ID	Metabolites	#Rules	Efficacy groups													
				1	2	3	6	8	10	11	14	15	16	17	18		
51	C1913	Hydroxycinnamic acid	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0
52	C2037	Kaempferol	16	0	0	0	0	0	0	1	0	0	0	0	0	1	0
53	C2731	Rutin	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
54	C182	(Z)-beta-Farnesene	37	0	0	0	0	0	0	0	1	0	1	0	0	0	0
55	C539	2-Phenylethanol	13	0	0	0	0	0	0	0	1	0	0	0	0	0	0
56	C967	8-Hydroxy-9-methoxycanthin -6-one	15	0	0	0	0	0	0	0	1	0	0	0	0	0	0
57	C1429	Curzerenone	3	0	0	0	0	0	0	0	1	0	0	0	0	0	0
58	C1531	Dihydropiperlonguminine	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
59	C1660	Ethyl-4E-octenoate	3	0	0	0	0	0	0	0	1	0	0	0	0	0	0
60	C2356	Myristicin	3	0	0	0	0	0	0	0	1	0	1	0	0	0	1
61	C3106	[10]-Shogaol	3	0	0	0	0	0	0	0	1	0	1	0	0	0	0
62	C3205	beta-Cubebene	3	0	0	0	0	0	0	0	1	0	0	0	0	0	0
63	C3306	delta-Cadinene	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
64	C1103	Asaraldehyde	49	0	0	0	0	0	0	0	0	1	0	0	0	0	0
65	C2868	T-Muurolol	3	0	0	0	0	0	0	0	0	1	0	0	0	1	0
66	C3125	alpha-Bergamotene	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0
67	C3323	ent-Gallocatechin	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0
68	C3473	trans-Pinocarveol	14	0	0	0	0	0	0	0	0	1	0	0	0	0	0
69	C122	(E)-Anethole	21	0	0	0	0	0	0	0	0	0	1	0	0	0	0
70	C166	(Z)-3-Hexenyl acetate	11	0	0	0	0	0	0	0	0	0	1	0	0	0	0
71	C309	1-Tetradecene	11	0	0	0	0	0	0	0	0	0	1	0	0	0	0
72	C1263	Carvone	20	0	0	0	0	0	0	0	0	0	1	0	0	0	0
73	C1347	Citronnellal	5	0	0	0	0	0	0	0	0	0	1	0	0	0	0
74	C1771	Geraniol	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1
75	C1996	Isopulegol	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
76	C2143	Linalool	2	0	0	0	0	0	0	0	0	0	1	0	0	0	1
77	C2185	Luteolin-7-methyl ether	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0
78	C2237	Menthol	3	0	0	0	0	0	0	0	0	0	1	0	0	0	0
79	C2432	Nonanol	4	0	0	0	0	0	0	0	0	0	1	0	0	0	0
80	C2655	Quercetin	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0

No	Compound		#Rules	Efficacy groups														
	ID	Metabolites		1	2	3	6	8	10	11	14	15	16	17	18			
81	C3031	Verbenone	6	0	0	0	0	0	0	0	0	1	0	0	0			
82	C3138	alpha-Copaene	3	0	0	0	0	0	0	0	0	1	0	0	0			
83	C3192	beta-Bourbonene	4	0	0	0	0	0	0	0	0	1	0	0	1			
84	C3162	alpha-Santalene	10	0	0	0	0	0	0	0	0	1	0	0				
85	C13	(+)-Catechin	14	0	0	0	0	0	0	0	0	0	1	0				
86	C411	2,3-Dihydrobenzofuran	5	0	0	0	0	0	0	0	0	0	1	0				
87	C1070	Apigenin	7	0	0	0	0	0	0	0	0	0	1	0				
88	C2451	Octacosane	5	0	0	0	0	0	0	0	0	0	1	0				
89	C2568	Phytol	14	0	0	0	0	0	0	0	0	0	1	0				
90	C2811	Sinensetin	2	0	0	0	0	0	0	0	0	0	1	0				
91	C2906	Tetradecanal	2	0	0	0	0	0	0	0	0	0	1	0				
92	C1609	Epoxysequithujene	20	0	0	0	0	0	0	0	0	0	0	1				
93	C2933	Thymol	2	0	0	0	0	0	0	0	0	0	0	1				
94	C3360	gamma-Terpinene	1	0	0	0	0	0	0	0	0	0	0	1				
Number of compounds for each efficacy group				15	5	17	16	4	5	18	7	32	4	14	8			

List of selected rules for every disease class according to the thresholds defined in Table 4.6.

Rule ID	Rules	Length	Freq.	Error	Class ID
R1	C98 = '1' & C3013 = '1'	2	0.006	0	E1
R2	C183 = '0' & C326 = '1' & C558 = '0' & C1501 = '0' & C1852 = '1'	5	0.004	0	E1
R3	C326 = '1' & C558 = '0' & C1531 = '0' & C1852 = '1' & C2025 = '0' & C2185 = '0'	6	0.006	0	E1
R4	C98 = '1' & C1418 = '0' & C2166 = '1' & C3197 = '1'	4	0.004	0	E1
R5	C326 = '1' & C558 = '0' & C1065 = '0' & C1531 = '0' & C2949 = '1' & C3197 = '0'	6	0.004	0	E1
R6	C183 = '0' & C326 = '1' & C558 = '0' & C1531 = '0' & C1852 = '1' & C2143 = '0'	6	0.004	0	E1
R7	C326 = '1' & C558 = '0' & C1501 = '0' & C1531 = '0' & C1852 = '1'	5	0.005	0	E1
R8	C183 = '0' & C2037 = '0' & C2811 = '0' & C3103 = '0' & C3189 = '1' & C3323 = '0'	6	0.004	0	E1
R9	C326 = '1' & C558 = '0' & C1531 = '0' & C1852 = '1' & C3106 = '0'	5	0.005	0	E1
R10	C2703 = '1' & C2731 = '0' & C3013 = '1'	3	0.005	0	E1
R11	C326 = '1' & C558 = '0' & C1531 = '0' & C1589 = '0' & C1852 = '1' & C2655 = '0'	6	0.006	0	E1
R12	C326 = '1' & C558 = '0' & C1531 = '0' & C1589 = '0' & C1852 = '1' & C2185 = '0'	6	0.006	0	E1
R13	C109 = '0' & C1148 = '0' & C1387 = '1' & C1531 = '0' & C1589 = '0' & C3106 = '0' & C3237 = '1'	7	0.004	0	E1
R14	C183 = '0' & C326 = '1' & C558 = '0' & C1852 = '1' & C3106 = '0'	5	0.004	0	E1
R15	C373 = '1' & C1418 = '0' & C2037 = '0' & C3117 = '0'	4	0.004	0	E1
R16	C183 = '0' & C326 = '1' & C558 = '0' & C1531 = '0' & C1585 = '0' & C1852 = '1'	6	0.005	0	E1

Rule ID	Rules	Length	Freq.	Error	Class ID
R17	C326 = '1' & C558 = '0' & C1531 = '0' & C2949 = '1' & C3013 = '0' & C3106 = '0'	6	0.005	0	E1
R18	C183 = '0' & C326 = '1' & C558 = '0' & C1531 = '0' & C1852 = '1' & C2568 = '0'	6	0.004	0	E1
R19	C326 = '1' & C558 = '0' & C1531 = '0' & C1852 = '1' & C3197 = '0'	5	0.004	0	E1
R20	C326 = '1' & C558 = '0' & C1531 = '0' & C1779 = '0' & C1852 = '1' & C2037 = '0'	6	0.005	0	E1
R21	C1227 = '1' & C1501 = '1' & C1589 = '0' & C1734 = '1' & C1996 = '0' & C3237 = '0'	6	0.004	0	E1
R22	C373 = '1' & C2037 = '0' & C3103 = '0' & C3117 = '0'	4	0.004	0	E1
R23	C326 = '1' & C558 = '0' & C1429 = '0' & C1589 = '0' & C1852 = '1' & C3106 = '0'	6	0.005	0	E1
R24	C326 = '1' & C558 = '0' & C1531 = '0' & C1585 = '0' & C1773 = '0' & C1852 = '1'	6	0.005	0	E1
R25	C558 = '0' & C1387 = '1' & C1501 = '0' & C1531 = '0' & C1589 = '0' & C1672 = '0' & C3103 = '0' & C3197 = '1'	8	0.004	0	E1
R26	C373 = '1' & C3106 = '0' & C3306 = '0'	3	0.004	0	E1
R27	C98 = '1' & C1672 = '0' & C3189 = '1'	3	0.005	0	E1
R28	C373 = '1' & C1780 = '0' & C3103 = '0'	3	0.004	0	E1
R29	C183 = '0' & C326 = '1' & C558 = '0' & C1531 = '0' & C2143 = '0' & C2844 = '0' & C2949 = '1'	7	0.004	0	E1
R30	C326 = '1' & C558 = '0' & C1501 = '0' & C1531 = '0' & C2949 = '1' & C3013 = '0'	6	0.005	0	E1
R31	C1418 = '0' & C1589 = '0' & C2037 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R32	C183 = '0' & C1589 = '0' & C2731 = '0' & C3213 = '1'	4	0.006	0	E2
R33	C1589 = '0' & C2037 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R34	C183 = '0' & C1418 = '0' & C2037 = '0' & C3103 = '1'	4	0.006	0	E2

Rule ID	Rules	Length	Freq.	Error	Class ID
R35	C1589 = '0' & C2451 = '0' & C3159 = '0' & C3213 = '1'	4	0.006	0	E2
R36	C183 = '0' & C1589 = '0' & C2037 = '0' & C3213 = '1'	4	0.006	0	E2
R37	C183 = '0' & C1047 = '0' & C2143 = '0' & C3213 = '1'	4	0.006	0	E2
R38	C45 = '0' & C183 = '0' & C558 = '0' & C1585 = '1' & C1779 = '1' & C2143 = '0' & C2619 = '0'	7	0.006	0	E2
R39	C166 = '0' & C183 = '0' & C1047 = '0' & C3213 = '1'	4	0.006	0	E2
R40	C1126 = '0' & C1418 = '0' & C3103 = '1' & C3197 = '0'	4	0.006	0	E2
R41	C183 = '0' & C1418 = '0' & C2143 = '0' & C3103 = '1'	4	0.006	0	E2
R42	C183 = '0' & C1047 = '0' & C1585 = '1' & C2143 = '0' & C3103 = '1' & C3106 = '0'	6	0.006	0	E2
R43	C1047 = '0' & C2731 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R44	C45 = '0' & C1589 = '0' & C1866 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R45	C2166 = '0' & C2619 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R46	C1501 = '0' & C1585 = '1' & C2166 = '0' & C2619 = '0' & C3117 = '1' & C3197 = '0'	6	0.006	0	E2
R47	C183 = '0' & C411 = '0' & C1589 = '0' & C2655 = '0' & C3213 = '1'	5	0.006	0	E2
R48	C1589 = '0' & C2451 = '0' & C3213 = '1' & C3360 = '0'	4	0.006	0	E2
R49	C183 = '0' & C1047 = '0' & C2166 = '0' & C3213 = '1'	4	0.006	0	E2
R50	C1585 = '1' & C2037 = '0' & C2143 = '0' & C2619 = '0' & C3197 = '0'	5	0.006	0	E2
R51	C183 = '0' & C1589 = '0' & C2143 = '0' & C2811 = '0' & C3213 = '1'	5	0.006	0	E2
R52	C45 = '0' & C183 = '0' & C2143 = '0' & C2619 = '0' & C3213 = '1'	5	0.006	0	E2
R53	C183 = '0' & C1047 = '0' & C2451 = '0' & C3213 = '1'	4	0.006	0	E2

Rule ID	Rules	Length	Freq.	Error	Class ID
R54	C1589 = '0' & C2166 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R55	C1047 = '0' & C2143 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R56	C1418 = '0' & C2143 = '0' & C2619 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R57	C183 = '0' & C1047 = '0' & C2731 = '0' & C3213 = '1'	4	0.006	0	E2
R58	C1418 = '0' & C1589 = '0' & C2143 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R59	C1047 = '0' & C2037 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R60	C166 = '0' & C183 = '0' & C1418 = '0' & C3103 = '1'	4	0.006	0	E2
R61	C1047 = '0' & C2166 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R62	C1418 = '0' & C1780 = '0' & C2037 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R63	C166 = '0' & C1047 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R64	C183 = '0' & C1589 = '0' & C2166 = '0' & C3213 = '1'	4	0.006	0	E2
R65	C1418 = '0' & C2619 = '0' & C2740 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R66	C1047 = '0' & C2655 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R67	C1047 = '0' & C2451 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R68	C1047 = '0' & C1585 = '1' & C2143 = '0' & C3197 = '0'	4	0.006	0	E2
R69	C1418 = '0' & C1585 = '1' & C1780 = '0' & C2166 = '0' & C3197 = '0'	5	0.006	0	E2
R70	C342 = '0' & C411 = '0' & C2619 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R71	C1070 = '0' & C1779 = '1' & C2143 = '0' & C2619 = '0' & C3103 = '1' & C3197 = '0'	6	0.006	0	E2



Rule ID	Rules	Length	Freq.	Error	Class ID
R72	C183 = '0' & C342 = '0' & C1047 = '0' & C3213 = '1'	4	0.006	0	E2
R73	C1418 = '0' & C1589 = '0' & C2655 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R74	C411 = '0' & C1779 = '1' & C2143 = '0' & C2619 = '0' & C3197 = '0' & C3360 = '0'	6	0.006	0	E2
R75	C2619 = '0' & C2731 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R76	C182 = '0' & C1734 = '0' & C2740 = '0' & C3213 = '1'	4	0.006	0	E2
R77	C2451 = '0' & C2619 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R78	C183 = '0' & C1418 = '0' & C2731 = '0' & C3103 = '1'	4	0.006	0	E2
R79	C45 = '0' & C183 = '0' & C666 = '0' & C1589 = '0' & C3213 = '1'	5	0.006	0	E2
R80	C45 = '0' & C1589 = '0' & C1773 = '0' & C2143 = '0' & C3213 = '1'	5	0.006	0	E2
R81	C1418 = '0' & C2143 = '0' & C2619 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R82	C1589 = '0' & C2451 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R83	C45 = '0' & C666 = '0' & C1589 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R84	C411 = '0' & C2143 = '0' & C2619 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R85	C183 = '0' & C411 = '0' & C1589 = '0' & C2143 = '0' & C3213 = '1'	5	0.006	0	E2
R86	C1070 = '0' & C2619 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R87	C411 = '0' & C1429 = '0' & C1589 = '0' & C2143 = '0' & C3213 = '1'	5	0.006	0	E2
R88	C183 = '0' & C1070 = '0' & C2619 = '0' & C3213 = '1'	4	0.006	0	E2
R89	C1047 = '0' & C1070 = '0' & C1779 = '1' & C3103 = '1' & C3106 = '0' & C3117 = '1' & C3197 = '0'	7	0.006	0	E2

Rule ID	Rules	Length	Freq.	Error	Class ID
R90	C1418 = '0' & C2655 = '0' & C3103 = '1' & C3360 = '0'	4	0.006	0	E2
R91	C182 = '0' & C2619 = '0' & C2731 = '0' & C3213 = '1'	4	0.006	0	E2
R92	C183 = '0' & C1047 = '0' & C1070 = '0' & C3213 = '1'	4	0.006	0	E2
R93	C1047 = '0' & C1418 = '0' & C1866 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R94	C182 = '0' & C1047 = '0' & C2143 = '0' & C3213 = '1'	4	0.006	0	E2
R95	C1047 = '0' & C1779 = '1' & C2143 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R96	C183 = '0' & C1418 = '0' & C2451 = '0' & C3103 = '1'	4	0.006	0	E2
R97	C183 = '0' & C1418 = '0' & C2655 = '0' & C3103 = '1'	4	0.006	0	E2
R98	C183 = '0' & C1418 = '0' & C2740 = '0' & C3103 = '1'	4	0.006	0	E2
R99	C183 = '0' & C1047 = '0' & C2037 = '0' & C3213 = '1'	4	0.006	0	E2
R100	C1418 = '0' & C2619 = '0' & C2655 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R101	C183 = '0' & C411 = '0' & C666 = '0' & C2619 = '0' & C3213 = '1'	5	0.006	0	E2
R102	C1418 = '0' & C2037 = '0' & C2619 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R103	C1047 = '0' & C1418 = '0' & C2037 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R104	C183 = '0' & C1585 = '1' & C2143 = '0' & C2619 = '0' & C2868 = '0'	5	0.006	0	E2
R105	C342 = '0' & C1047 = '0' & C1585 = '1' & C3197 = '0' & C3306 = '0'	5	0.006	0	E2
R106	C183 = '0' & C2451 = '0' & C2619 = '0' & C3213 = '1'	4	0.006	0	E2
R107	C183 = '0' & C1070 = '0' & C1418 = '0' & C3103 = '1'	4	0.006	0	E2
R108	C183 = '0' & C1070 = '0' & C1589 = '0' & C3213 = '1'	4	0.006	0	E2
R109	C183 = '0' & C1126 = '0' & C1418 = '0' & C3103 = '1'	4	0.006	0	E2
R110	C183 = '0' & C1418 = '0' & C2143 = '0' & C3213 = '1'	4	0.006	0	E2

Rule ID	Rules	Length	Freq.	Error	Class ID
R111	C1070 = '0' & C1501 = '0' & C1531 = '0' & C1589 = '0' & C1779 = '1' & C3197 = '0' & C3323 = '0' & C3360 = '0'	8	0.006	0	E2
R112	C1418 = '0' & C2619 = '0' & C2740 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R113	C1047 = '0' & C2143 = '0' & C3205 = '0' & C3213 = '1'	4	0.006	0	E2
R114	C183 = '0' & C1501 = '0' & C1585 = '1' & C2037 = '0' & C2143 = '0' & C2619 = '0' & C3103 = '1'	7	0.006	0	E2
R115	C1047 = '0' & C1585 = '1' & C2143 = '0' & C3171 = '0'	4	0.006	0	E2
R116	C183 = '0' & C411 = '0' & C1585 = '1' & C1589 = '0' & C2143 = '0' & C3103 = '1' & C3106 = '0'	7	0.006	0	E2
R117	C183 = '0' & C2166 = '0' & C2619 = '0' & C3213 = '1'	4	0.006	0	E2
R118	C411 = '0' & C1589 = '0' & C1779 = '1' & C2143 = '0' & C3031 = '0' & C3197 = '0'	6	0.006	0	E2
R119	C1963 = '0' & C2166 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R120	C1047 = '0' & C1779 = '1' & C2143 = '0' & C3197 = '0' & C3353 = '0'	5	0.006	0	E2
R121	C183 = '0' & C1047 = '0' & C1531 = '0' & C1585 = '1' & C1779 = '1' & C2143 = '0'	6	0.006	0	E2
R122	C411 = '0' & C1589 = '0' & C2740 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R123	C183 = '0' & C411 = '0' & C2143 = '0' & C2619 = '0' & C3213 = '1'	5	0.006	0	E2
R124	C45 = '0' & C1589 = '0' & C2143 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R125	C411 = '0' & C1126 = '0' & C1585 = '1' & C1780 = '0' & C2619 = '0' & C3197 = '0'	6	0.006	0	E2
R126	C1418 = '0' & C1780 = '0' & C1866 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2

Rule ID	Rules	Length	Freq.	Error	Class ID
R127	C183 = '0' & C1126 = '0' & C1589 = '0' & C2811 = '0' & C3213 = '1'	5	0.006	0	E2
R128	C1047 = '0' & C1779 = '1' & C2143 = '0' & C3103 = '1' & C3106 = '0' & C3205 = '0' & C3323 = '0'	7	0.006	0	E2
R129	C45 = '0' & C558 = '0' & C1585 = '1' & C1589 = '0' & C1773 = '0' & C1779 = '1' & C2143 = '0'	7	0.006	0	E2
R130	C183 = '0' & C411 = '0' & C1126 = '0' & C1589 = '0' & C3213 = '1'	5	0.006	0	E2
R131	C13 = '0' & C1866 = '0' & C1963 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R132	C183 = '0' & C1418 = '0' & C2740 = '0' & C3213 = '1'	4	0.006	0	E2
R133	C183 = '0' & C1501 = '0' & C1531 = '0' & C1589 = '0' & C1779 = '1' & C2731 = '0' & C3197 = '0' & C3306 = '0' & C3323 = '0'	9	0.006	0	E2
R134	C183 = '0' & C1418 = '0' & C1531 = '0' & C1585 = '1' & C1779 = '1' & C2143 = '0'	6	0.006	0	E2
R135	C1585 = '1' & C2166 = '0' & C2619 = '0' & C3103 = '1' & C3106 = '0' & C3159 = '0'	6	0.006	0	E2
R136	C1070 = '0' & C2619 = '0' & C3159 = '0' & C3213 = '1'	4	0.006	0	E2
R137	C1418 = '0' & C1963 = '0' & C2166 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R138	C1047 = '0' & C1126 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R139	C411 = '0' & C666 = '0' & C2619 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R140	C183 = '0' & C1589 = '0' & C2451 = '0' & C3213 = '1'	4	0.006	0	E2
R141	C1047 = '0' & C1418 = '0' & C2166 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R142	C1047 = '0' & C1531 = '0' & C1779 = '1' & C2451 = '0' & C3117 = '1' & C3197 = '0' & C3306 = '0'	7	0.006	0	E2
R143	C182 = '0' & C1047 = '0' & C2740 = '0' & C3213 = '1'	4	0.006	0	E2

Rule ID	Rules	Length	Freq.	Error	Class ID
R144	C183 = '0' & C666 = '0' & C1047 = '0' & C3213 = '1'	4	0.006	0	E2
R145	C1585 = '1' & C1589 = '0' & C2037 = '0' & C3117 = '1' & C3197 = '0'	5	0.006	0	E2
R146	C1047 = '0' & C1070 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R147	C183 = '0' & C666 = '0' & C1418 = '0' & C3103 = '1'	4	0.006	0	E2
R148	C182 = '0' & C1418 = '0' & C1780 = '0' & C2037 = '0' & C3103 = '1'	5	0.006	0	E2
R149	C183 = '0' & C1501 = '0' & C1585 = '1' & C2143 = '0' & C2619 = '0' & C2703 = '0' & C3103 = '1'	7	0.006	0	E2
R150	C183 = '0' & C1126 = '0' & C1418 = '0' & C3213 = '1'	4	0.006	0	E2
R151	C666 = '0' & C1418 = '0' & C2619 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R152	C1126 = '0' & C1418 = '0' & C1585 = '1' & C2143 = '0' & C3197 = '0'	5	0.006	0	E2
R153	C182 = '0' & C1047 = '0' & C1126 = '0' & C3213 = '1'	4	0.006	0	E2
R154	C45 = '0' & C183 = '0' & C1589 = '0' & C2740 = '0' & C3213 = '1'	5	0.006	0	E2
R155	C45 = '0' & C183 = '0' & C1589 = '0' & C2143 = '0' & C3213 = '1'	5	0.006	0	E2
R156	C1418 = '0' & C2655 = '0' & C3103 = '1' & C3159 = '0'	4	0.006	0	E2
R157	C1501 = '0' & C1531 = '0' & C1779 = '1' & C2166 = '0' & C2619 = '0' & C3117 = '1' & C3197 = '0' & C3353 = '0'	8	0.006	0	E2
R158	C1418 = '0' & C1585 = '1' & C2143 = '0' & C2619 = '0' & C3197 = '0'	5	0.006	0	E2
R159	C183 = '0' & C342 = '0' & C1418 = '0' & C3103 = '1'	4	0.006	0	E2
R160	C1734 = '0' & C2451 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R161	C1047 = '0' & C1779 = '1' & C2166 = '0' & C3159 = '0' & C3306 = '0'	5	0.006	0	E2

Rule ID	Rules	Length	Freq.	Error	Class ID
R162	C166 = '0' & C1418 = '0' & C2619 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R163	C45 = '0' & C183 = '0' & C1501 = '0' & C1585 = '1' & C2143 = '0' & C2619 = '0' & C3103 = '1'	7	0.006	0	E2
R164	C182 = '0' & C1418 = '0' & C1589 = '0' & C2143 = '0' & C3103 = '1'	5	0.006	0	E2
R165	C183 = '0' & C1418 = '0' & C2451 = '0' & C3213 = '1'	4	0.006	0	E2
R166	C1418 = '0' & C1589 = '0' & C2166 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R167	C1418 = '0' & C2037 = '0' & C2356 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R168	C1047 = '0' & C1660 = '0' & C2143 = '0' & C3213 = '1'	4	0.006	0	E2
R169	C1418 = '0' & C2166 = '0' & C2619 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R170	C1070 = '0' & C2619 = '0' & C3171 = '0' & C3213 = '1'	4	0.006	0	E2
R171	C45 = '0' & C166 = '0' & C1589 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R172	C45 = '0' & C2143 = '0' & C2619 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R173	C182 = '0' & C1418 = '0' & C2143 = '0' & C2619 = '0' & C3103 = '1'	5	0.006	0	E2
R174	C183 = '0' & C342 = '0' & C1418 = '0' & C3213 = '1'	4	0.006	0	E2
R175	C1672 = '0' & C1963 = '0' & C2143 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R176	C45 = '0' & C1387 = '0' & C1779 = '1' & C1866 = '0' & C2619 = '0' & C3031 = '0' & C3117 = '1' & C3197 = '0'	8	0.006	0	E2
R177	C1047 = '0' & C1773 = '0' & C2143 = '0' & C3213 = '1'	4	0.006	0	E2

Rule ID	Rules	Length	Freq.	Error	Class ID
R178	C342 = '0' & C1418 = '0' & C1734 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R179	C1418 = '0' & C1721 = '0' & C2143 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R180	C666 = '0' & C1047 = '0' & C1418 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R181	C1126 = '0' & C1418 = '0' & C3103 = '1' & C3159 = '0'	4	0.006	0	E2
R182	C183 = '0' & C2619 = '0' & C2731 = '0' & C3213 = '1'	4	0.006	0	E2
R183	C411 = '0' & C1585 = '1' & C2143 = '0' & C2619 = '0' & C3197 = '0'	5	0.006	0	E2
R184	C183 = '0' & C1047 = '0' & C1126 = '0' & C3213 = '1'	4	0.006	0	E2
R185	C183 = '0' & C1501 = '0' & C1779 = '1' & C2143 = '0' & C2166 = '0' & C2619 = '0' & C3103 = '1' & C3323 = '0'	8	0.006	0	E2
R186	C342 = '0' & C1418 = '0' & C2619 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R187	C183 = '0' & C1047 = '0' & C1779 = '1' & C2143 = '0' & C3103 = '1' & C3106 = '0' & C3323 = '0'	7	0.006	0	E2
R188	C1418 = '0' & C1721 = '0' & C2731 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R189	C183 = '0' & C1418 = '0' & C1585 = '1' & C2143 = '0' & C3306 = '0'	5	0.006	0	E2
R190	C1418 = '0' & C1589 = '0' & C1866 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R191	C1047 = '0' & C2740 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R192	C2451 = '0' & C2619 = '0' & C3171 = '0' & C3213 = '1'	4	0.006	0	E2
R193	C1418 = '0' & C1429 = '0' & C1589 = '0' & C2143 = '0' & C3103 = '1'	5	0.006	0	E2

Rule ID	Rules	Length	Freq.	Error	Class ID
R194	C45 = '0' & C1531 = '0' & C1585 = '1' & C1589 = '0' & C2731 = '0' & C3103 = '1' & C3106 = '0' & C3360 = '0'	8	0.006	0	E2
R195	C183 = '0' & C1047 = '0' & C1585 = '1' & C2143 = '0' & C2868 = '0'	5	0.006	0	E2
R196	C183 = '0' & C1418 = '0' & C2037 = '0' & C3213 = '1'	4	0.006	0	E2
R197	C1585 = '1' & C2143 = '0' & C2451 = '0' & C2619 = '0' & C3197 = '0'	5	0.006	0	E2
R198	C183 = '0' & C1418 = '0' & C2166 = '0' & C3103 = '1'	4	0.006	0	E2
R199	C1734 = '0' & C2143 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R200	C411 = '0' & C1779 = '1' & C2143 = '0' & C2619 = '0' & C3197 = '0' & C3353 = '0'	6	0.006	0	E2
R201	C1418 = '0' & C1721 = '0' & C2166 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R202	C183 = '0' & C2037 = '0' & C2619 = '0' & C3213 = '1'	4	0.006	0	E2
R203	C342 = '0' & C1418 = '0' & C3103 = '1' & C3171 = '0'	4	0.006	0	E2
R204	C182 = '0' & C1418 = '0' & C1585 = '1' & C2143 = '0' & C2619 = '0'	5	0.006	0	E2
R205	C183 = '0' & C1418 = '0' & C1866 = '0' & C3103 = '1'	4	0.006	0	E2
R206	C1047 = '0' & C1126 = '0' & C1585 = '1' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R207	C1418 = '0' & C2451 = '0' & C2619 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R208	C342 = '0' & C1047 = '0' & C1585 = '1' & C3106 = '0' & C3117 = '1' & C3197 = '0'	6	0.006	0	E2
R209	C45 = '0' & C1126 = '0' & C2619 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R210	C183 = '0' & C2143 = '0' & C2619 = '0' & C2811 = '0' & C3213 = '1'	5	0.006	0	E2
R211	C183 = '0' & C1963 = '0' & C2451 = '0' & C3213 = '1'	4	0.006	0	E2



Rule ID	Rules	Length	Freq.	Error	Class ID
R212	C166 = '0' & C1047 = '0' & C1418 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R213	C183 = '0' & C1418 = '0' & C2731 = '0' & C3213 = '1'	4	0.006	0	E2
R214	C1672 = '0' & C1773 = '0' & C2143 = '0' & C2619 = '0' & C3213 = '1'	5	0.006	0	E2
R215	C183 = '0' & C1418 = '0' & C1779 = '1' & C2143 = '0' & C3031 = '0'	5	0.006	0	E2
R216	C1585 = '1' & C1589 = '0' & C2451 = '0' & C3117 = '1' & C3197 = '0'	5	0.006	0	E2
R217	C45 = '0' & C1585 = '1' & C2143 = '0' & C2619 = '0' & C3197 = '0'	5	0.006	0	E2
R218	C183 = '0' & C1047 = '0' & C1866 = '0' & C3213 = '1'	4	0.006	0	E2
R219	C166 = '0' & C1418 = '0' & C3103 = '1' & C3159 = '0'	4	0.006	0	E2
R220	C183 = '0' & C411 = '0' & C1126 = '0' & C2619 = '0' & C3213 = '1'	5	0.006	0	E2
R221	C1047 = '0' & C1418 = '0' & C1429 = '0' & C2143 = '0' & C3103 = '1'	5	0.006	0	E2
R222	C166 = '0' & C1418 = '0' & C1589 = '0' & C3103 = '1' & C3197 = '0'	5	0.006	0	E2
R223	C182 = '0' & C1047 = '0' & C1418 = '0' & C2143 = '0' & C3103 = '1'	5	0.006	0	E2
R224	C342 = '0' & C1047 = '0' & C3197 = '0' & C3213 = '1'	4	0.006	0	E2
R225	C182 = '0' & C1047 = '0' & C1418 = '0' & C1866 = '0' & C3103 = '1'	5	0.006	0	E2
R226	C45 = '0' & C1773 = '0' & C2143 = '0' & C2619 = '0' & C3213 = '1'	5	0.006	0	E2
R227	C1047 = '0' & C1585 = '1' & C2037 = '0' & C3103 = '1' & C3159 = '0' & C3197 = '0'	6	0.006	0	E2
R228	C342 = '0' & C1418 = '0' & C1589 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R229	C1047 = '0' & C2143 = '0' & C3213 = '1' & C3360 = '0'	4	0.006	0	E2

Rule ID	Rules	Length	Freq.	Error	Class ID
R230	C45 = '0' & C2619 = '0' & C2655 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R231	C411 = '0' & C1589 = '0' & C2655 = '0' & C3197 = '0' & C3213 = '1'	5	0.006	0	E2
R232	C183 = '1' & C653 = '1' & C2037 = '0' & C2237 = '0' & C2311 = '0' & C2629 = '0' & C3106 = '0'	7	0.051	0.009	E3
R233	C1065 = '1' & C1126 = '1' & C3159 = '0'	3	0.018	0	E3
R234	C1227 = '0' & C2341 = '1' & C3106 = '0'	3	0.009	0	E3
R235	C1065 = '1' & C2158 = '1' & C3159 = '0'	3	0.018	0	E3
R236	C1065 = '1' & C2496 = '1' & C3170 = '1'	3	0.02	0	E3
R237	C653 = '1' & C1585 = '0' & C2655 = '0' & C3106 = '0' & C3154 = '1'	5	0.05	0.019	E3
R238	C183 = '1' & C653 = '1' & C1672 = '0' & C1780 = '1' & C2237 = '0'	5	0.035	0	E3
R239	C653 = '1' & C1189 = '1' & C1996 = '0' & C2659 = '0' & C3106 = '0'	5	0.05	0.019	E3
R240	C183 = '1' & C1065 = '1' & C1429 = '0'	3	0.023	0.02	E3
R241	C1065 = '1' & C3159 = '0' & C3353 = '1'	3	0.018	0	E3
R242	C183 = '1' & C1065 = '1' & C3159 = '0'	3	0.018	0	E3
R243	C1227 = '0' & C1501 = '0' & C2341 = '1'	3	0.009	0	E3
R244	C1065 = '1' & C1660 = '0' & C3106 = '0' & C3170 = '1'	4	0.022	0	E3
R245	C183 = '1' & C653 = '1' & C1501 = '0' & C2037 = '0' & C2237 = '0' & C2356 = '0' & C3237 = '0'	7	0.016	0	E3
R246	C183 = '1' & C1065 = '1' & C1531 = '0' & C1773 = '0'	4	0.021	0	E3
R247	C1065 = '1' & C3154 = '1' & C3159 = '0'	3	0.018	0	E3
R248	C1501 = '0' & C1531 = '0' & C1672 = '1' & C2619 = '0' & C2927 = '1' & C3237 = '0'	6	0.01	0	E3
R249	C316 = '1' & C1779 = '0' & C2412 = '0' & C2479 = '0' & C3106 = '0' & C3205 = '0' & C3237 = '0'	7	0.049	0.019	E3
R250	C653 = '1' & C1189 = '1' & C1996 = '0' & C3159 = '1'	4	0.035	0.013	E3

Rule ID	Rules	Length	Freq.	Error	Class ID
R251	C183 = '1' & C653 = '1' & C1507 = '0' & C2037 = '0' & C2237 = '0' & C2311 = '0' & C3106 = '0'	7	0.051	0.009	E3
R252	C183 = '1' & C653 = '1' & C1996 = '0' & C2237 = '0' & C3159 = '1'	5	0.035	0	E3
R253	C183 = '1' & C653 = '1' & C2037 = '0' & C2237 = '0' & C2629 = '0' & C3106 = '0' & C3237 = '0'	7	0.05	0	E3
R254	C1501 = '0' & C1531 = '0' & C1773 = '1' & C2166 = '0' & C2619 = '0' & C3154 = '0' & C3170 = '1' & C3237 = '0'	8	0.009	0	E3
R255	C1780 = '0' & C2164 = '0' & C2619 = '1' & C3103 = '0' & C3197 = '0'	5	0.012	0	E6
R256	C183 = '1' & C1280 = '0' & C1531 = '0' & C2311 = '0' & C2619 = '1' & C3323 = '0'	6	0.014	0	E6
R257	C183 = '1' & C200 = '0' & C1531 = '0' & C2311 = '0' & C2619 = '1' & C3323 = '0'	6	0.013	0	E6
R258	C183 = '1' & C1280 = '0' & C1531 = '0' & C2619 = '1' & C3192 = '0' & C3323 = '0'	6	0.014	0	E6
R259	C183 = '1' & C1197 = '0' & C1531 = '0' & C2619 = '1' & C3192 = '0' & C3323 = '0'	6	0.014	0	E6
R260	C1418 = '1' & C2619 = '1' & C2655 = '0' & C3323 = '0'	4	0.011	0	E6
R261	C555 = '1' & C1418 = '1' & C1756 = '0' & C3197 = '0'	4	0.013	0	E6
R262	C183 = '0' & C1963 = '0' & C2143 = '0' & C2619 = '1' & C3323 = '0'	5	0.012	0	E6
R263	C183 = '0' & C2143 = '0' & C2566 = '0' & C2619 = '1' & C3323 = '0'	5	0.016	0	E6
R264	C183 = '1' & C1197 = '0' & C1531 = '0' & C1890 = '0' & C2619 = '1' & C3323 = '0'	6	0.014	0	E6
R265	C1418 = '1' & C2341 = '0' & C2619 = '1' & C3323 = '0'	4	0.016	0	E6
R266	C1389 = '0' & C2619 = '1' & C3106 = '0' & C3197 = '1' & C3323 = '0'	5	0.012	0	E6

Rule ID	Rules	Length	Freq.	Error	Class ID
R267	C183 = '1' & C1331 = '0' & C1531 = '0' & C1890 = '0' & C2619 = '1' & C3323 = '0'	6	0.014	0	E6
R268	C1963 = '1' & C2566 = '0' & C3197 = '0' & C3323 = '0'	4	0.011	0	E6
R269	C183 = '0' & C2143 = '0' & C2619 = '1' & C3101 = '0' & C3323 = '0'	5	0.016	0	E6
R270	C1724 = '1' & C1780 = '0' & C2164 = '0' & C2619 = '1'	4	0.014	0	E6
R271	C1331 = '0' & C1501 = '0' & C1773 = '1' & C2619 = '1' & C3323 = '0'	5	0.011	0	E6
R272	C183 = '0' & C1589 = '1' & C2566 = '0' & C2619 = '1' & C3323 = '0'	5	0.015	0	E6
R273	C2619 = '1' & C2655 = '0' & C3103 = '1' & C3323 = '0'	4	0.012	0	E6
R274	C45 = '1' & C411 = '0' & C1418 = '1' & C1756 = '0' & C3197 = '0'	5	0.013	0	E6
R275	C183 = '1' & C1531 = '0' & C2311 = '0' & C2619 = '1' & C2817 = '0' & C3323 = '0'	6	0.013	0	E6
R276	C183 = '1' & C1197 = '0' & C1531 = '0' & C2311 = '0' & C2619 = '1' & C3323 = '0'	6	0.014	0	E6
R277	C183 = '1' & C1287 = '0' & C1589 = '1' & C2703 = '0' & C2844 = '1' & C2851 = '0' & C3106 = '0'	7	0.011	0	E6
R278	C2341 = '0' & C2619 = '1' & C3103 = '1' & C3323 = '0'	4	0.017	0	E6
R279	C183 = '1' & C1531 = '0' & C2619 = '1' & C2817 = '0' & C3192 = '0' & C3323 = '0'	6	0.013	0	E6
R280	C45 = '1' & C1531 = '0' & C1756 = '0' & C1890 = '0' & C3197 = '0' & C3237 = '0'	6	0.014	0	E6
R281	C555 = '1' & C1756 = '0' & C3103 = '1' & C3197 = '0'	4	0.013	0	E6
R282	C1531 = '0' & C1734 = '1' & C2164 = '0' & C2619 = '1' & C3323 = '0'	5	0.021	0	E6

Rule ID	Rules	Length	Freq.	Error	Class ID
R283	C2143 = '0' & C2619 = '1' & C3101 = '0' & C3323 = '0' & C3360 = '0'	5	0.015	0	E6
R284	C1418 = '0' & C1780 = '0' & C2164 = '0' & C2619 = '1' & C3197 = '0'	5	0.012	0	E6
R285	C1197 = '0' & C1501 = '0' & C2619 = '1' & C3171 = '1' & C3197 = '1' & C3323 = '0'	6	0.011	0	E6
R286	C1589 = '1' & C2164 = '0' & C2619 = '1' & C3360 = '0'	4	0.013	0	E6
R287	C2496 = '1' & C2566 = '0' & C2619 = '1' & C3106 = '0' & C3323 = '0'	5	0.016	0	E6
R288	C183 = '1' & C1531 = '0' & C1890 = '0' & C2619 = '1' & C2817 = '0' & C3323 = '0'	6	0.014	0	E6
R289	C183 = '0' & C1047 = '1' & C1287 = '0' & C1418 = '0' & C1501 = '0' & C2153 = '0' & C2566 = '0' & C2868 = '0' & C3323 = '0'	9	0.011	0	E6
R290	C183 = '0' & C1047 = '1' & C1963 = '0' & C2619 = '1'	4	0.011	0	E6
R291	C183 = '0' & C1189 = '0' & C1418 = '0' & C1585 = '0' & C2164 = '0' & C2619 = '1'	6	0.013	0	E6
R292	C1189 = '0' & C1589 = '1' & C2164 = '0' & C2619 = '1' & C2631 = '0'	5	0.023	0	E6
R293	C45 = '1' & C1756 = '0' & C2868 = '0' & C3103 = '1' & C3360 = '0'	5	0.015	0	E6
R294	C183 = '0' & C555 = '1' & C1756 = '0' & C3103 = '1' & C3323 = '0'	5	0.013	0	E6
R295	C45 = '1' & C1418 = '1' & C1756 = '0' & C2844 = '0' & C3197 = '0'	5	0.012	0	E6
R296	C13 = '0' & C45 = '1' & C1756 = '0' & C3103 = '1' & C3197 = '0'	5	0.013	0	E6
R297	C1672 = '1' & C2792 = '1' & C3323 = '0'	3	0.003	0	E8
R298	C1507 = '1' & C2792 = '1' & C3323 = '0'	3	0.003	0	E8
R299	C183 = '0' & C2792 = '1' & C3197 = '1' & C3323 = '0'	4	0.003	0	E8

Rule ID	Rules	Length	Freq.	Error	Class ID
R300	C1531 = '0' & C2792 = '1' & C3197 = '1' & C3323 = '0'	4	0.003	0	E8
R301	C1429 = '0' & C2792 = '1' & C3197 = '1' & C3323 = '0'	4	0.003	0	E8
R302	C182 = '0' & C2792 = '1' & C3197 = '1' & C3323 = '0'	4	0.003	0	E8
R303	C2143 = '0' & C2164 = '0' & C2356 = '0' & C2792 = '1' & C2844 = '0' & C3103 = '0' & C3237 = '0'	7	0.004	0	E8
R304	C281 = '1'	1	0.004	0	E10
R305	C45 = '0' & C1721 = '0' & C1771 = '0' & C1913 = '1' & C2037 = '1' & C3197 = '0'	6	0.001	0	E10
R306	C558 = '0' & C1913 = '1' & C2037 = '1' & C2143 = '0' & C3197 = '0'	5	0.001	0	E10
R307	C539 = '0' & C1227 = '1' & C2037 = '1' & C2731 = '1' & C2868 = '0' & C3013 = '0' & C3106 = '0' & C3197 = '0'	8	0.001	0	E10
R308	C1585 = '0' & C1779 = '1' & C3106 = '1' & C3197 = '0'	4	0.015	0	E11
R309	C967 = '1'	1	0.046	0.01	E11
R310	C967 = '1' & C2143 = '0'	2	0.02	0	E11
R311	C967 = '1' & C2479 = '0'	2	0.046	0	E11
R312	C967 = '1' & C1387 = '0'	2	0.03	0	E11
R313	C1387 = '1' & C1734 = '0' & C1780 = '0' & C2817 = '0' & C3197 = '1'	5	0.027	0	E11
R314	C967 = '1' & C2933 = '0'	2	0.04	0	E11
R315	C183 = '1' & C1387 = '1' & C1672 = '0' & C3237 = '0' & C3323 = '0'	5	0.019	0	E11
R316	C967 = '1' & C2731 = '0'	2	0.041	0	E11
R317	C1501 = '1' & C1531 = '1' & C1734 = '0' & C3192 = '0'	4	0.059	0.008	E11
R318	C967 = '1' & C3197 = '1'	2	0.019	0	E11
R319	C1531 = '1' & C1734 = '0' & C3106 = '1' & C3197 = '0'	4	0.015	0	E11

Rule ID	Rules	Length	Freq.	Error	Class ID
R320	C183 = '1' & C1387 = '1' & C2412 = '0' & C3237 = '0' & C3323 = '0'	5	0.024	0	E11
R321	C539 = '1' & C1771 = '0' & C2496 = '0' & C2655 = '0' & C3197 = '1' & C3237 = '0'	6	0.025	0	E11
R322	C1779 = '1' & C3106 = '1' & C3192 = '0' & C3197 = '0'	4	0.015	0	E11
R323	C1387 = '1' & C1429 = '1' & C3170 = '0' & C3237 = '0'	4	0.022	0	E11
R324	C1387 = '1' & C1773 = '1' & C2412 = '0' & C3237 = '0'	4	0.02	0	E11
R325	C967 = '1' & C3360 = '0'	2	0.025	0	E11
R326	C1531 = '1' & C1890 = '0' & C2703 = '0' & C3106 = '1' & C3197 = '0'	5	0.017	0	E11
R327	C1387 = '1' & C2158 = '0' & C3205 = '1' & C3237 = '0'	4	0.022	0	E11
R328	C1387 = '1' & C1660 = '1' & C3170 = '0' & C3237 = '0'	4	0.021	0	E11
R329	C1779 = '1' & C2311 = '0' & C3106 = '1' & C3197 = '0'	4	0.015	0	E11
R330	C1387 = '1' & C1660 = '1' & C2158 = '0' & C3237 = '0'	4	0.02	0	E11
R331	C1531 = '1' & C1852 = '0' & C3106 = '1' & C3189 = '0' & C3197 = '0'	5	0.017	0	E11
R332	C183 = '1' & C1387 = '1' & C3197 = '1' & C3237 = '0' & C3353 = '0'	5	0.022	0	E11
R333	C183 = '1' & C1387 = '1' & C1771 = '0' & C3197 = '1' & C3237 = '0'	5	0.021	0	E11
R334	C183 = '1' & C1387 = '1' & C1771 = '0' & C3237 = '0' & C3323 = '0'	5	0.024	0	E11
R335	C183 = '1' & C1387 = '1' & C2412 = '0' & C3197 = '1' & C3237 = '0'	5	0.021	0	E11
R336	C183 = '0' & C1263 = '0' & C1387 = '1' & C1501 = '1'	4	0.015	0	E11

Rule ID	Rules	Length	Freq.	Error	Class ID
R337	C182 = '0' & C316 = '0' & C1148 = '0' & C1531 = '1' & C2619 = '0' & C2933 = '0' & C3159 = '1'	7	0.015	0	E11
R338	C183 = '1' & C1387 = '1' & C1996 = '0' & C3197 = '1' & C3237 = '0'	5	0.026	0	E11
R339	C183 = '1' & C1387 = '1' & C1585 = '0' & C2851 = '0' & C3237 = '0' & C3323 = '0'	6	0.017	0	E11
R340	C1387 = '1' & C3170 = '0' & C3205 = '1' & C3237 = '0'	4	0.023	0	E11
R341	C1387 = '1' & C3031 = '0' & C3170 = '0' & C3197 = '1' & C3237 = '0'	5	0.028	0	E11
R342	C967 = '1' & C1779 = '0'	2	0.027	0	E11
R343	C967 = '1' & C2496 = '0'	2	0.041	0	E11
R344	C183 = '1' & C1387 = '1' & C2158 = '0' & C3197 = '1' & C3237 = '0'	5	0.021	0	E11
R345	C183 = '1' & C1227 = '0' & C1387 = '1' & C1589 = '0' & C1672 = '0'	5	0.017	0	E11
R346	C1501 = '1' & C1779 = '1' & C3154 = '0' & C3197 = '0'	4	0.015	0	E11
R347	C539 = '1' & C1531 = '1' & C1585 = '1' & C1734 = '0'	4	0.015	0	E11
R348	C1501 = '1' & C1531 = '1' & C1890 = '0' & C2166 = '0' & C3197 = '0'	5	0.016	0	E11
R349	C182 = '1' & C1501 = '1' & C1531 = '1' & C2143 = '0'	4	0.02	0	E11
R350	C967 = '1' & C1531 = '1'	2	0.023	0	E11
R351	C1227 = '0' & C2143 = '0' & C3106 = '1' & C3205 = '1' & C3237 = '0' & C3323 = '0'	6	0.019	0	E11
R352	C1779 = '1' & C3106 = '1' & C3197 = '0' & C3205 = '0'	4	0.015	0	E11
R353	C183 = '1' & C1387 = '1' & C2356 = '1' & C3237 = '0' & C3323 = '0'	5	0.015	0	E11
R354	C1501 = '1' & C1531 = '0' & C1779 = '1' & C2311 = '0' & C3170 = '0' & C3473 = '0'	6	0.019	0	E11



Rule ID	Rules	Length	Freq.	Error	Class ID
R355	C98 = '0' & C1501 = '1' & C1531 = '1' & C2037 = '0' & C3197 = '0'	5	0.016	0	E11
R356	C1387 = '1' & C1660 = '1' & C2412 = '0' & C3237 = '0'	4	0.02	0	E11
R357	C183 = '1' & C1387 = '1' & C3170 = '0' & C3197 = '1' & C3237 = '0'	5	0.022	0	E11
R358	C1387 = '1' & C1996 = '0' & C3197 = '1' & C3237 = '0' & C3473 = '0'	5	0.027	0	E11
R359	C967 = '1' & C3171 = '0'	2	0.025	0	E11
R360	C967 = '1' & C1263 = '0'	2	0.032	0	E11
R361	C666 = '0' & C1531 = '1' & C1780 = '0' & C2166 = '0' & C3197 = '1'	5	0.047	0.01	E11
R362	C183 = '1' & C1227 = '0' & C1387 = '1' & C1589 = '0' & C1771 = '0'	5	0.027	0	E11
R363	C1387 = '1' & C1780 = '0' & C3106 = '1' & C3117 = '0'	4	0.021	0	E11
R364	C183 = '1' & C1387 = '1' & C1589 = '0' & C1672 = '0' & C2817 = '0' & C3323 = '0'	6	0.02	0	E11
R365	C539 = '1' & C1501 = '1' & C1531 = '0' & C1636 = '0' & C2703 = '0' & C3170 = '0'	6	0.02	0	E11
R366	C183 = '1' & C1531 = '1' & C3117 = '0' & C3306 = '1'	4	0.016	0	E11
R367	C1531 = '1' & C1773 = '1' & C2143 = '0' & C2703 = '0' & C2844 = '0'	5	0.022	0	E11
R368	C1501 = '1' & C1531 = '1' & C1890 = '0' & C2703 = '0' & C3197 = '0'	5	0.017	0	E11
R369	C1263 = '0' & C1387 = '1' & C1589 = '0' & C3323 = '0' & C3360 = '0'	5	0.02	0	E11
R370	C183 = '1' & C1531 = '1' & C2143 = '0' & C3106 = '1'	4	0.019	0	E11
R371	C45 = '0' & C2619 = '0' & C3159 = '0' & C3323 = '1'	4	0.046	0.01	E14
R372	C45 = '0' & C2619 = '0' & C2927 = '0' & C3138 = '0' & C3323 = '1'	5	0.054	0	E14

Rule ID	Rules	Length	Freq.	Error	Class ID
R373	C1047 = '0' & C3106 = '0' & C3159 = '0' & C3323 = '1'	4	0.049	0.01	E14
R374	C1047 = '0' & C1589 = '0' & C2619 = '0' & C3323 = '1'	4	0.052	0.009	E14
R375	C555 = '0' & C2619 = '0' & C2927 = '0' & C3192 = '0' & C3323 = '1'	5	0.059	0.008	E14
R376	C1189 = '1' & C2847 = '0' & C3323 = '1'	3	0.023	0	E14
R377	C1047 = '0' & C2619 = '0' & C3138 = '0' & C3323 = '1'	4	0.053	0	E14
R378	C1418 = '0' & C1589 = '0' & C1672 = '0' & C2164 = '0' & C3323 = '1'	5	0.037	0	E14
R379	C45 = '0' & C2143 = '0' & C2619 = '0' & C3323 = '1'	4	0.047	0.01	E14
R380	C45 = '0' & C2619 = '0' & C2731 = '0' & C3323 = '1'	4	0.051	0.009	E14
R381	C1047 = '0' & C2619 = '0' & C3159 = '0' & C3323 = '1'	4	0.043	0	E14
R382	C2158 = '0' & C3125 = '1' & C3323 = '1'	3	0.02	0	E14
R383	C555 = '0' & C2619 = '0' & C3159 = '0' & C3323 = '1'	4	0.046	0.01	E14
R384	C45 = '0' & C2619 = '0' & C3138 = '0' & C3323 = '1'	4	0.055	0.008	E14
R385	C1047 = '0' & C1780 = '0' & C2619 = '0' & C3323 = '1'	4	0.036	0	E14
R386	C2817 = '0' & C3125 = '1' & C3323 = '1'	3	0.02	0	E14
R387	C1103 = '1' & C2817 = '0' & C3323 = '1'	3	0.019	0	E14
R388	C1389 = '0' & C3323 = '1' & C3473 = '1'	3	0.021	0	E14
R389	C555 = '0' & C2143 = '0' & C2619 = '0' & C3323 = '1'	4	0.047	0.01	E14
R390	C555 = '0' & C1724 = '0' & C1780 = '0' & C2619 = '0' & C3323 = '1'	5	0.034	0	E14
R391	C1103 = '1' & C2927 = '0' & C3323 = '1'	3	0.019	0	E14
R392	C1418 = '0' & C2619 = '0' & C3138 = '0' & C3323 = '1'	4	0.043	0	E14
R393	C45 = '0' & C2619 = '0' & C3170 = '0' & C3205 = '0' & C3323 = '1'	5	0.05	0.009	E14

Rule ID	Rules	Length	Freq.	Error	Class ID
R394	C45 = '0' & C2619 = '0' & C3159 = '0' & C3197 = '0' & C3323 = '1'	5	0.038	0	E14
R395	C45 = '0' & C1501 = '0' & C1600 = '0' & C1734 = '0' & C2143 = '0' & C3197 = '0' & C3323 = '1'	7	0.035	0	E14
R396	C1189 = '1' & C3170 = '0' & C3323 = '1'	3	0.022	0	E14
R397	C1047 = '0' & C2356 = '0' & C2619 = '0' & C3323 = '1'	4	0.052	0.009	E14
R398	C555 = '0' & C1589 = '0' & C2619 = '0' & C3162 = '0' & C3323 = '1'	5	0.052	0.009	E14
R399	C1047 = '0' & C1589 = '0' & C1779 = '0' & C3170 = '0' & C3323 = '1'	5	0.054	0	E14
R400	C555 = '0' & C1779 = '0' & C2619 = '0' & C3159 = '0' & C3323 = '1'	5	0.042	0	E14
R401	C539 = '0' & C1189 = '1' & C3323 = '1'	3	0.024	0	E14
R402	C45 = '0' & C1779 = '0' & C2158 = '0' & C2619 = '0' & C3323 = '1'	5	0.052	0.009	E14
R403	C1047 = '0' & C2143 = '0' & C3103 = '1' & C3323 = '1'	4	0.021	0	E14
R404	C2847 = '0' & C3125 = '1' & C3323 = '1'	3	0.02	0	E14
R405	C45 = '0' & C2619 = '0' & C3117 = '0' & C3159 = '0' & C3323 = '1'	5	0.042	0	E14
R406	C45 = '0' & C1779 = '0' & C2619 = '0' & C3170 = '0' & C3323 = '1'	5	0.052	0.009	E14
R407	C555 = '0' & C2619 = '0' & C3159 = '0' & C3197 = '0' & C3323 = '1'	5	0.038	0	E14
R408	C2619 = '0' & C3103 = '0' & C3138 = '0' & C3323 = '1'	4	0.043	0	E14
R409	C2158 = '0' & C2868 = '1' & C3323 = '1'	3	0.023	0	E14
R410	C1780 = '0' & C2619 = '0' & C3103 = '0' & C3323 = '1'	4	0.029	0	E14
R411	C1047 = '0' & C2164 = '0' & C2619 = '0' & C3323 = '1' & C3353 = '0'	5	0.052	0	E14

Rule ID	Rules	Length	Freq.	Error	Class ID
R412	C1418 = '0' & C1780 = '0' & C2619 = '0' & C3323 = '1'	4	0.029	0	E14
R413	C1047 = '0' & C1963 = '0' & C3159 = '0' & C3323 = '1'	4	0.051	0.009	E14
R414	C555 = '0' & C1070 = '0' & C1780 = '0' & C2619 = '0' & C3323 = '1'	5	0.033	0	E14
R415	C1418 = '0' & C1589 = '0' & C1600 = '0' & C2143 = '0' & C3323 = '1'	5	0.035	0	E14
R416	C1126 = '0' & C1189 = '1' & C3323 = '1'	3	0.021	0	E14
R417	C45 = '0' & C1589 = '0' & C2619 = '0' & C3197 = '0' & C3323 = '1'	5	0.036	0	E14
R418	C45 = '0' & C2619 = '0' & C3159 = '0' & C3306 = '0' & C3323 = '1'	5	0.032	0	E14
R419	C45 = '0' & C1734 = '0' & C2619 = '0' & C3162 = '0' & C3323 = '1'	5	0.05	0.009	E14
R420	C183 = '1' & C1501 = '1' & C1531 = '0' & C2432 = '1'	4	0.01	0	E15
R421	C653 = '1' & C1065 = '0' & C1621 = '0' & C1996 = '1'	4	0.01	0	E15
R422	C653 = '1' & C1065 = '0' & C1585 = '1' & C2906 = '0' & C3117 = '1'	5	0.01	0	E15
R423	C1126 = '1' & C1531 = '0' & C2432 = '1' & C3106 = '1'	4	0.01	0	E15
R424	C653 = '1' & C1065 = '0' & C1996 = '1' & C3117 = '1'	4	0.01	0	E15
R425	C309 = '1' & C1418 = '0' & C1531 = '0' & C2703 = '0' & C2785 = '0' & C3106 = '1' & C3170 = '1'	7	0.011	0	E15
R426	C309 = '1' & C2071 = '0' & C2237 = '1' & C3106 = '1'	4	0.013	0	E15
R427	C1672 = '0' & C2237 = '1' & C2655 = '1'	3	0.01	0	E15
R428	C183 = '1' & C1418 = '0' & C1501 = '1' & C1531 = '0' & C2158 = '1' & C2703 = '0' & C3125 = '0' & C3138 = '1'	8	0.014	0	E15
R429	C309 = '1' & C1501 = '1' & C2185 = '0' & C2432 = '1'	4	0.01	0	E15
R430	C183 = '1' & C1531 = '0' & C2432 = '1' & C3106 = '1'	4	0.01	0	E15
R431	C309 = '1' & C1429 = '0' & C1501 = '1' & C2237 = '1'	4	0.011	0	E15

Rule ID	Rules	Length	Freq.	Error	Class ID
R432	C653 = '1' & C1065 = '0' & C1996 = '1' & C3171 = '1'	4	0.01	0	E15
R433	C653 = '1' & C1065 = '0' & C1347 = '1' & C1621 = '0'	4	0.01	0	E15
R434	C326 = '0' & C1283 = '0' & C1672 = '1' & C1734 = '1' & C1996 = '1'	5	0.011	0	E15
R435	C183 = '0' & C1387 = '0' & C1501 = '1' & C1589 = '0' & C2143 = '1' & C3031 = '1' & C3125 = '0' & C3162 = '0'	8	0.012	0	E15
R436	C166 = '1' & C182 = '1' & C183 = '1' & C1501 = '1' & C1531 = '0'	5	0.01	0	E15
R437	C653 = '1' & C1065 = '0' & C1996 = '1' & C2237 = '0'	4	0.01	0	E15
R438	C109 = '0' & C166 = '1' & C183 = '1' & C1531 = '0' & C3106 = '1'	5	0.01	0	E15
R439	C183 = '1' & C1501 = '1' & C1531 = '0' & C1996 = '1' & C2237 = '1'	5	0.01	0	E15
R440	C162 = '0' & C309 = '1' & C2432 = '1' & C3106 = '1'	4	0.011	0	E15
R441	C122 = '1' & C1501 = '1' & C2237 = '1' & C2933 = '0'	4	0.01	0	E15
R442	C326 = '0' & C1189 = '1' & C1996 = '1' & C3106 = '1'	4	0.01	0	E15
R443	C183 = '0' & C211 = '0' & C1263 = '1' & C1501 = '1' & C1531 = '0' & C1780 = '1' & C3125 = '0' & C3311 = '0'	8	0.012	0	E15
R444	C309 = '1' & C1779 = '0' & C2237 = '1' & C3106 = '1'	4	0.013	0	E15
R445	C2432 = '1' & C2568 = '0' & C2655 = '1'	3	0.012	0	E15
R446	C2071 = '0' & C2185 = '1' & C2432 = '1'	3	0.01	0	E15
R447	C653 = '1' & C1065 = '0' & C1771 = '1' & C2844 = '0' & C3117 = '1'	5	0.01	0	E15
R448	C1126 = '1' & C2237 = '1' & C2868 = '0' & C3106 = '1'	4	0.012	0	E15
R449	C316 = '0' & C2185 = '1' & C2237 = '1'	3	0.012	0	E15
R450	C1779 = '0' & C1780 = '1' & C2185 = '1' & C3106 = '1'	4	0.012	0	E15
R451	C1126 = '1' & C2432 = '1' & C2568 = '0' & C3106 = '1'	4	0.012	0	E15

Rule ID	Rules	Length	Freq.	Error	Class ID
R452	C1126 = '1' & C1996 = '1' & C3013 = '0' & C3106 = '1' & C3125 = '0'	5	0.016	0	E15
R453	C162 = '0' & C2237 = '1' & C2655 = '1' & C2659 = '0'	4	0.013	0	E15
R454	C122 = '1' & C2071 = '0' & C2237 = '1' & C3106 = '1'	4	0.013	0	E15
R455	C1126 = '1' & C2237 = '1' & C3106 = '1' & C3125 = '0'	4	0.012	0	E15
R456	C458 = '0' & C1126 = '1' & C2432 = '1' & C3106 = '1'	4	0.012	0	E15
R457	C1263 = '1' & C1387 = '0' & C2185 = '1' & C3106 = '1' & C3125 = '0'	5	0.013	0	E15
R458	C326 = '0' & C1126 = '1' & C1501 = '1' & C1996 = '1'	4	0.014	0	E15
R459	C309 = '1' & C2432 = '1' & C2703 = '0' & C3106 = '1'	4	0.011	0	E15
R460	C316 = '0' & C2185 = '1' & C2432 = '1'	3	0.011	0	E15
R461	C458 = '0' & C1189 = '1' & C2432 = '1'	3	0.01	0	E15
R462	C122 = '1' & C162 = '0' & C2237 = '1' & C3106 = '1'	4	0.013	0	E15
R463	C1377 = '0' & C2185 = '1' & C2432 = '1'	3	0.011	0	E15
R464	C183 = '1' & C1347 = '0' & C1531 = '0' & C1996 = '1' & C3106 = '1'	5	0.01	0	E15
R465	C1126 = '1' & C1501 = '1' & C2432 = '1' & C2568 = '0'	4	0.012	0	E15
R466	C166 = '1' & C183 = '1' & C1531 = '0' & C2237 = '1' & C3106 = '1'	5	0.01	0	E15
R467	C2071 = '0' & C2237 = '1' & C2356 = '1' & C3013 = '0' & C3106 = '1'	5	0.019	0	E15
R468	C1126 = '1' & C1501 = '1' & C2237 = '1' & C3473 = '0'	4	0.013	0	E15
R469	C309 = '1' & C1501 = '1' & C2237 = '1' & C3125 = '0'	4	0.011	0	E15
R470	C183 = '1' & C1507 = '0' & C1773 = '0' & C2785 = '0' & C3106 = '1' & C3192 = '1'	6	0.012	0	E15
R471	C316 = '0' & C1126 = '1' & C2432 = '1' & C3106 = '1'	4	0.012	0	E15
R472	C558 = '0' & C1672 = '0' & C2237 = '1' & C2496 = '0' & C3106 = '1'	5	0.013	0	E15
R473	C326 = '0' & C1585 = '1' & C1589 = '1' & C1996 = '1'	4	0.012	0	E15

Rule ID	Rules	Length	Freq.	Error	Class ID
R474	C183 = '0' & C458 = '0' & C2655 = '1' & C3031 = '1' & C3117 = '1'	5	0.01	0	E15
R475	C182 = '1' & C1771 = '0' & C1779 = '0' & C3106 = '1' & C3360 = '0'	5	0.01	0	E15
R476	C309 = '1' & C1501 = '1' & C2071 = '0' & C2237 = '1'	4	0.013	0	E15
R477	C122 = '1' & C1501 = '1' & C2071 = '0' & C2237 = '1'	4	0.013	0	E15
R478	C458 = '0' & C1126 = '1' & C1501 = '1' & C2432 = '1'	4	0.012	0	E15
R479	C309 = '1' & C1501 = '1' & C2432 = '1' & C2703 = '0'	4	0.011	0	E15
R480	C1377 = '0' & C2432 = '1' & C2655 = '1'	3	0.011	0	E15
R481	C309 = '1' & C1501 = '1' & C2432 = '1' & C2568 = '0'	4	0.012	0	E15
R482	C558 = '0' & C1771 = '0' & C2166 = '0' & C2237 = '1' & C2568 = '0' & C2868 = '0' & C3106 = '1'	7	0.01	0	E15
R483	C183 = '1' & C558 = '0' & C1501 = '1' & C2432 = '1'	4	0.01	0	E15
R484	C183 = '1' & C558 = '0' & C2432 = '1' & C3106 = '1'	4	0.01	0	E15
R485	C326 = '0' & C1501 = '0' & C1852 = '1' & C2185 = '0' & C3197 = '0'	5	0.008	0	E16
R486	C183 = '0' & C1820 = '0' & C3162 = '1'	3	0.008	0	E16
R487	C183 = '0' & C1455 = '0' & C3162 = '1'	3	0.008	0	E16
R488	C326 = '0' & C1462 = '0' & C2949 = '1' & C3106 = '0' & C3197 = '0'	5	0.008	0	E16
R489	C326 = '0' & C1462 = '0' & C1501 = '0' & C2949 = '1' & C3197 = '0'	5	0.008	0	E16
R490	C1531 = '0' & C2037 = '0' & C3162 = '1' & C3237 = '1'	4	0.009	0	E16
R491	C183 = '0' & C1219 = '0' & C3162 = '1'	3	0.008	0	E16
R492	C326 = '0' & C1227 = '0' & C1501 = '0' & C1852 = '1' & C3197 = '0'	5	0.008	0	E16
R493	C326 = '0' & C1280 = '0' & C2949 = '1' & C3106 = '0' & C3197 = '0'	5	0.008	0	E16
R494	C183 = '0' & C1108 = '0' & C3162 = '1'	3	0.008	0	E16
R495	C1331 = '0' & C1734 = '0' & C3162 = '1' & C3237 = '1'	4	0.008	0	E16

Rule ID	Rules	Length	Freq.	Error	Class ID
R496	C326 = '0' & C1501 = '0' & C1852 = '1' & C2655 = '0' & C3197 = '0'	5	0.008	0	E16
R497	C326 = '0' & C1227 = '0' & C2949 = '1' & C3106 = '0' & C3197 = '0'	5	0.008	0	E16
R498	C326 = '0' & C1462 = '0' & C1501 = '0' & C1852 = '1' & C3197 = '0'	5	0.008	0	E16
R499	C326 = '0' & C2185 = '0' & C2949 = '1' & C3106 = '0' & C3197 = '0'	5	0.008	0	E16
R500	C183 = '0' & C326 = '0' & C1280 = '0' & C1852 = '1' & C2037 = '0' & C3106 = '0'	6	0.008	0	E16
R501	C326 = '0' & C1280 = '0' & C1501 = '0' & C1852 = '1' & C3197 = '0'	5	0.008	0	E16
R502	C1531 = '0' & C1734 = '0' & C3162 = '1' & C3170 = '0' & C3192 = '0'	5	0.008	0	E16
R503	C326 = '0' & C1462 = '0' & C1852 = '1' & C3106 = '0' & C3197 = '0'	5	0.008	0	E16
R504	C326 = '0' & C1852 = '1' & C2185 = '0' & C3106 = '0' & C3197 = '0'	5	0.008	0	E16
R505	C326 = '0' & C1280 = '0' & C1501 = '0' & C2949 = '1' & C3197 = '0'	5	0.008	0	E16
R506	C326 = '0' & C1227 = '0' & C1501 = '0' & C2949 = '1' & C3197 = '0'	5	0.008	0	E16
R507	C326 = '0' & C1227 = '0' & C1852 = '1' & C3106 = '0' & C3197 = '0'	5	0.008	0	E16
R508	C326 = '0' & C1501 = '0' & C2185 = '0' & C2949 = '1' & C3197 = '0'	5	0.008	0	E16
R509	C183 = '0' & C1890 = '0' & C3162 = '1'	3	0.008	0	E16
R510	C326 = '0' & C1280 = '0' & C1852 = '1' & C3106 = '0' & C3197 = '0'	5	0.008	0	E16
R511	C1531 = '0' & C1734 = '0' & C2619 = '0' & C3162 = '1' & C3237 = '1'	5	0.009	0	E16



Rule ID	Rules	Length	Freq.	Error	Class ID
R512	C1531 = '0' & C1734 = '0' & C3162 = '1' & C3192 = '0' & C3353 = '0'	5	0.008	0	E16
R513	C1672 = '1' & C2451 = '1' & C3103 = '0' & C3197 = '0'	4	0.01	0	E17
R514	C45 = '1' & C1418 = '0' & C2906 = '1' & C3197 = '0'	4	0.01	0	E17
R515	C45 = '1' & C2906 = '1' & C3103 = '0' & C3197 = '0'	4	0.01	0	E17
R516	C2451 = '1' & C2906 = '1' & C3103 = '0' & C3197 = '0'	4	0.01	0	E17
R517	C45 = '1' & C183 = '0' & C555 = '0' & C2037 = '1' & C2619 = '0' & C3103 = '0' & C3106 = '0'	7	0.01	0	E17
R518	C1189 = '0' & C1418 = '0' & C2868 = '1' & C3197 = '0'	4	0.01	0	E17
R519	C45 = '1' & C2451 = '1' & C3103 = '0' & C3197 = '0'	4	0.01	0	E17
R520	C45 = '1' & C1418 = '0' & C1724 = '0' & C2037 = '1' & C3197 = '0'	5	0.01	0	E17
R521	C45 = '1' & C1418 = '0' & C2568 = '1' & C3197 = '0'	4	0.01	0	E17
R522	C45 = '1' & C1589 = '0' & C2037 = '1' & C3103 = '0' & C3197 = '0'	5	0.01	0	E17
R523	C2451 = '1' & C2568 = '1' & C3103 = '0' & C3197 = '0'	4	0.01	0	E17
R524	C45 = '1' & C1734 = '0' & C2811 = '1' & C3103 = '0' & C3237 = '1'	5	0.012	0	E17
R525	C45 = '1' & C2868 = '1' & C3103 = '0' & C3197 = '0' & C3323 = '0'	5	0.01	0	E17
R526	C411 = '1' & C1418 = '0' & C2037 = '1' & C3197 = '0'	4	0.01	0	E17
R527	C1103 = '0' & C1418 = '0' & C2868 = '1' & C3197 = '0'	4	0.01	0	E17
R528	C183 = '0' & C1227 = '1' & C1418 = '0' & C2451 = '1' & C3106 = '0'	5	0.01	0	E17
R529	C1287 = '0' & C2868 = '1' & C3103 = '0' & C3197 = '0'	4	0.01	0	E17

Rule ID	Rules	Length	Freq.	Error	Class ID
R530	C183 = '0' & C411 = '1' & C1418 = '0' & C2949 = '0' & C3106 = '0'	5	0.01	0	E17
R531	C183 = '0' & C411 = '1' & C1418 = '0' & C2496 = '0' & C3106 = '0'	5	0.01	0	E17
R532	C411 = '1' & C1418 = '0' & C1429 = '0' & C1866 = '0' & C2143 = '0'	5	0.01	0	E17
R533	C1734 = '0' & C2037 = '1' & C2385 = '0' & C2811 = '1'	4	0.013	0	E17
R534	C1418 = '0' & C2451 = '1' & C2868 = '1' & C3197 = '0'	4	0.01	0	E17
R535	C1070 = '1' & C2868 = '1' & C3103 = '0' & C3197 = '0' & C3323 = '0'	5	0.01	0	E17
R536	C1418 = '0' & C1589 = '0' & C2037 = '1' & C2844 = '1' & C3197 = '0'	5	0.01	0	E17
R537	C1734 = '0' & C2385 = '0' & C2451 = '1' & C2811 = '1'	4	0.013	0	E17
R538	C411 = '1' & C1418 = '0' & C1589 = '0' & C2811 = '1'	4	0.012	0	E17
R539	C1070 = '1' & C1418 = '0' & C2868 = '1' & C3197 = '0' & C3323 = '0'	5	0.01	0	E17
R540	C183 = '0' & C411 = '1' & C1418 = '0' & C2568 = '1' & C3106 = '0'	5	0.01	0	E17
R541	C182 = '0' & C411 = '1' & C1418 = '0' & C2496 = '0'	4	0.01	0	E17
R542	C2451 = '1' & C2844 = '1' & C3103 = '0' & C3197 = '0'	4	0.01	0	E17
R543	C45 = '1' & C1721 = '0' & C2037 = '1' & C3103 = '0' & C3197 = '0'	5	0.01	0	E17
R544	C45 = '1' & C2568 = '1' & C3103 = '0' & C3197 = '0'	4	0.01	0	E17
R545	C2568 = '1' & C2868 = '1' & C3103 = '0' & C3197 = '0'	4	0.01	0	E17
R546	C2451 = '1' & C2868 = '1' & C3103 = '0' & C3197 = '0'	4	0.01	0	E17

Rule ID	Rules	Length	Freq.	Error	Class ID
R547	C183 = '0' & C411 = '1' & C2844 = '1' & C3103 = '0' & C3106 = '0'	5	0.01	0	E17
R548	C45 = '1' & C183 = '0' & C1589 = '0' & C1672 = '1' & C3103 = '0' & C3106 = '0'	6	0.01	0	E17
R549	C1418 = '0' & C2451 = '1' & C2844 = '1' & C3197 = '0'	4	0.01	0	E17
R550	C45 = '1' & C183 = '0' & C1418 = '0' & C2568 = '1' & C3106 = '0'	5	0.01	0	E17
R551	C45 = '1' & C183 = '0' & C1418 = '0' & C1589 = '0' & C2844 = '1' & C3106 = '0'	6	0.01	0	E17
R552	C45 = '1' & C183 = '0' & C1501 = '0' & C2037 = '1' & C2619 = '0' & C3103 = '0' & C3323 = '0'	7	0.01	0	E17
R553	C1103 = '0' & C2868 = '1' & C3103 = '0' & C3197 = '0'	4	0.01	0	E17
R554	C45 = '1' & C1418 = '0' & C2451 = '1' & C3197 = '0'	4	0.01	0	E17
R555	C1734 = '0' & C2037 = '1' & C2143 = '0' & C2906 = '1' & C3103 = '0' & C3205 = '0'	6	0.01	0	E17
R556	C13 = '1' & C2868 = '1' & C3103 = '0' & C3197 = '0'	4	0.01	0	E17
R557	C45 = '1' & C1418 = '0' & C1531 = '0' & C1672 = '1' & C1773 = '0' & C3117 = '0'	6	0.01	0	E17
R558	C45 = '1' & C1429 = '0' & C1589 = '0' & C2037 = '1' & C2143 = '0' & C3103 = '0'	6	0.01	0	E17
R559	C183 = '0' & C411 = '1' & C2037 = '1' & C3103 = '0' & C3106 = '0'	5	0.01	0	E17
R560	C183 = '0' & C411 = '1' & C1418 = '0' & C2037 = '1' & C3106 = '0'	5	0.01	0	E17
R561	C411 = '1' & C1734 = '0' & C2385 = '0' & C2811 = '1'	4	0.013	0	E17
R562	C183 = '0' & C1418 = '0' & C2451 = '1' & C2568 = '1' & C3106 = '0'	5	0.01	0	E17
R563	C2868 = '1' & C2906 = '1' & C3103 = '0' & C3197 = '0'	4	0.01	0	E17
R564	C411 = '1' & C1418 = '0' & C2451 = '1' & C3197 = '0'	4	0.01	0	E17

Rule ID	Rules	Length	Freq.	Error	Class ID
R565	C13 = '1' & C183 = '0' & C1227 = '1' & C1331 = '0' & C1501 = '0' & C3103 = '0' & C3237 = '1'	7	0.01	0	E17
R566	C183 = '0' & C411 = '1' & C1866 = '0' & C3103 = '0' & C3106 = '0'	5	0.01	0	E17
R567	C1227 = '1' & C1418 = '0' & C2451 = '1' & C3197 = '0'	4	0.01	0	E17
R568	C45 = '1' & C183 = '0' & C1418 = '0' & C1501 = '0' & C1589 = '0' & C2844 = '1'	6	0.01	0	E17
R569	C183 = '0' & C411 = '1' & C1501 = '0' & C1852 = '0' & C3103 = '0'	5	0.01	0	E17
R570	C1589 = '0' & C2037 = '1' & C2868 = '1' & C3103 = '0' & C3197 = '0'	5	0.01	0	E17
R571	C183 = '0' & C1531 = '0' & C1609 = '1' & C2185 = '0'	4	0.002	0	E18
R572	C183 = '0' & C1609 = '1' & C1672 = '0'	3	0.002	0	E18
R573	C109 = '0' & C183 = '0' & C1263 = '0' & C2655 = '0' & C2933 = '1' & C3197 = '0'	6	0.002	0	E18
R574	C183 = '0' & C1531 = '0' & C1609 = '1' & C2785 = '0'	4	0.003	0	E18
R575	C1609 = '1' & C2785 = '0' & C3197 = '0'	3	0.002	0	E18
R576	C183 = '0' & C1280 = '0' & C1531 = '0' & C1609 = '1'	4	0.003	0	E18
R577	C1280 = '0' & C1589 = '0' & C2629 = '0' & C3117 = '1' & C3192 = '1' & C3197 = '0'	6	0.002	0	E18
R578	C183 = '0' & C1780 = '0' & C2143 = '1' & C3192 = '1'	4	0.002	0	E18
R579	C1609 = '1' & C2568 = '0' & C3197 = '0'	3	0.002	0	E18
R580	C1609 = '1' & C1672 = '0' & C3197 = '0'	3	0.002	0	E18
R581	C183 = '0' & C1609 = '1' & C1780 = '0'	3	0.002	0	E18
R582	C1280 = '0' & C1609 = '1' & C3197 = '0'	3	0.002	0	E18
R583	C183 = '0' & C1531 = '0' & C1609 = '1' & C2496 = '0'	4	0.003	0	E18
R584	C1609 = '1' & C2496 = '0' & C3197 = '0'	3	0.002	0	E18
R585	C183 = '0' & C1531 = '0' & C1609 = '1' & C2655 = '0'	4	0.002	0	E18
R586	C183 = '0' & C1609 = '1' & C2412 = '0' & C2496 = '0'	4	0.003	0	E18
R587	C183 = '0' & C1531 = '0' & C1609 = '1' & C2568 = '0'	4	0.003	0	E18
R588	C183 = '0' & C1531 = '0' & C1609 = '1' & C2166 = '0'	4	0.002	0	E18

<b>Rule ID</b>	<b>Rules</b>	<b>Length</b>	<b>Freq.</b>	<b>Error</b>	<b>Class ID</b>
R589	C1609 = '1' & C2166 = '0' & C3197 = '0'	3	0.002	0	E18
R590	C183 = '0' & C558 = '0' & C1501 = '0' & C2356 = '1' & C3154 = '0' & C3192 = '1'	6	0.002	0	E18
R591	C309 = '0' & C1531 = '0' & C1609 = '1' & C2496 = '0'	4	0.003	0	E18
R592	C183 = '0' & C1418 = '0' & C1609 = '1' & C2166 = '0'	4	0.002	0	E18
R593	C1771 = '1' & C1773 = '0' & C1780 = '0' & C2933 = '1'	4	0.002	0	E18
R594	C1531 = '0' & C1609 = '1' & C2496 = '0' & C3159 = '0'	4	0.003	0	E18
R595	C1280 = '0' & C1531 = '0' & C1609 = '1' & C3159 = '0'	4	0.003	0	E18
R596	C183 = '0' & C1771 = '1' & C2412 = '0' & C2496 = '0' & C3360 = '1'	5	0.003	0	E18



## Appendix G

### Source code

List of source codes in the Jamu studies. The source codes below were used in Chapter 3. Text with blue color indicates a comment.

```
R package      : bmeasures.R
#' ----- The OTUs table -----
#' Description  : Generates the OTUs table from 2 binary vectors
#' Input       :
#'   @param x as the first binary vector (represented as matrix and row vector)
#'   @param y as the second binary vector (represented as matrix and row vector)
#' Output      :
#'   @return otu_out, i.e. a,b,c,d in the OTUs table are defined as follows:
#'   a is the number of features where the value of both x and y are 1 (positive
#'   matches), b and c are the number of features where the value of x is 0 and
#'   y is 1 and vice versa, respectively (absence mismatches) and d is the number
#'   of features where the values of both x and y are 0 (negative matches).
#' -----

bmeasures_otu <- function (x,y)
{
  # compare input dimension
  dx <- dim(x)
  dy <- dim(y)

  if (dx[1] != dy[1])
  {
    cat("Inputs have different number of rows \n")
  } else if (dx[2] != dy[2])
  {
    cat("Inputs have different number of columns \n")
  } else
  {
    cat("")
  }

  # binding input as two row vectors
  input <- as.matrix(rbind(x,y))
}
```

```

a <- 0
b <- 0
c <- 0
d <- 0

# calculate a, b, c and d
for (i in 1:dx[2])
{
  k <- input[1,i]
  l <- input[2,i]

  if ((k==1) & (l==1))
  {
    a <- a + 1
  } else if ((k==0) & (l==1))
  {
    b <- b + 1
  } else if ((k==1) & (l==0))
  {
    c <- c + 1
  } else # ((k==0) & (l==0))
  {
    d <- d + 1
  }
}

otu_out <- c(a=a,b=b,c=c,d=d)
remove (a,b,c,d)
return (otu_out)
}
# end of bmeasures_otu

```



```

#' ----- The Binary Similarity/Dissimilarity Coefficient -----
#' Description : Calculate the binary similarity/dissimilarity coefficient between
#'              2 binary vectors
#' Input      :
#'            @param x as the first binary vector (represented as matrix and row vector)
#'            @param y as the second binary vector (represented as matrix and row vector)
#'            @param method as equation IDs, e.g. "eq_xx" (xx is equation ID. Please refer
#'              to Table 3.1)
#' Output     :
#'            @return results, consist of a,b,c,d and binary similarity/dissimilarity
#'              coefficient
#' -----
bmeasures <- function (x,y,method)
{
  # calculate the OTU parameters and assign the results into a,b,c,d
  otuTable <- bmeasures_otu(x,y)
  a <- otuTable[1]
  b <- otuTable[2]
  c <- otuTable[3]
  d <- otuTable[4]
  n <- a+b+c+d;

  # calculate binary similarity or dissimilarity coefficient based on selected equation
  switch (method,
    eq_01 = { coef <- a/(a+b+c) },
    eq_02 = { coef <- a/(2*a+b+c) },
    eq_03 = { coef <- (2*a)/(2*a+b+c) },
    eq_04 = { coef <- (3*a)/(3*a+b+c) },
    eq_05 = { coef <- (2*a)/((a+b)+(a+c)) },
    eq_06 = { coef <- a/(a+(2*b)+(2*c)) },
    eq_07 = { coef <- (a+d)/n },
    eq_08 = { coef <- (2*(a+d))/((2*a)+b+c+(2*d)) },
    eq_09 = { coef <- (a+d)/(a+(2*(b+c))+d) },
    eq_10 = { coef <- (a+(0.5*d))/n },
    eq_11 = { coef <- (a+d)/(a+(0.5*(b+c))+d) },
    eq_12 = { coef <- a },
    eq_13 = { coef <- a+d },
    eq_14 = { coef <- a/n },
    eq_15 = { coef <- b+c },
    eq_16 = { coef <- sqrt(b+c) },
    eq_17 = { coef <- sqrt((b+c)^2) },
    eq_18 = { coef <- sqrt((b+c)^2) },
    eq_19 = { coef <- b+c },
    eq_20 = { coef <- (b+c)/n },
    eq_21 = { coef <- b+c },
  )
}

```

```

eq_22 = { coef <- b+c },
eq_23 = { coef <- (b+c)/(4*n) },
eq_24 = { coef <- ((b+c)^2)/(n^2) },
eq_25 = { coef <- ((n*(b+c))-((b-c)^2))/(n^2) },
eq_26 = { coef <- (4*b*c)/(n^2) },
eq_27 = { coef <- (b+c)/(2*a+b+c) },
eq_28 = { coef <- (b+c)/(2*a+b+c) },
eq_29 = { coef <- 2 * (sqrt(1-(a/sqrt((a+b)*(a+c)))) ) },
eq_30 = { coef <- (sqrt(2*(1-(a/(sqrt((a+b)*(a+c)))))) ) },
eq_31 = { coef <- a/(sqrt((a+b)*(a+c))) },
eq_32 = { coef <- log(a) - log(n) - log((a+b)/n) - log((a+c)/n) },
eq_33 = { coef <- a/(sqrt((a+b)*(a+c))) },
eq_34 = { coef <- (n*a)/((a+b)*(a+c)) },
eq_35 = { coef <- (n*((a-0.5)^2))/((a+b)*(a+c)) },
eq_36 = { coef <- (a^2)/((a+b)*(a+c)) },
eq_37 = { coef <- a/(0.5*(a*b + a*c)+(b*c)) },
eq_38 = { coef <- a/(((a+b)*(a+c))^0.5) },
eq_39 = { coef <- ((a^2)-(b*c))/((a+b)*(a+c)) },
eq_40 = { coef <- ((n*a) - (a+b)*(a+c))/((n*a) + (a+b)*(a+c)) },
eq_41 = { coef <- ((a/2)*(2*a+b+c))/((a+b)*(a+c)) },
eq_42 = { coef <- (a/2)*((1/(a+b))+1/(a+c)) },
eq_43 = { coef <- (a/(a+b))+a/(a+c) },
eq_44 = { coef <- ((a*d)-(b*c))/(sqrt(n*(a+b)*(a+c))) },
eq_45 = { coef <- a/(min((a+b),(a+c))) },
eq_46 = { coef <- a/(max((a+b),(a+c))) },
eq_47 = { coef <- (a/sqrt((a+b)*(a+c)))-(max((a+b),(a+c))/2) },
eq_48 = { coef <- ((n*a)-((a+b)*(a+c)))/((n*min(a+b,a+c))-((a+b)*(a+c))) },
eq_49 = { coef <- 0.25 * ((a/(a+b))+a/(a+c)+(d/(b+d))+d/(c+d)) },
eq_50 = { coef <- (a+d)/(sqrt((a+b)*(a+c)*(b+d)*(c+d))) },
eq_51 = { x2 <- (n*(((a*d)-(b*c))^2))/((a+b)*(a+c)*(c+d)*(b+d))
      coef <- x2 },
eq_52 = { x2 <- (n*(((a*d)-(b*c))^2))/((a+b)*(a+c)*(c+d)*(b+d))
      coef <- sqrt(x2/(n+x2)) },
eq_53 = { p <- ((a*d) - (b*c))/(sqrt((a+b)*(a+c)*(b+d)*(c+d)))
      coef <- sqrt(p/(n+p)) },
eq_54 = { p <- ((a*d) - (b*c))/(sqrt((a+b)*(a+c)*(b+d)*(c+d)))
      coef <- p },
eq_55 = { coef <- cos((pi*sqrt(b*c))/(sqrt(a*d)+sqrt(b*c))) },
eq_56 = { coef <- (a+d)/(b+c) },
eq_57 = { coef <- (a*d)/(((a+b)*(a+c)*(b+d)*(c+d))^0.5) },
eq_58 = { coef <- (sqrt(2) * (a*d - b*c)) / (sqrt(((a*d - b*c)^2) -
      (a+b)*(a+c)*(b+d)*(c+d))) },
eq_59 = { coef <- log10( (n*(((abs(a*d - b*c)) - n/2)^2)) / ((a+b)*(a+c)*(b+d)*(c+d)))
      },
eq_60 = { coef <- (a*d)/(sqrt((a+b)*(a+c)*(b+d)*(c+d))) },

```

```

eq_61 = { coef <- (a*d - b*c)/(a*d + b*c) },
eq_62 = { coef <- ((2*b*c)/(a*d + b*c)) },
eq_63 = { coef <- (sqrt(a*d) - sqrt(b*c))/(sqrt(a*d) + sqrt(b*c)) },
eq_64 = { coef <- a/(b+c) },
eq_65 = { coef <- a/((a+b)+(a+c)-a) },
eq_66 = { coef <- (a*d - b*c)/(n^2) },
eq_67 = { coef <- ((a+d)-(b+c))/n },
eq_68 = { coef <- (4*(a*d - b*c))/(((a+d)^2)+((b+c)^2)) },
eq_69 = { sig <- max(a,b) + max(c,d) + max(a,c) + max(b,d)
          sigt <- max(a+c,b+d) + max(a+b,c+d)
          coef <- (sig-sigt)/(2*n - sigt) },
eq_70 = { sig <- max(a,b) + max(c,d) + max(a,c) + max(b,d)
          sigt <- max(a+c,b+d) + max(a+b,c+d)
          coef <- (sig - sigt)/(2*n) },
eq_71 = { coef <- (sqrt(a*d)+a)/(sqrt(a*d)+a+b+c) },
eq_72 = { coef <- (sqrt(a*d)+a-(b+c))/(sqrt(a*d)+a+b+c) },
eq_73 = { coef <- (a*b + b*c)/((a*b)+(2*b*c)+(c*d)) },
eq_74 = { coef <- ((n^2) * (n*a - (a+b)*(a+c))) / ((a+b)*(a+c)*(b+d)*(c+d)) },
eq_75 = { coef <- (a*(c+d))/(c*(a+b)) },
eq_76 = { coef <- abs((a*(c+d))/(c*(a+b))) },
eq_77 = { coef <- log(1+a)/log(1+n) },
eq_78 = { coef <- log(1+a)/log(1+a+b+c) },
eq_79 = { coef <- (log(1+a*d)-log(1+b*c))/log(1+(n^2)/4) },
{ cat("No desired equation. Please check it again.") }
)

result <- c(a,b,c,d,coef)
result <- t(as.matrix(result))
colnames(result) <- c("a", "b", "c", "d", "coef")
return(result)
}
# end bmeasures

```

```

# ----- Binding binary similarity/dissimilarity coefficients -----
# Description : combine the similarity and dissimilarity coefficients obtained from
#              79 binary equations as a matrix of Jamu pairs as rows and equations
#              as columns
# Input       : 79 text files (.txt) as an output of similarity or dissimilarity
#              measures of Jamu pairs using an equation. Each file consists of
#              7 columns, i.e. JamuID1, JamuID2, similarity/dissimilarity
#              coefficient, a,b,c,d
# Output      : result or "jamu_relations.RData" (Jamu pair-similarity matrix).
#              It consists of 81 columns data, which are JamuID1, JamuID2, and
#              similarity/dissimilarity coefficients.
# -----

# setting working directory
setwd("~/jamu/jamu_relations")

# read the first similarity/dissimilarity coefficient obtained by Eq.1
data_relation1 <- read.table("relation1.txt")
result <- data_relation1[,1:3]
remove(data_relation1)

# bind the similarity/dissimilarity coefficient with other similarity/dissimilarity
# coefficients (Eq.2-Eq.79)
for(i in 2:79)
{
  filename <- paste("relation",i,".txt")
  filename <- gsub(" ","",filename)
  cat(filename,"\n")

  new_relation <- read.table(filename)
  rel_n <- new_relation[,3]
  result <- cbind(result,rel_n)
  remove(new_relation)
  gc()
}

# set colnames
colname <- sprintf("eq_%02d",1:79)
colname <- c("id1", "id2", colname)
colnames(result) <- colname

# save the result as an RData
save(result, file="jamu_relations.RData")
unlink(".RData")

```

```

# ----- Compare efficacy between Jamu formulas -----
# Description   : compare efficacy in a Jamu pair. Efficacy is match if both Jamu
#               : formulas have the same efficacy. Otherwise, efficacy is mismatch.
# Input        : Jamu efficacy groups (disease classes)
# Output       : match or "match_jamu4e1.RData". It consists of 6 columns, i.e.
#               : #Jamu1, #Jamu2, efficacy_#Jamu1, efficacy_#Jamu2, 1/0
#               : (match/mismatch) and similarity/dissimilarity coefficients.
# -----

# load input data and transform as matrix
class <- read.csv("jamu_class.csv")
class <- as.matrix(class)

dclass <- dim(class)
rclass <- dclass[1]
cclass <- dclass[2]

# temporary matrix for output
match <- matrix(0,4921953,6)
x <- 1

# print Jamu pairs index and compare efficacy between Jamu formulas in a pair
for (i in 1:(rclass-1))
{
  for (j in (i+1):rclass)
  {
    match[x,1] <- i
    match[x,2] <- j

    classx <- class[i,2]
    classy <- class[j,2]
    match[x,3] <- classx
    match[x,4] <- classy

    # assign 1 if both Jamu formulas have the same efficacy
    if (classx == classy)
    {
      match[x,5] <- 1
      match[x,6] <- classx
    }
    x <- x+1
  }
}
save(match, file="match_jamu4e1.RData")
unlink(".RData")

```

```

# ----- Hierarchical clustering of equations -----
# Description   : Hierarchical clustering with centroid linkage between equations
                  based on coefficients of Jamu pairs
# Input         : jamu_relations.RData (relationship between Jamu pairs and binary
#                  similarity and dissimilarity coefficients)
# Output        : dendrogram and heatmap between equations
# -----

# load the library
library(gplots)
library(gclus) # for order.hclust

# set filename of output
ofname<-"sony3c_clust.pdf"

# setting the color matrix for heatmap
colmat <- matrix(0,ncol=8,nrow=4)
colmat[1,] <- c( 0, 9, 9, 9, 0, 0, 0, 1)
colmat[2,] <- c(-1, 0, 0, 1, 0, 1, 1, 1)
colmat[3,] <- c( 0, 1, 1, 1, 1, 1, 0, 0)
colmat[4,] <- c( 0, 1, 0, 0, 0, 9, 9, 9)

# set working directory and load the input data
setwd("~/jamu")
load("jamu_relations.RData")
mat <- as.matrix(result)
remove(result)

# load disease class and match/mismatch efficacy
load("match_jamu4e1.RData")
tmpmatch <- as.matrix(match[,3:4])
colnames(tmpmatch) <- c("eff1", "eff2")
mat <- cbind(mat,tmpmatch)
remove(match, tmpmatch)

# exclude some disease classes
# class 0
mat <- subset(mat, mat[,82] != 0)
mat <- subset(mat, mat[,83] != 0)
# class 4
mat <- subset(mat, mat[,82] != 4)
mat <- subset(mat, mat[,83] != 4)
# class 5
mat <- subset(mat, mat[,82] != 5)
mat <- subset(mat, mat[,83] != 5)

```

```

# remove JamuID1, JamuID2, efficacy_JamuID1 and efficacy_JamuID2
mat <- mat[,c(-1,-2,-82,-83)] # character
class(mat) <- "numeric" # convert char to numeric

colname <- sprintf("eq_%02d",1:79)
colnames(mat) <- colname

# remove 'duplicate' equations
dupli_eq <- sprintf("eq_%02d",c(5,11,17,18,19,21,22,28,33,38,60,65,69,70))
mat <- mat[,!(colnames(mat) %in% dupli_eq)]

# remove equations with infinite and NA values
mat[is.infinite(mat)] <- NA
mat <- mat[,!is.na(colSums(mat))]
mat <- mat[!is.na(rowSums(mat)),]

# centers and scales the columns of a numeric matrix
mat<- scale(mat, apply(mat, 2, mean), apply(mat, 2, sd))

pdf(height=34/2.5,width=34/2.5,file=ofname)

# hierarchical clustering with centroid linkage. Euclidean distance used to measure
# the distance between equations
hc <- hclust(dist(t(mat),method="euclidean"),method="centroid")
plot(hc,hang=-1, xlab="Equations", sub=" ", ylab="Distances", main=NULL)

# draw the heatmap between equations
cor1 <- cor(mat)
mycol <- myColor(cor1,colmat)

dend1 <- as.dendrogram(hc)
heatmap.2(cor1,
          Colv=dend1,Rowv=rev(dend1),
          col=mycol,
          scale='none',
          trace='none',keysize=1.0,density.info="none",key=TRUE,dendrogram="both",
          margin=c(7,12),cexCol=1.0,cexRow=1.0)

# list of eliminated equations (according to the clustering results)
lstrm <- sprintf("eq_%02d",c(3,13,14,20,23,30,41,42,43,67,72))
matlim <- mat[!(colnames(mat) %in% lstrm),!(colnames(mat) %in% lstrm)]

```

```

# assign name for selected equations
label <- c("Sjaccard", "Sdice", "S3w-jaccard", "Ssokal&sneath-1", "Ssokal&michener",
          "Ssokal&sneath-2", "Sroger&tanimoto", "Sfaith", "Sintersection",
          "Dhamming", "Deuclid", "Dsize-difference", "Dshapedifference",
          "Dpattern-difference", "Dlance&williams", "Dhellinger", "Scosine",
          "Sochiai-1", "Sfossun", "Ssorgenfrei", "Smcconnaughey", "Starwid",
          "Skulczynski-2", "Sdennis", "Ssimpson", "Sbraun&banquet", "Sfager&mcgowan",
          "Sforbes-2", "Ssokal&sneath-4", "Sgower", "Spearson-1", "Spearson-2",
          "Spearson-heron-1", "Spearson-heron-2", "Ssokal&sneath-5", "Sstiles",
          "Syuleq", "Dyuleq", "Syulew", "Sdispersion", "Smichael", "Sgoodman&kruskal",
          "Sanderberg", "Sbaroni-urbani&buser-1", "Seyraud")

colnames(matlim) <- label

# hierarchical clustering with centroid linkage for selected equations
hclim <- hclust(dist(t(matlim),method="euclidean"),method="centroid")
plot(hclim,hang=-1, xlab="Equations", sub=" ", ylab="Distances", main=NULL)

# draw the heatmap between equations
corlim <- cor1[!(colnames(mat) %in% lstrm),!(colnames(mat) %in% lstrm)]
mycollim <- myColor(corlim,colmat)

dendlim <- as.dendrogram(hclim)
heatmap.2(corlim,
          Colv=dendlim,Rowv=rev(dendlim),
          col=mycollim,
          scale='none',
          trace='none',keysize=1.0,density.info="none",key=TRUE,dendrogram="both",
          margin=c(12,12),cexCol=1.0,cexRow=1.0)

dev.off()
remove(mat)
gc()

```



```

# ----- Finding a suitable binary similarity measures -----
# Description      : evaluation of selected equations using ROC analysis (AUC)
#                  (case study on Jamu data)
# Input           : "jamu_relations.RData" (relationship between Jamu pairs and binary
#                  similarity and dissimilarity coefficients) and
#                  "match_jamu4e1.RData" (match/mismatch efficacy in Jamu pairs)
# Output          : minimum distance between optimum point to ROC curve and AUC score
#                  eq_auc ("jamuSim4g_update5a_aucAll.csv") is 20 AUCs for each equation
#                  idsample ("jamuSim4g_update5a_idsample.csv") is random index from mismatch
#                  roc_out ("jamuSim4g_update5a_rocAll.RData") is list of roc
# -----

# load similarity/dissimilarity coefficients of Jamu
setwd("~/jamu")
load("jamu_relations.RData")
mat <- as.matrix(result)
remove(result)

# load disease class and match/mismatch data
load("match_jamu4e1.RData")
tmpmatch <- as.matrix(match[,3:5])
colnames(tmpmatch) <- c("eff1", "eff2", "match")
mat <- cbind(mat, tmpmatch)
remove(match, tmpmatch)

# exclude some disease classes
# class 0
mat <- subset(mat, mat[,82] != 0)
mat <- subset(mat, mat[,83] != 0)
# class 4
mat <- subset(mat, mat[,82] != 4)
mat <- subset(mat, mat[,83] != 4)
# class 5
mat <- subset(mat, mat[,82] != 5)
mat <- subset(mat, mat[,83] != 5)

# remove efficacies from each Jamu pair
mat <- mat[,c(-82,-83)]      # character
class(mat) <- "numeric"    # convert char to numeric

# column label
eqname <- sprintf("eq_%02d",1:79)
colname <- c("i", "j", eqname, "match")
colnames(mat) <- colname

```

```

# remove equations that produced infinite and NA values in Jamu data
remcol <- sprintf("eq_%02d",c(32,37,56,64,69,70,73,75,76))
mat <- mat[,!(colnames(mat) %in% remcol)]
mat[is.infinite(mat)] <- NA
mat <- mat[,!is.na(colSums(mat))]

# eliminate identical equations and equations that belong to the same cluster with
# another equation
lstrm <- sprintf("eq_%02d",c(3,5,11,13,14,17,18,19,20,21,22,23,28,30,33,38,41,
42,43,60,65,67,72))
mat <- mat[,!(colnames(mat) %in% lstrm)]

label <- colnames(mat)
z <- length(label)

# subset of Jamu pairs with match and mismatch classes
oneJ <- subset(mat, mat[,z] == 1)
zeroJ <- subset(mat, mat[,z] == 0)

# define number of threshold for ROC analysis
numrange <- 100

# list of equations classified as dissimilarity measures
dissim <- c("eq_15", "eq_16", "eq_17", "eq_18", "eq_19", "eq_20", "eq_21", "eq_22",
"eq_23", "eq_24", "eq_25", "eq_26", "eq_27", "eq_28", "eq_29", "eq_30",
"eq_62")

# list of equations that include parameter d of OTU table
includeD <- c("eq_07", "eq_08", "eq_09", "eq_10", "eq_24", "eq_25", "eq_26", "eq_35",
"eq_40", "eq_44", "eq_48", "eq_49", "eq_50", "eq_51", "eq_52", "eq_54",
"eq_55", "eq_57", "eq_59", "eq_61", "eq_62", "eq_63", "eq_66", "eq_68",
"eq_71", "eq_74", "eq_77", "eq_79")

# setting working directory for ROC analysis
setwd("~/jamu/roc")
fileout <- "jamuSim4g_update5a_roc.pdf"
pdf(height=34/2.5,width=34/2.5,file=fileout)
roc_out <- list()

library(ROCR)
doneJ <- dim(oneJ)
dzeroJ <- dim(zeroJ)
data_match <- matrix(1,doneJ[1],2) # cols: sim/dissim coeff values, label(1/0)
data_mismatch <- matrix(0,doneJ[1],2)
eq_auc <- matrix(0,(z-3),22) # cols: EqID, type(S/D), AUC from 20 samples

```

```

# ROC analysis for all selected equations (45 equations)
for (eq in 3:(z-1))
{
  cat("Processing:", label[eq], "\n")

  # determine min and max similarity/dissimilarity values
  maxSim1J <- max(oneJ[,eq])
  minSim1J <- min(oneJ[,eq])
  maxSim0J <- max(zeroJ[,eq])
  minSim0J <- min(zeroJ[,eq])
  maxSimJ <- max(maxSim1J, maxSim0J)
  minSimJ <- min(minSim1J, minSim0J)

  # normalize the data by using (X-min)/(max-min)
  oneJ[,eq] <- as.numeric(as.matrix((oneJ[,eq]-minSimJ)/(maxSimJ-minSimJ)))
  zeroJ[,eq] <- as.numeric(as.matrix((zeroJ[,eq]-minSimJ)/(maxSimJ-minSimJ)))

  # transform dissimilarity into similarity values (S = 1 - D^2), set sim/dissim
  mysim <- label[eq] %in% dissim
  if (mysim == "TRUE")
  {
    type <- "D"
    msg0 <- paste("Type: Dissimilarity measure")

    # S = 1 - D^2
    oneJ[,eq] <- 1 - (oneJ[,eq]^2)
    zeroJ[,eq] <- 1 - (zeroJ[,eq]^2)
  } else
  {
    type <- "S"
    msg0 <- paste("Type: Similarity measure")
  }

  # in term of Jamu data, number of match < mismatch, then random sampling mismatch
  # class (create 20 samples, each equal to the size of match class)
  data_match[,1] <- oneJ[,eq]

  for (zsample in 1:20)
  {
    cat(" Sample:", zsample, "\n")
    # select sample from zeroJtemp randomly (having the same dimension with match class)
    # generate random index
    zero_sample <- sample(1:dzeroJ[1], doneJ[1], replace=FALSE)
    zeroJtemp <- zeroJ[zero_sample,eq]
    data_mismatch[,1] <- zeroJtemp
  }
}

```

```

# binding match and mismatch data for ROC analysis using ROCR package
rocr <- rbind(data_match, data_mismatch)
colnames(rocr) <- c("simCoefVal", "labels")
pred <- prediction(rocr[,1], rocr[,2])
perf <- performance(pred,"tpr","fpr")

# AUC analysis
auc <- performance(pred,"auc")
auc_label <- unlist ( auc@y.name)
auc_values <- round(as.numeric( unlist ( auc@y.values) ), digits=4)

if(zsample > 1)
{
  eq_auc[(eq-2),(zsample+2)] <- auc_values
  idsample <- cbind(idsample, zero_sample)
} else
{
  eq_auc[(eq-2),1] <- label[eq]
  eq_auc[(eq-2),2] <- type
  eq_auc[(eq-2),3] <- auc_values

  # save idsample
  idsample <- zero_sample
}
remove(rocr, zeroJtemp, zero_sample)
}

# draw the ROC curves
mytitle <- paste(label[eq], " (", type, ")")
plot(perf, main=mytitle, avg='none', spread.estimate='stddev', colorize=F,
      col="blue")
abline(0,1, lty=2)
mylabel1 <- paste(auc_label, ":", auc_values)
legend(0.5,0.2,c(mylabel1), box.lwd = 0,box.col = "white",bg = "white", cex=1.3,
      pt.cex = 1)
# end of ROC analysis using ROCR package

# define max & min data, and divide it into numrange breaks
maxSim1J <- max(oneJ[,eq])
maxSim0J <- max(zeroJ[,eq])
maxSimJ <- max(maxSim1J, maxSim0J)
minSim1J <- min(oneJ[,eq])
minSim0J <- min(zeroJ[,eq])
minSimJ <- min(minSim1J, minSim0J)
simJrange <- seq(minSimJ,maxSimJ,by=((maxSimJ-minSimJ)/numrange))

```

```

# set the optimal (FPR,TPR) point and define equations type (sim/dissim)
maxxy <- c(0,1)
mysim <- label[eq] %in% dissim
if (mysim == "TRUE")
{
  type <- "D"
  msg0 <- paste("Type: Dissimilarity measure")
}
else
{
  type <- "S"
  msg0 <- paste("Type: Similarity measure")
}
mytitle <- paste(label[eq], "(", type, ") \n Match")

# determine whether an equation include negative match quantity d or not
inclD <- label[eq] %in% included
if (inclD == "TRUE")
{
  infoD <- "include d"
}
else
{
  infoD <- "exclude d"
}

# calculate the distribution of match & mismatch data and determine min and max freq
fmax1J <- hist(oneJ[,eq], plot=FALSE)
fmax0J <- hist(zeroJ[,eq], plot=FALSE)
fmaxJ <- max(fmax1J$counts, fmax0J$counts)

# histogram of match Jamu pairs
hist1J <- hist(oneJ[,eq], main=mytitle, xlab="", ylab="", col="BLUE", cex.main=1.5,
              breaks=simJrange, xlim=c(minSimJ,maxSimJ), ylim=c(0,fmaxJ),
              plot=FALSE)
# histogram of mismatch Jamu pairs
hist0J <- hist(zeroJ[,eq], main="Mismatch", xlab="", ylab="", col="RED",
              cex.main=1.5, breaks=simJrange, xlim=c(minSimJ, maxSimJ),
              ylim=c(0,fmaxJ), plot=FALSE)
# count the difference between match and mismatch data
histJ <- hist1J$counts - hist0J$counts
ymax <- max(abs(histJ))
barplot(histJ, main="Match - Mismatch", names.arg=histJ, las=2,
        ylim=c(-ymax,ymax), plot=FALSE)

```

```

# calculate FPR and TPR for each threshold, then measure the distance between (0,1)
# point to (FPR,TPR) point --> determine the min distance
# cols of roc: 1-range, 2-fdr, 3-tpr, 4-distance, 5-tp, 6-fn, 7-fp, 8-tn
roc <- matrix(0,(numrange+1), 8)
roc[,1] <- as.matrix(hist1J$breaks)
for (range in 1:(numrange+1))
{
  # FPR
  if (range <= numrange)
  {
    fp <- sum(hist0J$counts[range:numrange])
    if (range == 1)
    {
      tn <- 0
    }
    else
    {
      tn <- sum(hist0J$counts[1:(range-1)])
    }
  }
  else
  {
    fp <- 0
    tn <- sum(hist0J$counts[1:numrange])
  }

  fpr <- fp/(fp+tn)
  roc[range,2] <- fpr
  roc[range,7] <- fp
  roc[range,8] <- tn

  # TPR
  if (range <= numrange)
  {
    tp <- sum(hist1J$counts[range:numrange])
    if (range == 1)
    {
      fn <- 0
    }
    else
    {
      fn <- sum(hist1J$counts[1:(range-1)])
    }
  }
}

```

```

else
{
  tp <- 0
  fn <- sum(hist1J$counts[1:numrange])
}
tpr <- tp/(tp+fn)
roc[range,3] <- tpr
roc[range,5] <- tp
roc[range,6] <- fn

# calculate the distance of (FPR,TPR) points to maxxy
maxxy <- t(as.matrix(maxxy))
roc_dist <- sqrt(((maxxy[1]-fpr)^2) + ((maxxy[2])-tpr)^2)
roc[range,4] <- roc_dist
}

# calculate average distance
roc_distavg <- sum(as.matrix(roc[,4]))/(numrange+1)

# get index of (FPR,TPR) point with the minimum distance
# get the distance, TP, FN, FP, TN
roc_idmin <- which.min(roc[,4])
roc_distmin <- roc[roc_idmin[1],4]
mintp <- roc[roc_idmin,5]
minfn <- roc[roc_idmin,6]
minfp <- roc[roc_idmin,7]
mintn <- roc[roc_idmin,8]

# plot the (FPR,TPR) points. It will also generate an ROC curve
# plot the optimum point and draw the diagonal line from (0,0) to (1,1)
plot(roc[1:(numrange+1),2:3], main=paste(label[eq], " ( ", type, " / ", infoD, " )"),
      xlab="FPR", ylab="TPR", xlim=c(0,1), ylim=c(0,1))
points(0,1,pch=4)
abline(0,1)

# show the summary (legend)
msg1 <- paste("Min distance:", round(roc_distmin, digits=8))
msg2 <- paste("Avg distance:", round(roc_distavg, digits=4))
msgtp <- paste("TP:", mintp)
msgfn <- paste("FN:", minfn)
msgfp <- paste("FP:", minfp)
msgtn <- paste("TN:", mintn)
legend(0,0.8,c(msg0,msg1,msgtp,msgfn,msgfp,msgtn,msg2), box.lwd = 0,box.col =
      "white",bg = "white", cex=1.3, pt.cex = 1)

```

```
# save the results (range, fpr, tpr, distance, tp, fn, fp, tn)
# from each equation as list
roc_out[[eq-2]] <- roc
}

dev.off()

# save the results
write.csv(eq_auc, "jamuSim4g_update5a_aucAll.csv", row.names=FALSE)
write.csv (idsample, file="jamuSim4g_update5a_idsample.csv", row.names=FALSE)
save(roc_out, file="jamuSim4g_update5a_rocAll.RData")
unlink(".RData")
```