KERNEL P SYSTEM-MULTI OBJECTIVE BINARY PARTICLE SWARM OPTIMIZATION FEATURE SELECTION METHOD IN MICROARRAY CANCER DATASET

NAEIMEH ELKHANI

UNIVERSITI KEBANGSAAN MALAYSIA

KERNEL P SYSTEM-MULTI OBJECTIVE BINARY PARTICLE SWARM OPTIMIZATION FEATURE SELECTION METHOD IN MICROARRAY CANCER DATASET

NAEIMEH ELKHANI

THESIS SUBMITTED IN FULFILMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2018

SISTEM KERNEL P - PENGOPTIMUM MULTI OBJEKTIF KUMPULAN ZARAH PENDUAAN BAGI CIRI KAEDAH PEMILIHAN DALAM TATASUSUN DATASET KANSER

NAEIMEH ELKHANI

TESIS YANG DIKEMUKAKAN UNTUK MEMPEROLEHI
IJAZAH DOKTOR FALSAFAH

FAKULTI TEKNOLOGI DAN SAINS MAKLUMAT
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2018

DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

06 July 2018

NAEIMEH ELKHANI
P72752

ACKNOWLEDGEMENT

First and foremost, I adore Almighty Allah for His mercies on me. If not for His benevolence, my dream of becoming a PhD holder someday would have been a mirage.

I would also like to register my indebtedness to my supervisor and advisor, Associate Prof. Dr. Ravie Chandren Muniyandi for his supports right from the day I registered in UKM. Because Associate Prof. Dr. Ravie Chandren Muniyandi is a goal-getter, he has always been aggressive towards ensuring that jobs were not only done promptly but were also done very well. He was always available to share his wealth of experience on Membrane Computing with me any time, any day. I will forever be grateful, Sir.

Also, I would like to express my high appreciation to my life advisors Mr. Y.S. Yap, Mr. B.Y. Liew, and Mr. Peter Ong who believed on me and supported me to finish my PhD.

Last but not least, I gratefully acknowledge financial support provided by the Science Fund of the MOSTI - Ministry of Science, Technology, and Innovation (Malaysia; Grant code: 01-01-02-SF1104).

I thank you all

ABSTRAK

Kanser merupakan isu utama kesihatan awam di merata dunia disebabkan oleh kesukaran dalam pengesanan awal. Diagnosis awal berdasarkan profil ekspresi gen telah menunjukkan sumbangan penting bagi kemajuan kajian kanser. Tatasusunan mikro DNA ekspresi gen ialah alat yang paling biasa digunakan yang mampu untuk memantau tahap ekspresi beribu gen secara serentak. Kesukaran utama dalam teknik ini ialah terdapat sebilangan besar gen dibandingkan dengan saiz-saiz sampel kecil gen di mana ianya memberi kesan impak negatif pada tahap kelajuan dan ketepatan teknik. Pemilihan relevan gen-gen merupakan tugas penting bagi pengelasan sampel dalam data tatasusunan mikro di mana penyelidik-penyelidik cuba mengenalpasti kemungkinan set gen-gen yang paling kecil masih boleh mencapai prestasi ramalan baik. Dalam kajian ini, pengkomputeran membran digunakan untuk meningkatkan ketepatan dan kelajuan dalam ciri pilihan dan kaedah pengelasan berkaitan dengan set data kanser. Cadangan model tersebut terdiri daripada tiga bahagian utama di mana pertama sekali, tesis ini memperkenalkan kriteria baru untuk mereka bentuk dan membangunkan KP-MObPSO yang menyerupai pilihan ciri jenis pembungkus. Peraturan bahagian, menulis semula dan input/output digunakan untuk membuat satu tindak balas antara dalam gen dan zarah-zarah. Kedua, satu pemilihan ciri tertanam dan kriteria klasifikasi dibangunkan berdasarkan sistem KP. Dalam bahagian kedua, set-set gen penanda dikeluarkan oleh bahagian tertanam model menunjukkan lebih kestabilan dan kebolehpercayaan berdasarkan ukuran ROC serta kadar ralat lebih baik dibandingkan dengan bahagian kertas pembungkus model. Akhirnya, disebabkan ciri keselarian besar-besaran sedia ada pengkomputeran membran, mana-mana model inspirasi pengkomputeran membran hanya boleh mewakili sepenuhnya pengiraan model dalam kes menggunakan platform selari. Cadangan model tersebut di aplikasi di atas dataset kolorektal dan dataset dada yang mengandungi 100 gen berserta 6 sampel. Dalam bahagian pertama, pilihan ciri model KP-MObPSO mengatasi ketepatan tulen “Pure-MObPSO” yang diukur oleh mesin vektor sokongan. Cadangan KP-MObPSOmodel yang dilaksanakan ke atas pelbagai teras dan unit pemprosesan (GPU) grafik adalah untuk meningkatkan kelajuan pelaksanaan. Kadar ralat terendah oleh model tertanam dipamerkan sebagai 0.1111 untuk kanser payudara dan 0.0769 untuk data kolorektal. Walaupun pelaksanaan cadangan di pelbagai teras tidak mampu mengurangkan kos masa nyata sekali. Pelaksanaannya di NVIDA Geforce 680 GPU menunjukkan satu penurunan ketara kos masa 164 saat untuk bebas 100 kali lelaran berbanding dengan 25 min di unit pemprosesan pusat (CPU) untuk 25 butir zarah dalam 100 kali lelaran. Inspirasi ciri pilihan Membran yang diperkenalkan dan kaedah pengelasan untuk set data kanser mencapai prestasi lebih baik dalam soal ketepatan dan kos masa daripada kaedah pengoptimuman tulen.

ABSTRACT

According to the statistics, cancer related diseases are the most challenging health problem in all over the world which leads to fatality in the case of late diagnosis. Diagnosis tools based on gene expression profiles have shown significant contribution to the progression of cancer studies. DNA microarrays gene expression is the most commonly used tool because of its ability to monitor few thousand of genes at the same time in their expression level. The main difficulty in this technique is that there are large number of genes (features) compared to the small sample sizes which makes negative impact on the speed and accuracy of technique. Selection of relevant genes is the crucial task for sample classification in microarray data and many research in this field try to extract the smallest group of genes those can provide good diagnosis result. In this study, membrane computing is used to improve accuracy and speed in feature selection and classification methods related to cancer datasets. The proposed model consists of three main part. Firstly, the thesis introduces new criteria to design and develop KP-MObPSO which resembles a wrapper type feature selection. Division rule, rewriting and input/output are used to make an interaction among the genes inside and between the particles. Secondly, an embedded feature selection and classification criteria developed based on KP system. In the second part, the marker gene sets are extracted by the embedded part of the model indicate more stability and reliability based on ROC measure as well as better error rate in compared to wrapper part of the model. Finally, due to the inherent large-scale parallelism feature of membrane computing, any membrane computing inspired model can fully represent this computation model only in the case of using parallel platform. The proposed model applied on the colorectal and breast dataset contains 100 genes with 6 samples. In the first part, KP-MObPSO feature selection model outperforms accuracy of Pure-MObPSO measured by support vector machine (SVM). The proposed KP-MObPSO model implemented on multi-core and graphic processing unit (GPU) to improve the speed of execution. The lowest error rate by embedded model displayed as 0.1111 for breast cancer and 0.0769 for colorectal data. Although the execution of the proposed on multi-core was not able to decrease the time cost significantly, its execution on NVIDA Geforce 680 GPU demonstrates a significant drop of time cost as 164 sec for independent 100 times iterations in compared to 25 min on the central processing unit (CPU) for 25 particles in 100 times iteration. The introduced membrane inspired feature selection and classification method for cancer datasets achieved better performance in terms of accuracy and time cost than pure optimization method.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

Figure No. **Page**

LIST OF ABBREVIATIONS

PSO	Particle Swarm Optimization
MOBPSO	Multi Objective Binary Particle Swarm Optimization
KP	Kernel P System
SVM	Support Vector Machine
SVM-RFE	Support Vector Machine-Recursive Features Elimination
KP-MObPSO	Kernel P System- Multi Objective Binary Particle Swarm Optimization
GASVM	Genetic Algorithm Support Vector Machine
KNN	K-nearest neighbour
IG	Information Gain
HPSOTS	Hybrid Particle Swarm Optimization Tabu Search
CRC	Colorectal Cancer
FS/CL	Feature selection/Classification
GPU	Graphic Processing Unit
CPU	Central Processing Unit
SNR	Signal Noise Ratio
POS, yy	Position
P	Particle Number
Max_c	Maximum number of genes inside particle
c	The number of genes selected in each particle out of max_c
NGENES	Genes with position 0 or 1
NewNGENES	Genes with position 1
Q	Gene ID objects involving the IDs of selected genes in each particle
C	Dissimilarities of genes measured by correlation
sum_diss	Sum of dissimilarities
sum_snr	Sum of SNR
FITs	Fitness values

ERR	Error Rate
pBestScore	Local best scores
gBestScore	Global best scores
pBest	Indicator for local best score
gBest	Indicator for global best score
a	Dataset
Vel	Velocity
Rand	Random
Master	Master membrane
Ave	Average
ROC	Receiver Operating Characteristic curve
SMO	Sequential Minimal Optimization
Fn	Function
EVs	Evalutionary Algorithms
GSA	Genetic Swarm Algorithm
LMSL	large-margin subspace learning
LLFS	local linear feature selection

CHAPTER I

INTRODUCTION

1.1 MOTIVATION

According to the statistics, cancer related diseases are the most challenging health problem in all over the world which leads to fatality in the case of late diagnosis. It varies to different cancer types and cancer types differ in terms of stages. According to the Malaysia National Cancer Registry Report 2016, cancer is undeniably one of the most important non-communicable diseases in Malaysia and contributed to 13.56% of all deaths occurred in the Ministry of Health Hospitals in 2015 (National Cancer Registry Report (MNCR), http://www.cancer.org.my/). According to the report published in MNCR, in the period of (2007-2011) 103,507 numbers of new cancer patients are registered as diagnosed cases. In terms of gender, 46,794 (45.2%) were male cancer patients and 56,713 (54.8%) were female cancer patients. Therefore, according to the report, the risk of affect to cancer in male Malaysian was 1 case in 10 people and in females Malaysian was 1 lady in 9 ladies. In the developed countries, such as the US, the same condition is going on. In the US and in the year 2017, 1,688,780 number of new cancer patients are diagnosed and it is estimated 600,920 number of fatality will occur from the total amount of diagnosed cases (Siegel, Miller et al. 2017). According to the available cancer statistics, in the period of (2004-2013) the number of female cases diagnosed were constant and the number of male cases recognized decreased about 2% annually, while in terms of fatality, in the period of (2005-2014) death rate decreased by approximately 1.5% per year in both male and female cases (Siegel, Miller et al. 2017). Although, there was a continuous decline in the fatality of cancerous cases, it was mostly because of sever screening and early diagnosis rather than capability to manage and cure the cancer types effectively. As far as computer science field is concerned, more robust diagnosis methods will be helpful to decline the fatality of cancerous cases.

In simple explanation, cancer is defined as the cell-related disease. The duties of cells are to work to replace exhausted cells, recover damaged ones and contribute to organism’s growth. In the center of all cells there is a nucleus which is constituted of DNA. DNA encodes program that is needed for the development of future organisms. DNA is composed of two parts; one part is called coding DNA and the other part called non-coding DNA. The coding part is also named as genes which are responsible to do fundamental works in organisms. The problem occurs when these genes duplicate or grow abnormally inside a lump which becomes cancer. Generally, cancer related issues consist of two types. The fist type of issue is known to *Benign* which a cancerous issue is not actually and does not spread to other organs of body, although some of the subtypes can be precancerous and may develop as cancer if it is not diagnosed and cured. The second type, which is cancerous, is called *Malignan* and has potential to afflict other organs like an invasive if it is not treated in early stages. Therefore, our focus is the second type as explained before.

The situation of cells at molecular level builds a collection which is called gene expression profile. Diagnosis tools based on gene expression profiles have shown significant contribution to the progression of cancer studies. DNA microarrays gene expression is the most commonly used tool because of its ability to monitor few thousand of genes at the same time in their expression level (Schena, Shalon et al. 1995, Harrington, Rosenow et al. 2000). Through this capability many machine learning techniques are developed for computational analyses. These methods are useful to extract the most significant genes and make a pattern of classification in the gene levels which got tremendous contribution in cancer diagnosis and prediction (Giallourakis, Henson et al. 2005, Shang and Shen 2005, Rocha, Mendes et al. 2007, Brazma 2009) and prognosis (Bard and Hu 2011, Gupta, Kumar et al. 2011, Vanneschi, Farinaccio et al. 2011). Cancer has been a perfect candidate for evaluation by microarrays (Wu, Dong et al. 2017). Literature strongly believes that microarray data is capable to provide significant contribution in diagnosis, prognosis of cancers (Van't Veer, Dai et al. 2002) and classification of human cancers (Perez-Diez, Morgun et al. 2007). Therefore, the microarray cancer dataset has chosen to implement the proposed diagnosis method.

All these techniques are facing a common challenge which is having quite high number of genes (features) against usual small sample size (Piatetsky-Shapiro and Tamayo 2003, Rocha, Mendes et al. 2007). This issue makes negative impact on the speed and accuracy of techniques. In microarray data, efficient and effective management of gene's (feature's) datasets in terms of accuracy and time-cost becomes increasing challenging with respect to both high dimensionality and small sample size barriers in datasets. To tackle with these two barriers, dimensionality reduction is considered as a solution in literature and the basis of our proposed diagnosis method will be on dimensionality reduction concept.

1.2 RESEARCH BACKGROUND

[Remove: The source of high dimensionality problem is this fact that the collected data is usually associated with a high level of noise resulted mainly from both imperfection in the technologies that collected the data and the source of the data itself.] [Add: The source of high dimensionality backs to high level of noise in the collected data. The noise problem itself comes from the imperfection in the technologies as well as type of data]. In the field of dimensionality reduction feature selection is the most popular technique to remove noisy (i.e. irrelevant and redundant) features. Feature selection approaches aim to select a small subset of features that minimize redundancy and maximize relevance to the target such as the class labels in classification. A relevant feature is neither irrelevant nor redundant to the target concept; an irrelevant feature is not directly associates with the target concept but affect the learning process, and a redundant feature does not add anything new to the target concept (Piatetsky-Shapiro and Tamayo 2003). Therefore, gene selection as dimensionality reduction technique is a critical task which aimed to enhance the classification performance as well as improving the accuracy of the methods (Shang and Shen 2005, George and Raj 2011).

Feature (gene) selection is capable of improving learning ability of the methods which leads to more accurate and less complex method (Almuallim and Dietterich 1994, Koller and Sahami 1996). All the efforts have been done in the machine learning feature selection and classification area is mainly related to study, understand, model, simulate and implement these techniques through imitating the way nature computes (Păun 2005). The way nature computes are the fundamental concept in natural computing. Indeed, natural computing attempts to build new methods of computation which are inspired by nature's dynamic processes (de Castro 2007). The methods of natural computing are involved Evolutionary Algorithm, Neural Network, DNA Computing and Membrane Computing.

These two latter methods of natural computation- DNA Computing and Membrane Computing -provide novel methods with high efficiency. The most prominent feature of these methods is trading space to decrease computation time, in other words, they are able to create an exponential workspace in polynomial time.

DNA computing is based on molecular actions and is a computational paradigm in which data is encoded as biomolecules such as DNA stands to perform various operations. The aim of DNA computing is to solve hard problems in a feasible time which is gained through massive parallelism by storing information on DNA molecules (storing information at bit levels and improve efficiency by using silicon). The first practical application of DNA computing to link genes with disease initiated at university Tokyo (2002) (Rani and Jain 2014). Their attempt was to develop a method that can synthesize around 10,000 different DNA strands which are known to bond with genes related to specific diseases such as cancer (Rani and Jain 2014).

Membrane computing is initiated based on cells and higher structures of living cells, such as tissues and organs, Păun in 2000 (Păun 2000). The models of membrane computing are called P systems that are parallel computing devices. P systems have been proved to be a rich framework for handling many problems related to computing in a feasible time such as solving NP-complete problems (Li and Yang 2002, Stephanopoulos, Hwang et al. 2002, Li 2006) or PSPACE-complete problems (Nguyen and Rocke 2002). Many membrane algorithms have been proposed for solving various optimization problems, such as knapsack problem (Tan, Shi et al. 2004), point set matching problem (Chen, Zhong et al. 2016), numerical optimization problem (Wang, Makedon et al. 2004), multi-objective optimization problem (Symons and Nieselt 2006, Lee, Lin et al. 2011), DNA sequence design problem (Holland 1975), and many practical problems (Goldberg 1989, Liu, Cutler et al. 2005)."

Moreover, membrane computing is used for optimization problems to improve efficiency and accuracy (Xiao, Huang et al. 2014, Zhang, Rong et al. 2014)."Also, potentials of membrane computing is already developing in other bio-applications to improve efficiency and accuracy such as HIV infection (in Edinburgh, e.g. (Frisco and Corne 2007)), photosynthesis (Nishida 2002), Brusselator (e.g. (Suzuki 2007)), imitating of p53 protein paths via a P system (Suzuki and Tanaka 2006), demonstrating epidermal growth factor receptor in the field of signalling network (Pérez-Jiménez and Romero-Campero 2005),"quorum sensing in bacteria (in Nottingham, Sheffield, Sevilla universities, e.g. (Chan, Chin et al. 2015), circadian cycles (in Verona university, e.g. (Kon, Ode et al. 2017)), apoptosis (in Ruston-Louisiana university, e.g. (Păun, Pérez-Jiménez et al. 2006), signaling pathways in yeast (in Milano, e.g. (Pescini, Cazzaniga et al. 2012)), peripheral proteins in Trento, e.g. (Cavaliere and Sedwards 2007), Mechanosensitive Channels (Fernandes, Guseva et al. 2017), Immunity (Ciobanu 2006).

Assigning discriminatory genes (that are resulted from feature selection technique) to the specific class label make the pattern of classification. The main challenge in classification is to assign a new set of recognized significant genes to the right class of normal genes or cancerous genes. A training set of genes will be trained and then will be tested by test set of genes to show how they have categorized correctly. To help correct function in classification, feature selection aims to highlight the most significant genes out of total number of genes dataset. It means a robust feature selection method should be capable of distinguish between different samples of genes those belong to the same class either normal or cancerous class. Both normal and cancerous classes are labeled to add the relevant genes which have been recognized by feature selection method. For instance, a gene called g_i belongs to the class of normal genes labeled as c_i if g_i and c_i are correlated. The importance of feature selection method is that, in most of classification issues removing not relevant and redundant genes in huge dataset will influence on building a good classifier. The most important influence is related to reducing time cost of classifier in the case not relevant and redundant genes are removed by feature selection method before sending them to classifier (Alelyani, Tang et al. 2013) and building a general classifier without overfitting issue (Alelyani, Tang et al. 2013) as well as more accurate classifier (Janecek, Gansterer et al. 2008).

Achieving high classification accuracy via a set of genes which are suggested by a well-designed feature selection is the main objective has followed by many researchers (Li and Yang 2002) but the challenge is not only the number of genes are suggested as significant genes but also the algorithms will be applied to distinguish the significant set of genes. Another problem arises due to the small sample size of gene datasets when researchers try to achieve a high classification accuracy and it cause trapping them in overfitting issue. Over-fitting (Mitchell 1997) is an unwanted consequence of high accuracy. It occurs when the accuracy results achieved through a dataset is not repeatable by another dataset and clearly the achieved performance cannot be generalized to different dataset of another cancer type. Even worst, the results are not repeatable by a new dataset of the same type of cancer dataset and more likely it leads to misclassification. Thus, overfitting is a well-known risk in the field of classification accuracy as researchers try to best adjust their proposed method to the chosen dataset. There are some methods which have been proposed to train the classifier effectively to prevent over-fitting issue (Dietterich 1998, Nadeau and Bengio 2000). In this regard, a well-designed feature selection method for classification will be helpful to prevent overfitting and at the same time predicting the best performance of the method.

Due to the inherent large-scale parallelism feature of membrane computing, any membrane computing inspired model can fully represent this computation model only in the case of using the parallel platform. From the beginning of introducing this model, it was a big concern in all membrane related studies. For instance, to fully implement parallelism of such membrane computing model and to support an efficient execution (Van Nguyen and Gioiosa 2010) used a platform based on reconfigurable hardware. Without parallelism, all subsequent studies face a challenge of how to make rules available in all steps of computation. In (Alhazov, Freund et al. 2012) a sequential computing of membrane computing, they just had an option of using one membrane and made the rules periodically available based on time-varying sequential P system. At least through choosing a membrane rule from a set of rules and make a minimum parallelism of rules, e.g., (Ciobanu, Pan et al. 2007) with the active membrane and solving NP-complete problems through trading place against time to make a more efficient model of membrane computing.

Recently, several studies attempted to utilize parallel membrane computing to improve intelligent algorithms. For instance, multi-core processing used in the study of (Maroosi and Muniyandi 2013) utilized a membrane computing inspired genetic algorithm and (Maroosi and Muniyandi 2013) have highlighted parallelism in membrane computing in the case of solving the N-queens problem.

The architectural differences between CPUs and GPUs cause CPUs to perform better on latency-sensitive, partially sequential, single sets of tasks. In contrast, GPUs performs better with latency-tolerant, highly parallel and independent tasks. In the literature there are some efforts to parallelize methods such as parallel feature selection and classification methods (Slavik, Zhu et al. 2009) and efforts on parallelization of intelligent optimization algorithms, such as Parallel genetic algorithm on CPUs/computers to identify informative genes for classification (Liu, Iba et al. 2001, Sarkar, Sana et al. 2011), parallel Genetic algorithm on GPU (Li, Wang et al. 2007, Cano, Zafra et al. 2010, Pospichal, Jaros et al. 2010), and Parallel PSO on GPU (Zhou and Tan 2009, Mussi, Daolio et al. 2011, Kentzoglanakis and Poole 2012, Nobile, Besozzi et al. 2012, Nobile, Besozzi et al. 2013).

1.3 STATEMENT OF THE PROBLEM

The potentials of membrane computing model have been proved in the literature to improve the performance of optimization (available in section 2.5) and evolutionary algorithms (available in section 2.6) as well as to solve biological and NP-complete problems (available in section 2.2). Moreover, parallel stucture of membrane computing (available in section 2.11) and mimicking the biological nature (available in section 2.10.1) of cells non-deterministic approach (available in section 4.2.1.1.6) can be helpful to tackle with the deficiency of feature selection and classification methods in the problems of stable accuracy and time efficiency. Therefore, proposing a membrane inspired feature selection and classification method in the concept of microarray cancer dataset (available in section 2.12.1) which covers all the potentials of membrane computing can provide basis of a solution for the previous problems of this field.

Traditional feature selection for classification methods

1. In the literature (Alelyani, Tang et al. 2013), from one point of view, methods of feature selection for classification (available in section 2.7) can be divided into three families 1) methods for flat features (filter models, wrapper models, embedded models), 2) methods for structured features (graph structure) and 3) methods for streaming features. A main disadvantage of the filter approach despite its lower time consumption is the fact that it does not interact with the classifier, usually leading to worse performance results than those obtained with wrappers. However, the wrapper model comes with an expensive computational cost, which is particularly aggravated by the high dimensionality of microarray data. An intermediate solution for researchers can be the use of embedded methods that are usually a mix of two or more feature selection methods from different origins which use the core of the classifier to establish a criterion to rank features. But it is worth noting that the embedded method such as SVM-RFE, in spite of the fact that it is, in theory,

better than the filter methods, achieves comparable or even worse results than them in terms of classification accuracy. Graph Structure is another method of feature selection for classification. Many biological studies have suggested that genes tend to work in groups according to their biological functions, and there are some regulatory relationships between genes (Alelyani, Tang et al. 2013). Therefore, embedded and graph-based method has chosen as candidate in this study.

Intelligent feature selection for classification methods

2. From another perspective for the division of feature selection and classification methods, considerable number of hybrid intelligent optimization algorithms have been developed widely based on biology intelligence (available in section 2.7). Wide range of mixed methods are developed mainly based on evolutionary learning methods such as genetic algorithm (GA), neighborhood search like K nearest neighbor (KNN) and swarm intelligence algorithms such as particle swarm optimization (PSO).” Based on our review on GA and KNN, the problems of various proposed methods can be categorized to three parts. First; e.g., pure genetic algorithm generally has limitations such as 1) slow convergence, 2) lacks of rank based fitness function and 3) being a time-consuming approach. Mixed methods of GA and KNN were not capable to tackle with these problems completely. Second; in terms of classification accuracy, resulted accuracy in intelligent feature selection and classification algorithms varies greatly either in different types of cancer or different datasets, also due to building unstable method overfitting risk increases dramatically when they examine on high density datasets like microarray dataset. PSO because of first; its ability to match with graph model as genes (nodes) and define relationship between them (edge), second; higher accuracy in compared with flat (filter and wrapper) methods third; reasonable time complexity on CPU is our candidate in proposing a membrane-inspired feature selection method.”

Traditional deterministic methods

3. Another problem rises from the lack of imitating biological process as they really are. From biological point, all the actions in biological-based systems happens in discrete and nondeterministic approach (available in section 2.9). However, in traditional methods such as Ordinary Differential Equations (ODE) approach procedures are handled in continuous and deterministic approach which totally ignores the real actions and reactions as they really are in biological systems (Chandren and Abdullah 2011). The limitation of non-spatial methods is that they can only be accurately applied to spatially homogeneous systems, but this assumption does not hold in many (or even most) cases of interest. For instance, the membrane of the cell is an extremely heterogeneous environment, and even the cytoplasm contains many macromolecules that impede diffusion (Sturrock 2016). Our candidate method is non- deterministic- spatial discrete stochastic approach- to be defined on compartment-based structure of membrane computing.

Parallelism

4. Computational cost is a big challenge for almost all intelligent algorithms which are run on CPU (available in section 2.11). Recently new attempts have been started to develop parallel feature selection and classification methods such as (Slavik, Zhu et al. 2009) and some efforts are focused on parallelization of intelligent optimization algorithms, such as parallel genetic algorithm on CPUs/computers to identify informative genes for classification (Liu, Iba et al. 2001, Sarkar, Sana et al. 2011), parallel Genetic algorithm on GPU (Li, Wang et al. 2007, Cano, Zafra et al. 2010, Pospichal, Jaros et al. 2010), parallel PSO on GPU (Zhou and Tan 2009, Mussi, Daolio et al. 2011, Kentzoglanakis and Poole 2012, Nobile, Besozzi et al. 2012, Nobile, Besozzi et al. 2013) and parallel processing of microarray data (Guzzi, Agapito et al. 2014). In this regard, our proposed membrane-inspired feature selection method should perform with an efficient time that we aimed to settle via using potentials of membrane computing in parallel processing and nondeterminism. Based on our best knowledge there is not any parallel membrane-inspired feature selection and classification method in microarray cancer studies.

1.4 RESEARCH QUESTIONS

Therefore, our main aim is to introduce new application of membrane computing in cancer studies through improving the accuracy and time complexity of feature selection and classification methods. For this aim following questions will address:

i) How to design a membrane-inspired feature selection method in non-determinism and and optimized approach to improve accuracy and time efficiency?

ii) How to strengthen the proposed membrane-inspired feature selection method with a membrane-inspired classification method to achieve sufficient accuracy and avoid overfitting?

iii) How to use parallelism privilege of membrane computing to design and implement the proposed membrane-inspired feature selection and classification method in parallel environment to improve time efficiency?

1.5 AIM AND OBJECTIVES

As discussed earlier, more investigation is required in feature selection and classification of microarray dataset. Therefore, the aim of this research is to build more accurate, stable and time efficient feature selection and classification model in microarray dataset. To achieve these goals, the following specific objectives were set:

1. To propose a membrane-inspired feature selection method in non-deterministic and optimized approach to improve accuracy and time efficiency

2. To develop an embedded membrane-inspired feature selection and classification method to achieve sufficient accuracy and overcome overfitting

3. To evaluate the proposed membrane-inspired feature selection and embedded membrane feature selection and classification method in parallel environment.

Objective one will answer research question one, objective two will respond to the second research question and objective three will meet the third research question. Objective one meets the problem statements of 1, 2 and 3. Objective two is to answer the problem statements of 1 and 2. Objective three will cover the problem statement of 4.

1.6 RESEARCH OUTLINE AND SCOPE

The research activities to achieve the research objectives of this study are categorized as follow:

Firstly, advantages and disadvantages of different feature selection for classification methods were investigated to find out the most robust feature selection method as well as the most suitable one to be adapted with membrane system in terms of the definitions of objects, rules and structure. The wrapper feature selection method has chosen to design membrane- inspired feature selection method. Second, from intelligent algorithm point of view to develop a feature selection method, PSO method has been selected and more studies on evolutionary algorithms lead to the selection of MObPSO method which has graph-based concept. Third, a review has been done to select a suitable variant of P system to design membrane-inspired MObPSO method. After investigation on different variants of P system, the tissue like P system as kernel P system (KP) selected which have graph based fundamental. All the rules of KP system as rewriting and communication rule, division, input/output, link creation have used to develop the proposed model. The proposed KP-MObPSO feature selection method developed by MATLAB R2014a and to define the objects inside each compartment we have used object-oriented coding for the proposed model. The proposed model examined by two microarray cancer datasets as breast cancer and colorectal cancer. To do experiment, cell line dataset of colorectal cancer and breast cancer are downloaded from publicly available datasets in Gene Expression Omnibus (GEO) repository (http://www.ncbi.nlm.nih.gov/gds). After identifying the significant genes through the proposed KP-MObPSO feature-selection method, highlighted genes were classified using classification method: support vector machines in Weka 3.6.9 software (http://www.cs.waikato.ac.nz/ml/weka/documentation.html). Using Weka

3.6.9, n-fold cross validation and 70% split training/test dataset methods assessed the classification accuracy of the proposed feature-selection method relative to MObPSO through performance measures, such as F-score, precision, and recall. In this stage the objective was to execute the proposed model in sequential which is done by a computer with two cores, the Intel Core-i5-2450, 2.5-GHz CPU with 4 GB RAM. The time complexity of the proposed KP-MObPSO feature-selection method and pure MObPSO feature-selection method calculated by big O notation. The accuracy and time complexity of KP-MObPSO compared with MObPSO in colorectal cancer and breast cancer context.

Forth, due to the weaknesses of filter and wrapper feature selection methods as is discussed in problem statement section, we aimed to take advantages of embedded feature selection methods and classification method to strengthen the proposed KP-MObPSO feature selection method. Kernel P system from the family of tissue like p system is chosen to make the embedded model with MObPSO feature selection method. Criteria of embedded methods and various kernel P system rules for designing an embedded KP-MObPSO were investigated with the aim of understanding design criteria. Support vector machine (SVM) and nested embedded method are chosen as criteria. To develop KP-MObPSO, boundaries are as mentioned in third part in previous paragraph.

Fifth, to make a fully membrane inspired model, the designed models are implemented in parallel environment. To do this multi-core processing and GPU are chosen with the aim of taking advantage of parallelism characteristic of a membrane inspired model in executing rules in parallel as well as to improve the time cost of execution. In this stage first; implementation of both KP-MObPSO feature selection method and KP-MObPSO-SVM embedded feature selection and classification method are done in multi-core processing. We define parallel configuration on MATLAB by cluster, and we have defined jobs and job’s tasks in the MATLAB client. As predefinitions, the number of physical cores on system indicates the maximum number of workers in MATLABPOOL. In this study, there are 4 cores in our machine. The proposed KP-MObPSO feature selection methods assume to have 25 particles. To configure 25 particles on 4 cores of the system, we assume to have 4 compartments including 6 particles, 6 particles, 6 particles and 7 particles respectively. Each particle consists of a random number of gene objects from 1 to 100 and sets of rules will be executed on objects. Since then, we will assign each rule to a specific job and each job will assign to each compartment through a set of 4 tasks repeating execution of rules on each object inside a particle. Thus, the multicore KP-MObPSO model will be initialized by:

4 compartments including 6, 6, 6 and 7 particles

4 jobs/rules inside each particle

4 tasks assign to each rule/job to be executed on particle inside the 4 compartments

The experiment hardware included a computer with the Intel Core-i5-6200U, 2.40-GHz CPU and 12 GB RAM. The time complexity of the proposed KP-MObPSO-SVM feature selection and classification method and KP-MObPSO feature-selection method calculated by big O notation.

At the second stage, implementation of both KP-MObPSO feature selection method and KP-MObPSO-SVM embedded feature selection and classification method are done on GPU. To design KP-MOBPSO-SVM model on GPU, two important points are concerned, first, the dependency between the objects and rules to decrease the rate of communication, second; access to the lesser cost memories in the execution of threads like local and shared memory. First the program written in MATLAB 2014a for embedded KP-MObPSO-SVM transferred to the C++ programs to use the parallel platform. This step is necessary because a CPU based programming cannot be simply run on GPU. After translate the program to C++ first it should be run in sequential approach to make sure of correct debugging and running. Then it needs analysis of the algorithm to find parallelization opportunities. The GPU processing was done on a NVIDA Geforce 680 GPU with computation capability 3.

1.7 STRUCTURE OF THE THESIS

This thesis includes seven chapters. First chapter deals with the motivation, aim and objective of this research work as well as the research outline and scope. Chapter two introduces the concepts related to membrane computing, feature selection methods, evolutionary algorithms with the mix of p systems and parallel computing. This chapter present the literature and investigation done on the existing membrane inspired evolutionary algorithms, feature selection methods and methods of parallel simulation to fulfil parallelism aspect of membrane computing.

The methodology that was adapted to carry out the research based on the objectives and scopes outlined in the first chapter is introduced in chapter three. This chapter illustrate the procedures, tools and case studies used in the research. Chapter four investigates criteria for designing the KP-MObPSO feature selection method. The designed feature selection method examined by two microarray cancer datasets and its performance compared with the pure MObPSO feature selection method. In this stage implementation is done sequentially.

Chapter five investigate criteria and rules to embed the designed feature selection method with classification method. The designed KP-MObPSO system improved by SVM classification rules and the performance of KP-MObPSO-SVM method compared with KP-MObPSO via two microarray cancer datasets. In this stage implementation is done sequentially.

Chapter six illustrates a fully membrane inspired model with parallelism. The designed models are implemented on multi-core CPU and then on GPU with the aim of taking advantage of parallelism characteristic of a membrane inspired model as well as to improve the time cost of execution.

Finally, chapter seven summarizes some of the contributions of this research study. Outlines for future work are also stated in this chapter.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

The main topic of this study was the feature selection and classification of microarray cancer dataset by membrane computing. The membrane computing is a broad research field of natural computing and have many variants. In this study, we have used kernel P system as the variant of membrane computing. We have discussed different methods of feature selection for classification according to literature. Also, due to the parallel nature of membrane computing parallel environment such as GPU discussed in this chapter.

2.2 NATURAL COMPUTING AND MEMBRANE COMPUTING

Natural computing is a widespread research discipline that ties computer science to nature science (Brijder 2008). Two main research directions in the natural computing can be recognized. The first one deals with the computation processes taking place in nature, while the second one is related to the computational methods inspired by nature (Rozenberg 2008). In comparison to other technology, natural computing is considered as new; however, the yielded results are very promising. Natural computing attempts to integrate the computing performed in the computer science with that observed in nature (Sburlan 2006). Inspiring from the way nature works, makes it possible to develop tasks, such as resolving complex computational problem, which has resulted in promising outcomes in various areas (Arteta Albert 2010).

For instance, some natural networks were inspired by the way the human brain works. Neural Networks have parallel, distributed structure. They have ability to store knowledge that will be later used for learning (Arteta Albert 2010). These nature-inspired abilities of the neural network make it useful in areas, such as the image processing, the speech recognition and the control (Arteta Albert 2010).

Darwin's theory of Evolution, according to the "Natural Selection", have had an important role in understanding Nature's processes. The "Natural Selection" acts to preserve and accumulate minor advantages that enable a species to compete better in the wild are preserved (Arteta Albert 2010). Therefore, genetic algorithms employ the notion of natural selection, mutation and recombination from biology to solve optimization problems (Sburlan 2006).

Some of the natural computations are inspired by the molecular actions such as the DNA computing. In the DNA computing, data are encoded as bio-molecules to perform logical and arithmetic operations (Arteta Albert 2010). The DNA computing is efficient for solving complex NP-complete problems. The DNA molecules are used in the DNA computing as a basic element to make computational devices in which conventional electronic chips can be replaced by molecular chips to improve the efficiency of traditional computers. For example, the tools of molecular biology were used to solve an instance of the directed Hamiltonian path problem (Adleman 1994) and DNA experiments are proposed to solve the famous “SAT” problem of computer science (Lipton 1995). Furthermore, traditional computers have no parallelism introduced by the DNA computing (Arteta Albert 2010). Recently, the new area of membrane computing, introduced by (Păun 2000), is growing parallel to the DNA computing as a fast-emerging branch of the computer science (Păun 2000).

Many membrane algorithms have been proposed for solving various optimization problems, such as knapsack problem (Tan et al., 2004), point set matching problem (Chen, Zhong et al. 2016), numerical optimization problem (Y. Wang et al., 2004), multi-objective optimization problem (Symons and Nieselt 2006, Lee, Lin et al. 2011), DNA sequence design problem (Holland 1992), and many practical problems (Goldberg 1989, Liu, Cutler et al. 2005). On the other hand, many kinds of NP-problems have been addressed on membrane computing through P systems. The first solutions to NP-complete problems in membrane computing were designed in the cell-like model called P systems with active membranes (Díaz-Pernil, Pérez-Jiménez et al. 2008) and after that problems such as the Satisfiability problem, several numerical problems (Subset Sum, Knapsack, Partition, etc.), and also graph problems (3-Coloring, Clique, Vertex Cover, etc.). Moreover, Some NP-complete problems have been efficiently solved with tissue-like P systems: SAT (Păun, Pérez Jiménez et al. 2004), 3-coloring (Díaz-Pernil, Gutiérrez-Naranjo et al. 2007), Subset Sum (Díaz-Pernil, Gutiérrez-Naranjo et al. 2007) and a uniform linear-time solution to the Vertex Cover problem(Díaz-Pernil, Pérez-Jiménez et al. 2008).

Membrane system which is also known to P system have parallel structure and all the processes of rules and objects imitate the real function such some biological systems (Bianco 2007). Due to its biological concepts, membrane systems have been used to solve wide range of biological problems such as molecular interactions (Twycross et al., 2010; Muniyandi and Abdullah, 2012), bearded vulture evolution prediction (Cardona et al., 2009) and predator and prey relationship modeling. Moreover, membrane computing method has been used to solve other challenges, for instance computation of the threshold of two dimensional images (Christinal et al., 2010), segmentation of images (Christinal et al., 2011) and robot controlling (Buiu et al., 2012). In addition, it was capable of solving optimization problems, for example N-queens problem (Gutierrez-Naranjo and Perez-Jimenez, 2011), three coloring problems (Adrian and Florentin, 2012) and satisfiability problems (Ishdorj et al., 2010). Also, membrane computing methods are used to improve intelligent algorithms (Cheng et al., 2011; Zhang et al., 2011; 2012a; 2012b).

The basic concept of membrane computing divided to four parts: (i) structure of membrane including compartments, (ii) set of objectives are defined inside the compartments (iii) rules which are defined to make action and reaction between objects. In addition to the predefined objects inside the compartments, also objects can be produced by rules and rules can evolve objects and membrane as well (Păun, Perez-Jimenez et al. 2010). There are three types of membrane computing including cell-like, tissue-like, or spiking like computing models (Păun, Perez-Jimenez et al. 2010).

2.3 TISSUE LIKE MEMBRANE SYSTEM

In the tissue-like membrane systems, several cells share same environment (Martín-Vide, Păun et al. 2003, Song, Zhang et al. 2017). Similar to the cell inter-communication in tissues, the cells can communicate directly with each other through the channels among them. A tissue-like membrane system with the degree $m \geq l$ is expresses as follows:

$\pi = (O, E, w_1, ..., w_m, R, i_{out})$

where:

1. m is the number of cells in the system;

2. O is the finite non-empty alphabet of objects;

3. $E \subseteq O$ is the set of objets present in the environment;

4. $w_1, ..., w_m$ are string over O, representing multi-sets of objects associated with m cells at the initial state of the computation;

5. R is a finite set of communication and transformation rules in the following forms:

The transformation rules $x \rightarrow y$ allow the cell $i \in O$ to consume a multiset x to produce a new multiset y inside the cell i;

The communication rules $(i,u/v,j)$ for $i,j \in \{0,1,2,...,m\}$

, $i \neq j$ and $u,v \in O$;

6. $i_{out} \in \{0,1,2,...,m\}$ is the output cell.

A tissue-like membrane system of degree m is a set of m cells (each one consisting of an elementary membrane) labeled by *1,2, ..., m*. Here, 0 refers to the label of the environment and i_{out} denotes the output region, which can be the region inside a cell or the environment. The strings $w_1, \ldots, w_m$ describe the multisets of objects placed in m cells of the membrane system and $E \subseteq O$ is the set of objects placed in the environment. Each set is available in arbitrary large amount of copies.

The communication rule *(i,u/v,j)* can be applied over two cells labeled by i and j, so that u is located in the cell i, and v is located in cell j. The communication rule denotes that the objects of the multi-sets, represented by u and v, can be interchanged between the two cells. When either $i=0$ or $j=0$, the objects are interchanged between the cell and the environment. The rules are used in the framework of membrane computing, that is in the maximum parallel way (a universal clock is considered). Each object in a membrane can be used only in one rule, which is non-deterministically chosen when there are several possibilities. But every object should participate in a rule of any form, meaning that in each step, a maximal set of rules should be applied.

2.3.1 Kernel P System

As a variant of P system, kernel P system (KP system) introduced for the first time in the study of (Gheorghe, Ipate et al. 2013, Gheorghe, Ceterchi et al. 2017). This variant of P system integrates most of the features of membrane computing which have been successfully used for modelling problems and are applied in various application. There are many case studies which kernel p system was selected as most suitable variant of p system to model and solve the problem, for instances (Martin-Vide, Pazos et al. 2002, Păun, Pérez Jiménez et al. 2004, Díaz-Pernil, Gutiérrez-Naranjo et al. 2007, Díaz-Pernil, Gutiérrez-Naranjo et al. 2007, Díaz-Pernil, Pérez-Jiménez et al. 2008, Gheorghe, Konur et al. 2017). The main structure of kernel p system includes compartments which have designed in graph type of presentation.

In addition to well-known features of membrane computing which are integrated in KP system, there are some new features as well like division feature. Most importantly, 1) kernel p system integrates previous and new features into a coherent and comprehensive formalism, 2) kernel p system is very flexible in the modelling aspect in the way any required feature and constraints can be added to the modelling even though it was not defined in the beginning of modelling, moreover 3) designing model with KP system is simpler and steps are shorter while at the same time provide more clarification where the system getting complex.

Kernel p system as its definition, concept basically is near to tissue P systems and from structure point of view has graph-based structure. Objects are like nodes of the graph and the rules define the interaction between nodes which represent by edge of the graph. Therefore, objects and rules have labelled and methods and functions will define how a set of rules will consume a set of objects to produce new set of objects or only for communication purpose. Generally, there is two types of the rules in KP systems: first type of rules deals with the objects to transfer them between compartments or send the objects from compartment to environment and vice versa; second type of rules deal with the membrane structure to change the topology of the compartments. Both type of rules includes with a guard which indicates when the statement of the guard is correct rules are allowed to apply on objects. Any set of rules can assign to any compartment independently. First type of rules composite of rewriting and communication, symport and antiport. Second type of rules include division, link creation, dissolution.

According to (Gheorghe et al., 2013), A KP system of degree n is a tuple, $k_{\Pi} = (O, \mu, C_1, . . . , C_n, i_0)$, where O is a finite set of objects, called an alphabet; μ defines the membrane structure, which is a graph, (V,E), where V represents vertices indicating compartments and belongs to a set of labels $L(l_{i,}...)$, and E represents edges; $C_i= (t_i,w_i)$, $1 \leq i \leq n$, is a compartment of the system consisting of a compartment type from T and an initial multiset, w_i, over O; i_0 is the output compartment, where the result is obtained (this will not be used in this study). Each rule r may have a guard g, in which case r is applicable when g is evaluated to true. Its generic form is $r \{g\}$. KP systems use a graph-like structure (similar to that of tissue P systems) and two types of rules:

1) Rules to process objects: these rules used to transform object or to move objects inside compartments or between compartments. These rules are called rewriting, communication and input-output rules:

a. Rewriting and communication rule: $x \rightarrow y$ *{g}*, where $x \in A+$, $y \in A^*$, $g \in$ Finite regular Expressions *FE* over $(A \cup \bar{A})$; y at the right side defines as $y = (a_1, t_1) \ldots (a_h, t_h)$, where $a_j \in A$ and $t_j \in L$, $1 \leq j \leq h$, a_j is an object and t_j is a target, respectively.

b. The input-output rule: *(x/y) {g}*, where $x, y \in A^*$, $g \in$ Finite regular Expressions *FE* over $(A \cup \bar{A})$; means that x can be sent from current compartment to the environment or y can be brought from environment to the target compartment.

2) System structure rules: these rules make a fundamental change in the topology of the membranes for example with division rule on a compartment, dissolution rule on a specific compartment, make a link between compartments or dissolve the link between them. These rules are described as follow:

(c1) Division rule: $[]_{l_i} \rightarrow []_{l_{i_1}} \ldots []_{l_{i_h}}$ *{g}*, where $g \in$ Finite regular Expressions *FE* over $(A \cup \bar{A})$; means compartment l_i can be replaced with h number of compartments. All newly created compartments inherit objects and links of li;

(c2) Dissolution rule: $[]_{l_i} \rightarrow \lambda$ *{g}*; means compartment l_i is not exist anymore as well as all its links with other compartments.

(c3) Link-creation rule: $[]_{l_i}$; $[]_{l_j} \rightarrow []_{l_i} - []_{l_j}$ *{cg}*; means a link will be created between compartment l_i with compartment l_j. If there is more than one compartment with the label l_j, one of them will have a link with l_j non-deterministically.

(c4) Link-destruction rule*:* $[]_{l_i} - []_{l_j} \rightarrow []_{l_i}[]_{l_j}$*{cg}*; means the existence link between l_i and l_j will eliminate and there will not be any link between them anymore. The same as link creation, if there are more than one compartment which have a link with l_i then one of them will be selected non-deterministically to apply this rule.”

2.4 MULTI OBJECTIVE OPTIMIZATION

Multi objective optimization referes to the problem of finding a set of values which meet the limitations and are capable of optimizing the set of values to another set of values. Usually the criteria which makes the values are come from mathematic backgroud which conflict together. Thus, finding a solution for optimization problem means ultimately to find the value for each criteria in such a way it is acceptable for decision maker.

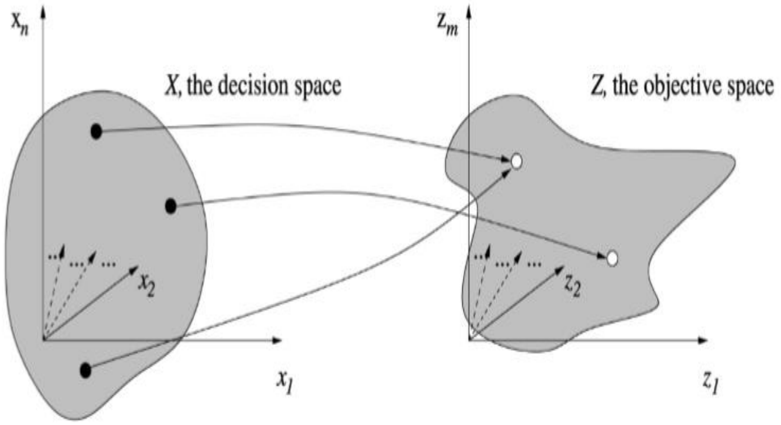


Figure 2.1 Multi Objective Optimization

According to Figure 2.1, the multi-objective optimization problem (MOOP) defines as the maximum value for the statement *{max* $(f_1(x),...,f_m(x))$ *s.t.* $x \in X\}$, where *X* indicates the space of solution value, while $f_j : X \rightarrow R$ $(j = 1,...,m)$ and *m* defined as $(m \geq 2)$. From the suggested solutions, it depends on the decision maker to choose which set of solutions, see, e.g., (Deb 2014, Deb, Sindhya et al. 2016, Azzouz, Bechikh et al. 2017).

There are three types of optimization methods, including lexicographic optimization method which does not assign the same priority to each solution but can be categorized as rank-based method, scalarizing optimization method which reduce a multi-objective problem to only one-objective problem. In this method, decision maker has to choose one approach of integrating problem to one- objective problem which usually are not able to do it. Due to these reasons it is better to put aside those methods require to assign a weight or rank to solutions and force the decision maker to choose one solution. Through these candidate solutions, decision maker will be able to choose the most efficient solution for the given problem according to the constraints.

The last type of method is pareto-type optimization method which is defined as follow:

Definition 1. According to Figure 2.1, If X represent a set of solutions then $x \in X$ consider as efficient solution regarding functions $f_1,...,f_m$ only if it does not dominated by any solution like y which is $y \in X$ $(y \neq x)$, in another word it means there is not any solution such as y which $y \in X$ and (i) $f_j(y) \geq f_j(x)$ for all $j = 1,...,m$, and (ii) for minimum one j, $f_j(y) > f_j(x)$. Therefore, there will be a set of solutions as indicate the points $(f_1(x),...,f_m(x))$ which is known to Pareto-optimal points.

The proof of the pareto-optimal solution explains as: assume there is a vector of functions as $(f_1,...,f_m)$ which have been given on X. In the case x^* is a solution which maximizes $u(f_1(x),...,f_m(x))$ on X then the in the function $u : R_m \rightarrow R$, x^* have to be efficient. Button to top, in the case x^* is efficient, then there will be a function like $u : R_m \rightarrow R$ which is able to maximize x^* in $u(f_1(x),...,f_m(x))$ and on X.

2.4.1 Binary Particle Swarm Optimization for Feature Selection

Basically, particle swarm optimization is a population-based type of optimization tool which has been used in real-number spaces. In the concept of PSO, every particle resembles as a “fish” in a fish pool. The swarm of fishes are consisting of N number of fishes which are moving around a D-dimensional place. The procedure will begin through choosing a random number of fishes -which will be called particles thereafter- then the algorithm will look for optimal set of particles via updating the initial chosen set. In general, every particle will use its own knowledge consisting of values met through swarm in the space which ultimately will lead to find the best set of particles. If we consider a D-dimensional space, the exact place of each particle will be shown by $x_i = (x_{i1}, x_{i2}, ..., x_{iD})$. In the same way, for each particle their velocity can be written as a vector like $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. The limitation for the position and velocity of particles represent as $[X_{min}, X_{max}]_D$ and $[V_{min}, V_{max}]_D$, respectively. The value called pbest will present the best position which have met by the i^{th} particle previously $p_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$. The ultimate best value from all of the positions have been met so far will represent by gbest value as $g = (g_1, g_2, \ldots, g_D)$. In every iteration, the pbest and gbest value for position and velocity of i^{th} particle will update in the swarm.

However, there are optimization problems which happen in a discrete space and they need qualitative distinctions not only between variables but also between levels of variables. Due to this requirement, binary particle swarm optimization (BPSO) introduced which is capable to be applied in discrete binary variables. In the binary concept of PSO, every particle can swarm in various positions which indicate various number of bits, therefore overall velocity of a specific particle will define according to the number of bits have been changed in every iteration (Nguyen, Xue et al. 2017). Thus, updating the particles will follow the bellow Eq (2.1):

$$v_{id}^{new} = w \times v_{id}^{old} + c_1 \times r_1 \times (pbest_{id} - x_{id}^{old}) + c_2 \times r_2 \times (gbest_d - x_{id}^{old})$$

$$\text{If } v_{id}^{new}\ (v_{min}, v_{max})\ then\ v_{id}^{new} = max\ (min\ (v_{max}, v_{id}^{new}), v_{min})$$

$$S(v_{id}^{new}) = \frac{I}{1+e^{-v_{id}^{new}}} \tag{2.1}$$

$$\text{If } (\, r_3 < S(v_{id}^{new}))\ then\ x_{id}^{new} = 1\ else\ x_{id}^{new} = 0$$

Where w represents inertia weight to control the effect of particle previous and new velocity, the variables r1 r2 and r3 represent any random numbers between (0, 1), and constant numbers as c1 and c2 represent the acceleration. It means these constant numbers will control the distance that a particle can travel far in every iteration. The variables v_{id}^{new} and v_{id}^{old} indicate the velocities of the new and old particle, respectively. In the same way, variables x_{id}^{new} and x_{id}^{old} indicate the new and old position of the particle. The variables v_{max} and v_{min} indicate the maximum and minimum velocity which will be defined by user (for example v_{max} = 6, v_{min} = -6). The function S(v_{id}^{new}) will update the position of particles (CrammerK 2000). In the case the function S(v_{id}^{new}) indicates a value bigger that r3, means the position will be selected for next iteration and the value of the position will assign as $\{1\}$. In contrast in the case the value of the function S(v_{id}^{new}) is lesser than r3, then this position will not be chosen for next iteration and its value will assign as $\{0\}$ (Chuang, Hsiao et al. 2011, Mandal and Mukhopadhyay 2012, Mukhopadhyay and Mandal 2014).

2.5 MEMBRANE COMPUTING IN OPTIMIZATION PROBLEMS

Many membrane algorithms have been proposed for solving various optimization problems, such as knapsack problem (Tan et al., 2004), point set matching problem (Chen, Zhong et al. 2016), numerical optimization problem (Y. Wang et al., 2004), multi-objective optimization problem (Symons and Nieselt 2006, Lee, Lin et al. 2011), DNA sequence design problem (Holland 1992), and many practical problems (Goldberg 1989, Liu, Cutler et al. 2005). On the other hand, many kinds of NP-problems have been addressed on membrane computing through P systems. The first solutions to NP-complete problems in membrane computing were designed in the cell-like model called P systems with active membranes (Díaz-Pernil, Pérez-Jiménez et al. 2008) and after that problems such as the Satisfiability problem, several numerical problems (Subset Sum, Knapsack, Partition, etc.), and also graph problems (3-Coloring, Clique, Vertex Cover, etc.). Moreover, Some NP-complete problems have been efficiently solved with tissue-like P systems: SAT (Păun, Pérez Jiménez et al. 2004), 3-coloring (Díaz-Pernil, Gutiérrez-Naranjo et al. 2007), Subset Sum (Díaz-Pernil, Gutiérrez-Naranjo et al. 2007) and a uniform linear-time solution to the Vertex Cover problem(Díaz-Pernil, Pérez-Jiménez et al. 2008).

2.6 MEMBRANE-INSPIRED EVOLUTIONARY ALGORITHM (MIEA)

Based"on the interaction on membrane computing and evolutionary computation, the field of membrane-inspired evolutionary algorithms (MIEAs) have been introduced in the study (G. Zhang, M. Gheorghe, et al., 2014). After that, more studies have been done to mix the evolutionary algorithms and P systems (e.g., (Nishida, 2004; G. Zhang et al., 2013; G. Zhang, H. Rong, et al., 2014; (Zhang, Pérez-Jiménez et al. 2017)). Some other studies also focused on hybridization of P system with optimization algorithms: for example, (Singh and Deep 2014) combined P system with particle swarm optimization for the aim of minimizing nonlinear optimization problem, (Du, Xiang et al. 2014) combined P system with particle swarm optimization for the objective of enhancing accuracy of particle swarm optimization as well as overcoming the premature convergence, and (F. Zhou et al., 2010) proposed a particle swarm optimization P system, the so-called PSOPS, and examined the model on seven-bench function optimization problem and concluded that effectiveness of method improved compared to PSO.” Moreover, there are other combinations of PSO with membrane computing, for example, (Zhang, Zhou et al. 2012, Zhang, Rong et al. 2014, Cheng, Zhang et al. 2015, Zhang, Pérez-Jiménez et al. 2017). A review on membrane inspired evolutionary algorithms collected in Table 2.1.

Table 2.1 Membrane-inspired evolutionary algorithms (MIEAs)

Author (year)	Method	Description
(Zhang, Rong et al. 2014)	optimization spiking neural P system (OSNPS).	an“extended spiking neural P system (ESNPS) has been proposed by probabilistic selection of evolution rules and multi-neurons output, called optimization spiking neural P system (OSNPS). approximately solve combinatorial optimization problems. experiments on knapsack problems have been reported”
(Zhang, Gheorghe et al. 2014)	overview“of the evolutionary membrane computing state-of-the-art”	survey“their theoretical developments and applications, sketch the differences between them, and compare the advantages and limitations.”
(Nishida 2004)	Membrane inspired algorithm for optimization problems to solves the traveling salesman problem	Sub“algorithms improve tentative solutions simultaneously. Then the best and worst solutions in a region are sent to adjacent inner and outer regions, respectively. By repeating this process, a good solution will appear in the innermost region.”
(Zhang, Cheng et al. 2013)	presents a hybrid Differential Evolution algorithms and Tissue P Systems (DETPS), used for solving a class of constrained manufacturing parameter optimization problems.	DETPS“uses a network membrane structure, evolution and communication rules like in a tissue P system to specify five widely used DE variants respectively put inside five cells of the tissue membrane system”
(Singh and Deep 2014)	Proposed a broad framework for the hybridization of P-	with“a view to minimize nonlinear optimization problems. concludes that a

To be continued…

…continuation

	system and Particle Swarm Optimization	lot of scope of research is possible in this domain.”
(Du, Xiang et al. 2014)	Proposed P system based particle swarm optimization (P-PSO)	to overcome the premature convergence and improve the precision of the algorithm. (P-PSO) keep the balance of global search and partial optimization. has high accuracy. can effectively improve the performance of the original PSO algorithm.”
(Zhou, Zhang et al. 2010)	Proposed PSOPS, based on particle swarm optimization and P systems.	Employing“the formal framework and communication rules of membrane computing and the evolutionary principles of PSO in each membrane. Its effectiveness and validity are verified by function optimization.”
(Wang, Zhang et al. 2015)	Proposed a modified membrane inspired algorithm based on particle swarm optimization (mMPSO)	a“smoothness algorithm is proposed to remove the redundant information of a feasible path; inspired by the idea of tightening the fishing line, a moving direction adjustment for each node of a path”
(Cheng, Zhang et al. 2015)	Proposed a membrane-inspired evolutionary algorithm based on population P systems and differential evolution for multi-objective optimization.	In“the algorithm, the cells of population P systems are divided into two groups. The first group, consisting of most of cells, focuses on evolving objects using differential evolution rules while the second group, consisting of only one cell, aims at selecting and re-distributing objects across the first group of cells for next generation using a special selection rule.”
(Zhang, Zhou et al. 2012)	HPSOPS, is proposed by appropriately combining membrane systems and a hybrid particle swarm optimization with wavelet mutation (HPSOWM).	HPSOPS“is designed with the hierarchical membrane structure and transformation/communication-like rules of membrane systems, the representation of individuals and the evolutionary mechanism of HPSOWM. Experimental results from various broadcasting problems show that HPSOPS performs better than its counterpart HPSOWM and genetic algorithms”

2.7 FEATURE SELECTION BASICS AND METHODS

Analysis of microarray data for thousands of genes in arrays of abnormal and normal cells is an effective approach for investigating gene expression in cancer. Microarray data trials involve small samples (as small as a few dozen) and gene expression of high dimensionality (as high as a few thousand). A very high number of genes are found to be irrelevant for analysis, which may hinder correct prediction (Li and Yang 2002, Nguyen and Rocke 2002, Tan, Shi et al. 2004, Li 2006).

Many machine learning techniques have been developed for computational analyses of microarray data. These techniques are used to extract patterns and build classification models for gene expression data and have significantly aided in cancer prediction (Rocha, Mendes et al. 2007) and prognosis (Vanneschi, Farinaccio et al. 2011)."

In the literature (Alelyani, Tang et al. 2013), from one point of view, methods of feature selection for classification can be divided into three families 1) methods for flat features, 2) methods for structured features and 3) methods for streaming features as demonstrated in Figure 2.2. In streaming features, unlike flat features and structured features that all features are known in advance, the candidate features are generated dynamically and the size of features are unknown. Thus, based on the target of study which is making pattern from known genes, either flat or structured methods are mostly focused.

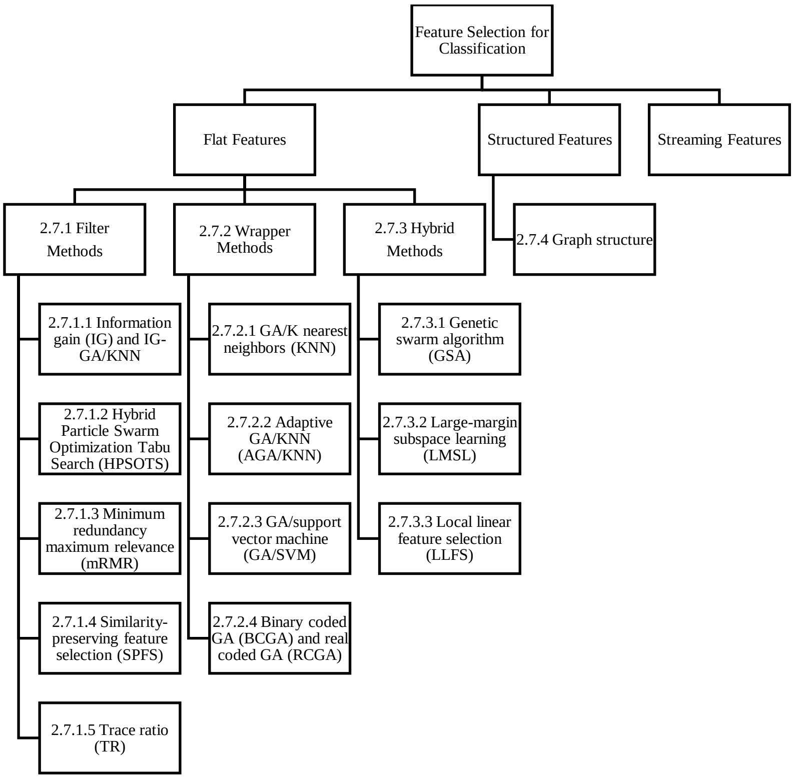

Feature Selection for Classification
Flat Features
Structured Features
Streaming Features
2.7.1 Filter Methods
2.7.2 Wrapper Methods
2.7.3 Hybrid Methods
2.7.4 Graph structure
2.7.1.1 Information gain (IG) and IG-GA/KNN
2.7.1.2 Hybrid Particle Swarm Optimization Tabu Search (HPSOTS)
2.7.1.3 Minimum redundancy maximum relevance (mRMR)
2.7.1.4 Similarity-preserving feature selection (SPFS)
2.7.1.5 Trace ratio (TR)
2.7.2.1 GA/K nearest neighbors (KNN)
2.7.2.2 Adaptive GA/KNN (AGA/KNN)
2.7.2.3 GA/support vector machine (GA/SVM)
2.7.2.4 Binary coded GA (BCGA) and real coded GA (RCGA)
2.7.3.1 Genetic swarm algorithm (GSA)
2.7.3.2 Large-margin subspace learning (LMSL)
2.7.3.3 Local linear feature selection (LLFS)

Figure 2.2 Methods of feature selection for classification

2.7.1 Filter Methods

In the context of filter method, the common practice is to investigate the relevance of every gene as an individual entity to the predefined class such as normal gene class or cancerous gene class. In this approach, all of the genes will be categorized according to their relevance value. The most practiced filter methods are ReliefF (Ang, Mirzal et al. 2016), χ2 (Tan et al., 2004), Information Gain (Wang, Wang et al. 2017). Through filter approach, any selected gene will be considered as significant gene without the practice of sending the selected genes to any classifier. It means significant genes will be chosen independent from classifier and that is the reason filter methods are efficient in terms of computation cost, although the main drawback of these method such as mRMR and RFS is that they do not interact with the classifier.

a. Information Gain (IG) and Information Gain- GA/KNN (IG-GA/KNN)

Information gain (IG) is a popular feature selection method used to rank genes in a data set according to their significance. According to (Hira and Gillies 2015), IG is a univariate filter method. Univariate IG ranking approximates the conditional distribution ($C \mid F$), where C is the class label and F is the feature vector. IG is used as a surrogate for the conditional distribution.

Another popular method is based on t-score feature selection. Markov blanket filtering based on t-scores is categorized as a multivariate filter method (Hira and Gillies 2015). Multivariate Markov blanket filtering finds features that are independent of the class label so that their removal will not affect the accuracy. In multivariate methods, paired t-scores are used to evaluate gene pairs depending on how well they can separate two classes; the aim is to identify genes that work together to provide a better classification (Bø and Jonassen 2002).

Multivariate methods are able to find relationships among the features, while univariate methods consider each feature separately. We now describe examples of these filter methods: Information Gain- GA/KNN (IG-GA/KNN).

IG is a feature ranking technique based on decision trees and has good classification performance (Martín-Valdivia, Díaz-Galiano et al. 2008). The principle behind IG is to choose features that present information about classes. According to (Mukras, Wiratunga et al. 2007), these features are discriminative in nature and occur within a single class. In the initial stage, a subset of the original feature set is usually acquired by implementing IG in the form of filtering criteria. This is usually performed by categorizing the genes. Genes with an information value that exceeds the threshold are eligible to enter the next stage. In the second stage, the GA is applied to the set of filtered genes. The IG-GA/KNN method uses a KNN classifier to check the cross-validation accuracy.

b. Hybrid Particle Swarm Optimization Tabu Search (HPSOTS)

HPSOTS is a hybrid classification model comprising particle swarm optimization (PSO) and tabu search (TS) (Shen, Shi et al. 2008). Shen et al. (2004) proposed a modified discrete PSO with the information-sharing mechanism of PSO. Before the heuristic search procedure, genes with lower absolute t-test values among normal and tumor samples are removed. A heuristic search method (HPSOTS, pure TS and PSO) is then applied to the data set.

TS can provide solutions for various difficult optimization problems (Glover, 1986). TS is an iterative process in which the fitness function of a random solution is first evaluated. Then, by tracing the current suggested solution, all neighbors of this solution are identified and evaluated. The tracing is based on a primitive transformation. A new current solution is selected if TS does not identify best neighbors for the current solution. The best neighbor identified is constantly compared with the current solution; if it is worse than the new one, TS tracing is continued upwards. Using this approach, local minima can be easily overcome (Shen, Shi et al. 2008).

c. Minimum Redundancy Maximum Relevance (mRMR)

mRMR is a two-stage feature selection algorithm that is based on a specific formula (Peng, Long et al. 2005). In the first step, a contender feature set is allocated with the feature selection method, specifically mRMR feature selection method. Afterwards, these schemes are used in order to select compact subsets from nominated sets.

d. Similarity-Preserving Feature Selection (SPFS)

SPFS is used for samples with redundant features. The process starts with a conventional combinatorial optimization formula, $K\text{-}\ X_A X_A^T\ =\ K\text{-}\ \sum_{i \in A} f_i f_i^T$, where $X_A = (f_{i1},\ldots,f_{ik}),\ ip\ \epsilon\ A,\ p=1,\ldots,k$. Feature selection in the SPFS framework can be established as a multiple-output regression problem (Zhao, Fung-Leung et al. 2014).

e. Trace Ratio (TR)

TR is an iterative algorithm developed to identify the optimal subset of features for which the subset level score is maximized (Nie, Xiang et al. 2008).

2.7.2 Wrapper Methods

In wrapper methods (Lee, Lin et al. 2011) a classification method will be mixed with feature selection part to select the ultimate genes as significant genes from a set of genes have been already suggested by feature selection. There are two practices in wrapper methods. In the first practice, it starts from a single gene and then other genes gradually will be added to the set of genes one by one to investigate which set of genes produce a better accuracy. In the second practice, it starts from whole set of genes and genes will be dropped one by one gradually to investigate which sets leads to more accurate result. So, clearly wrapper methods are computationally more expensive than the filtering methods. Moreover, most probably the selected genes as significant genes are only relevant to the classifier has been used, therefor overfitting risk increases in wrapper methods. The bottom line regarding filter and wrapper method is that, although filter methods consume lesser time they don’t cooperate with classifiers which leads to worse performance measures. In contrast, wrapper methods end up with more expensive time cost but they cooperate with classifier to produce better performance measures which is important in high density of dataset like microarray datasets.

According to (Hira and Gillies 2015, Brankovic, Falsone et al. 2017), wrappers can be divided into two categories: randomized and deterministic strategies. Although they have a higher risk of overfitting in comparison to deterministic methods, randomized wrappers are less prone to local optima and are computationally intensive. Randomized wrapper methods, which mostly use genetic algorithms (GAs), are more prevalent in feature selection for microarray cancer data.

The GA was initially developed by (Holland 1975). According to (Li, Weinberg et al. 2001), the GA imitates natural selection. In addition, this algorithm helps in identifying optimal solutions by mimicking evolution in biological systems (D. Liu et al., 2004). The performance of wrapper methods such as the GA has reviewed as follow for feature selection in combination with different classifiers.

a. GA/K Nearest Neighbors (KNN)

The GA/KNN approach combines a GA feature selection method and a KNN classifier to identify genes that can discriminate between different sample classes such as tumor and normal tissue (Li, Weinberg et al. 2001, Alarcón-Paredes, Alonso et al. 2017). The GA feature selection method can identify small subsets of genes for training, and then applies an evolutionary tool. (Srinivas and Patnaik 1994) demonstrated that the GA/KNN method can identify the existence of different subtypes within classes. The method can also be used in computationally intensive searches for many chromosomes (near-optimal solutions), for which approximately 10,000 near-optimal solutions are usually needed for a typical run. The parameters are first loaded and then the initial generation is produced using approximately 200 chromosomes. Each chromosome has 30 genes that are picked randomly from the gene pool. After a preselection step to enhance the probability of identifying the most optimal chromosomes, the top 100 chromosomes are chosen to create the initial generation. In the next stage, the program goes through mutation and crossover operations. If optimal chromosomes are found, they are recorded and placed into a panel of discriminated genes; mutation and crossover operations are run for five more loops before beginning a new iteration with a new initial generation. The next generation is subjected to mutation and crossover operations up to the maximum number of iterations. This entire process is continued until the recognized number of optimal chromosomes is produced.

The principle underlying KNN classifiers is supervised learning (Mitchell 1997). A classifier initially searches for the nearest k samples when a new sample appears in the pre-existing training data. These new samples are classified according to the most similar class.

b. Adaptive GA/KNN (AGA/KNN)

A GA solely finds the nearest optimal solution, while in each run of high-dimensional data, nearest optimal string is not similar. To solve this problem, AGA was developed by adding three techniques, which were comprised of immigration and extinction strategy, adaptive possibilities of mutation and crossover, and elitist strategy. Afterwards, it was combined with KNN. The reason for the combination of AGA and KNN is: some assumptions are needed for most feature selection methods while to use in high-dimensional scopes they are not appropriate. AGA is an appropriate search tool for analyzing high-dimensional and noisy data, since it follows biological principles in searching for near-optimal strings. KNN is a simple and effective classifier among trialed and implemented classification techniques.

The examining AGA/KNN in different areas indicates satisfactory result (Li, Weinberg et al. 2001). (Lee, Lin et al. 2011) were the first to use a combination of KNN and AGA in five different stages: termination, genetic operators (this element incorporates the adaptive probabilities and selection of mutation and crossover), fitness function, initial population, and encoding.

c. GA/support Vector Machine (GA/SVM)

GA/SVM is found to be the genetic algorithm, which is, based on the SVM classification utilized to identify or recognize the optimal parameters for a conventional SVM classifier (Chen and Yang 2012, Phan, Le Nguyen et al. 2017).

Studies show that SVM is one of the most powerful and integrated learning classifiers, which offers great opportunity to have an effective pattern recognition approach. Initially, the paradigm of SVM classifier was identified by various researchers and practitioners (Huang and Wang 2006). It has been observed that SVM utilizes a linear separating plane, which is referred as the hyper-plane. One of the major objectives of this plane is to increase or maximize the distance amid two classes (Pierna, Baeten et al. 2004).

(Huang and Wang 2006) reported that four common kernel functions are used as SVM classifiers: sigmoid, radial basic function, polynomial, and linear functions. The kernel parameters for these functions need to be set properly to maximize the SVM classification accuracy (Huang and Wang 2006).

d. Binary Coded GA (BCGA) and Real Coded GA (RCGA)

A BCGA is a probabilistic search algorithm that transforms a population set (mathematical objects with a uniform length) to a new offspring according to the Darwinian principle of natural selection. In particular, a set of chromosomes that individually represent one probable solution is modified and transformed via genetic processes to create a new population. The entire process continues up to a predetermined number of iterations or until further enhancement and improvements are achieved. According to a mutation operator, a given chromosome or a gene is selected in a random manner and its value is exchanged (e.g., 1 for 0, and vice versa; (Holland 1975).

RCGA operates on a population set that represents a variable of the problem, and the chromosome size is kept the same as the length of the problem solution. The starting point is the initial population and then the main loop algorithm is run. The main loop comprises preprocessing, three genetic operations, and post-processing, and it continues until the termination condition is satisfied. The process includes a fitness function test to avoid premature junctions in the initial stages of the evolution process and to stimulate convergence in the more advanced stages of the process. In addition, genetic operators incorporate mutation, crossover, and selection.

2.7.3 Hybrid Models

A middle solution for researchers can be the use of hybrid methods that are usually a mix of two or more feature selection methods from different origins which use the core of the classifier to establish a criterion to rank features. But it is worth noting that the embedded method such as SVM-RFE, in spite of the fact that it is, in theory, better than the filter methods, achieves comparable or even worse results than them in terms of classification accuracy. In the study of (Bolón-Canedo, Sánchez-Marono et al. 2014), ReliefF method even when combined with the SVM classifier (SVM-RFE) it could not obtain the highest accuracy against other methods, in contrary with what was expected (in average 0.79 which is the same result of CFS, FCBF and even worse than pure ReliefF, IG, mRMR with the avg-accuracy 0.81, 0.86 and 0.80, respectively). Thus, it is not possible to generalize that all embedded methods can overcome with aforementioned minus points in filter and wrapper methods.”

In hybrid models, similar to wrapper methods, feature selection is linked to the classification stage, but this link is much stronger. Embedded methods offer the same advantages as wrapper methods concerning the interaction between feature selection and classification. Moreover, they have better computational complexity since feature selection is directly included in the classifier construction during training. The genetic swarm algorithm (GSA), large-margin subspace learning (LMSL), and local linear feature selection (LLFS) are examples of embedded methods.

a. Genetic Swarm Algorithm (GSA)

GSA combines the strengths of the GA and PSO (Kumar, Victoire et al. 2012). The GSA design is based on a fuzzy expert system for classification of microarray data. PSO is a population-based algorithm that uses swarm intelligence to solve optimization problems. Each individual in a population is referred to as a particle and designates a solution. Moreover, each particle flies with an adaptable speed in the search space according to its own experience or that of other particles. Thus, each particle tries to progress by imitating the traits of other particles traits. This is possible because each particle has memory to hold positions in the search space that are met. The best position is denoted pbest and the best particle in the population is denoted gbest (Tse and Tso 1993).

b. Large-Margin Subspace Learning (LMSL)

LMSL is a subspace learning algorithm that is based on a large-margin framework (B. Liu et al., 2013). Initially, it uses the nearest neighbor along with the same label and different labels for a given sample.

c. Local Linear Feature Selection (LLFS)

(Sun, Todorovic et al. 2010) proposed LLFS, which is based on well-structured numerical and machine learning analysis techniques, without making any assumptions regarding the underlying allocation of data. LLFS can process a wide range of features within minutes on a personal computer, while achieving quite high reliability and integrity that is approximately insensitive to an increasing number of irrelevant features. Table 2.2 shows the methodology used by various models for analysis of microarray data and Table 2.3 explain the achieved accuracy.

Table 2.2 Methodology for analysis of microarray data

Model	Type	Reference	Methodology
LMSL/SVM RFS/SVM LLFS/SVM SPFS/SVM	Embedded Embedded Embedded Filter	(B. Liu et al., 2013)	In three algorithms including (LMSL, LLFS, RFS) fivefold cross validation used to determine regularization parameter. Samples randomly selected 70%-30% as training and test data, respectively. Finally, the highlighted features are classified by linear SVM to

mRMR/SVM TR/SVM	Filter Filter		determine accuracy of classification.
t-test/HPSOTS	Filter	(Shen, Shi et al. 2008)	Before a heuristic search, genes of top-ranked are selected by t-test filtering algorithm. In terms of datasets, 50 colon samples are randomly selected as training datasets from 62 samples, and 12 samples are used as test dataset. The result of classification shows 83.87% accuracy by fivefold cross validation. In the case of leukemia 38 samples are selected as training dataset (27 ALL, 11 AML) and 24 samples (20 ALL, 14 AML) are selected as test data. The result of classification shows 90.28% accuracy by fivefold cross validation. In terms of breast cancer, from a total of 49 breast tumor samples, 40 samples are selected as training set and the rest nine samples are selected as test dataset. The result of classification shows 85.71% accuracy by fivefold cross validation.
GA/SVM	Wrapper	(Chen and Yang 2012)	In"this study Four different methods including: all genes (All), 70 correlation-selected genes (C70), 15 medical literature-selected genes (R15), and 50 t-test-selected genes (T50) are used for gene selection. The results of classification accuracy indicate 95% for T50 and 90% for C70 or R15."
AGA/KNN GA/KNN	Wrapper Wrapper	(Lee, Lin et al. 2011)	The study used three groups of genes including; group 1, 50 genes which are selected by AGA/KNN, group 2, 50 genes which have the smallest max-T adjusted p value and group 3 with 50 genes are selected randomly. First dataset is colon data with 62 samples which 40 are tumor and 22 are normal genes. From 62 samples, 40 samples selected as training set and the remaining are test datasets. Second data set is small, round blue cell tumors (SRBCTs) with 2308 genes are divided to 63 training samples from 23 tumors.
GSA BCGA RCGA	Embedded Wrapper Wrapper	(Kumar, Victoire et al. 2012)	For gene selection from the original gene profile, mutual information technique is used followed by fuzzy expert system for classification. Fuzzy system is including if-then rules using GA and membership functions evolving by PSO. Standard leave-one-out cross-validation (LOOCV) determine generalizability of the proposed system. Data sets including colon cancer, leukemia, and lymphoma are considered in simulations.
IG-GA/KNN IG/KNN GA/KNN	Filter Filter Wrapper	(Yang, Chuang et al. 2010)	First, information gain used for feature selection, second GA as a random wrapper method followed by KNN classifier. Standard leave-one-out cross-validation (LOOCV) determine generalizability and accuracy of the proposed system.
NRGA IG/KNN IG-GA/KNN IG-NRGA/KNN	Filter Filter Filter Filter	(Sungheetha and Suganthi 2013)	First, information gain and genetic algorithm used for pre-feature selection of microarray data, then, non-dominated ranked GA (NRGA) is used as actual feature selection and KNN used to evaluate the NRGA algorithm. The details of datasets are as follow:

To be continued…

…continuation

	Brain tumor: five human brain tumor types, 90 samples, 5920 genes
	Lung cancer: five lung cancer types and normal tissues, 203 samples, 1260 genes
	Prostate tumor: two prostate tumors and normal tissues, 102 samples, 10509 genes

Table 2.3 Accuracy of algorithms for cancer data sets

Author/s	Method	Proposed algorithm	Comparator algorithm/s	Classification accuracy (%)		Data set
				Proposed algorithm	Comparator algorithm/s	
(Yang, Chuang et al. 2010)	Statistics, machine learning	IG-GA/KNN	IG/KNN GA/KNN	93.33	88.89 92.22	Brain tumor
				100.00	93.06 97.22	Leukemia
				95.57	90.15 94.09	Lung cancer
				96.08	89.22 91.18	Prostate tumor
(Lee, Lin et al. 2011)	Machine learning	AGA/KNN	GA/KNN	~90% after 40 runs, increasing to ~100% after 70 runs	~80% when >1000 runs were executed	Pediatric SRBCTs
(Chen and Yang 2012)	Machine learning	GASVM	+SVM +Correlation-based method +Decision tree +Nearest-centroid with multiple random validation +Bayesian network +ANN +Nearest neighbors	90	60 83 89.47 69 74 78.65 76.34	Breast cancer
(Kumar, Victoire et al. 2012)	Fuzzy expert system	GSA	BCGA RCGA PSO	58.7	56.5 52.8 51.2	Colon cancer
				81.2	79.1	Leukemia

To be continued…

…continuation

					75.5 76.3	
				69.5	65.2 66.7 68.9	Lymphoma
(Sungheetha and Suganthi 2013)	Machine learning	NRGA/KNN	IG/KNN IG-GA/ KNN	89.1	70 73.4	Brain Tumor
				77.4	70.15 74.8	Lung cancer
				86.3	82.22 77.6	Prostate tumor
(Shen, Shi et al. 2008)	Machine learning	HPSOTS	Pure TS Pure PSO	93.55	90.32 90.33	Colon
(B. Liu et al., 2013)	Machine learning	LMSL	RFS LLFS SPFS mRMR TR	95.61	95.10 93.10 94.73 94.52 95.03	Lung
				92.31	91.31 91.46 77.93 79.79 81.51	Prostate

(Yang, Chuang et al. 2010) presented an IG-GA/KNN framework combining a wrapper method (GA) and filter method (IG) for feature selection among microarray data sets (Table 2). They used IG to choose significant gene subsets from all elements in the gene expression data, and applied a GA for selection of actual features. The KNN method with LOOCV was applied to evaluate IG-GA. Using a KNN classifier, the accuracy for a brain cancer data set was 93.33% for IG-GA feature selection, compared to 88.89% for IG and 92.22% for GA. For a leukemia data set the accuracy was 100% for IG-GA, 93.06% for IG, and 97.22% for GA. For a lung cancer data set the accuracy was 95.57% for IG-GA, compared to 90.15% for IG and 94.09% for GA. For a prostate cancer data set, IG-GA had 96.08% accuracy, compared to 89.22% for IG and 91.18% for GA.

(Lee, Lin et al. 2011) found that a KNN classifier with AGA feature selection can reduce the dimensionality of a data set. For pediatric small, round, blue-cell tumors (SRBCTs), all test samples were categorized correctly after 70 runs, while with GA feature selection accuracy reached just 80% after 1000 runs.

(Chen and Yang 2012) applied the GASVM model to data for 97 patients with breast cancer. They used four different gene selection strategies: all genes, 70 correlation-selected genes, 15 medical literature-selected genes, and 50 t-test-selected genes. GA feature selection improved the SVM classification accuracy (90%) in comparison to SVM (60%), correlation (83%), decision tree (89.47%), nearest-centroid with multiple random validation (69%), Bayesian network (74%), ANN (78.65%), and nearest neighbor (76.34%) methods.

(Kumar, Victoire et al. 2012) compared the performance of GSA, PSO, RCGA, and BCGA. For a colon cancer data set, GSA (including GA feature selection and PSO) achieved 58.7% accuracy, compared to 56.5% for BCGA, 52.8% for RCGA, and 51.2% for PSO. For leukemia classification, GSA yielded 81.2% accuracy, compared to 79.1% for BCGA, 75.5% for RCGA, and 76.3% for PSO. For a lymphoma data set, GSA achieved 69.5% accuracy, compared to 65.2% for BCGA, 66.7% for RCGA, and 68.9% for PSO. The simulation results show that GSA generates a compact and integrated fuzzy system with higher levels of accuracy for all the data sets compared to the other approaches.

A hybrid NRGA/KNN proposed by (Sungheetha and Suganthi 2013) incorporates IG GA for feature selection in microarray data sets. IG is used to choose significant gene subsets from all elements in the gene expression data, whereas NRGA is applied for selection of actual features. The KNN method is utilized to examine the NRGA algorithm. Using the KNN classifier for a brain cancer data set, NRGA feature selection achieved 89.1% accuracy, compared to 70% for IG and 73.4% for IG-GA. For lung cancer classification, NRGA yielded 77.4% accuracy, compared to 70.15% for IG and 74.8% for IG-GA. For a prostate cancer data set, NRGA achieved 86.3% accuracy, compared to 82.22% for IG and 77.6% for IG-GA. The experimental results indicate that NRGA/KNN effectively simplifies the number of gene expression levels and provides more accurate and reliable classification.

(Shen, Shi et al. 2008) compared the performance of HPSOTS to that of pure PSO and TS algorithms. For a colon cancer data set, t-test feature selection with the HPSOTS classifier achieved 93.55% accuracy compared to 90.32% for a t-test with pure TS and 90.33% for a t-test with pure PSO.

(B. Liu et al., 2013) performed a wide range of experiments to evaluate the LMSL efficiency in comparison to five characteristic feature selection algorithms. Using SVM as the classifier for a lung cancer data set, LMSL for feature selection yielded better accuracy (95.61%) than RFS (95.10%), LLFS (93.10%), SPFS (94.73%), mRMR (94.52%), and TR (95.03%). LMSL also achieved better accuracy (92.31%) than RFS (91.31%), LLFS (91.46%), SPFS (77.93%), mRMR (79.79%), and TR (81.51%) for a prostate cancer data set.

The RFS and LLFS algorithms are closely associated with LMSL: the principle of large margins underlies both LLFS and LMSL, and LMSL benefits from RFS for effective resolution of objective functions (He, Cai et al. 2005). SPFS, mRMR, and TR are state-of-the-art feature selection algorithms with different effective characteristics. mRMR removes redundant elements by considering them in a pairwise manner. TR characterizes data set structures via a Laplacian graph and has considerably better performance than similar algorithms such as Laplacian Score. The SVM classifier showed accuracy of 30%, which is much lower than the accuracy of TR, RFS, and LMSL. After RFS, LMSL is the next fastest method. Moreover, LMSL takes more time than RFS in PROS (0.06 s). LLFS is considerably slower than LMSL for all three data sets. LMSL is slower than RFS because it requires calculation of the sample margins for improved and integrated feature selection and better performance.

2.7.4 Graph Structure

Graph is another method of feature selection for classification (Wang, Zhang et al. 2017). We often have knowledge about pair-wise dependencies between features in many real-world applications (McAdams and Arkin 1997). Many biological studies have suggested that genes tend to work in groups according to their biological functions, and there are some regulatory relationships between genes (Alelyani, Tang et al. 2013). In these cases, features form an undirected graph, where the nodes represent the features, and the edges imply the relationships between features. Several recent studies have shown that the estimation accuracy can be improved using dependency information encoded as a graph (Alelyani, Tang et al. 2013). For example, the study of (Mandal and Mukhopadhyay 2014) proposed a graph based multi objective particle swarm optimization (PSO)-based algorithm that optimizes average node-weight and average edge weight of the candidate sub graph simultaneously. The proposed algorithm is applied for identifying relevant and non-redundant disease-related genes from microarray gene expression data. In the study of (Mandal and Mukhopadhyay 2014) graph based MObPSO got better performance in compared with SBE, CFS, mRMR, SFS in terms of classification accuracy. For example, the result of 10-fold cross-validation in prostate cancer indicates it got 94.12 classification accuracy in compared with SBE, CFS, mRMR, SFS (88.63, 95.1, 93.51, 91), respectively."

From another perspective for the division of feature selection and classification methods, considerable number of hybrid intelligent optimization algorithms have been developed widely based on biology intelligence. The most popular intelligent methods in feature selection and classification methods are shown in three categories as Figure 2.3. Base on this category (Figure 2.3), wide range of mixed methods are developed mainly based on evolutionary learning methods such as genetic algorithm (GA), neighbourhood search like K nearest neighbor (KNN) and swarm intelligence algorithms such as particle swarm optimization (PSO).

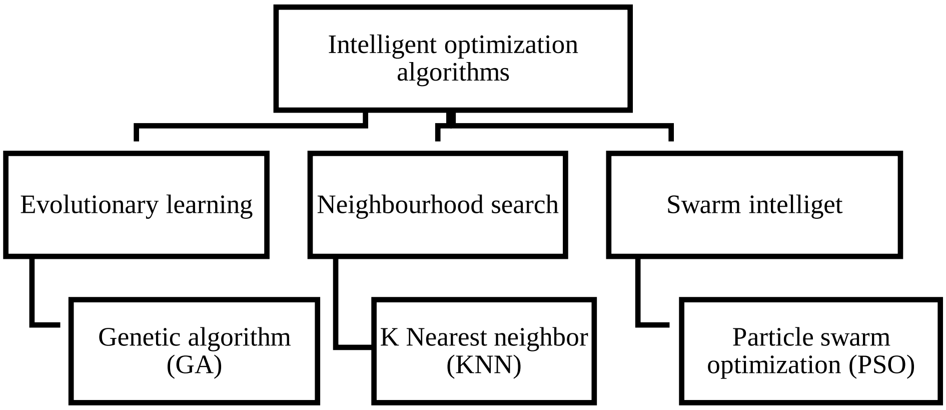


Figure 2.3 Methods of intelligent optimization feature selection for classification

Although the mix of intelligent optimization methods with other methods has brought advantages, many limitations are remained unsolved. Based on our review, these problems can be categorized to three parts. First; e.g., pure genetic algorithm generally has limitations such as 1) slow convergence, 2) lacks of rank based fitness function and 3) being a time-consuming approach. Mixed methods of GA were not capable to tackle with these problems completely. Mixed of GA with IG and KNN, IG-GA/KNN (Yang, Chuang et al. 2010) faces with the increased time complexity; AGA/KNN (Lee, Lin et al. 2011) fell into local optima or could not converge. Also, it faces with the limitation of the fitness value. BCGA (Kumar, Victoire et al. 2012), has slow convergence because exchanging a number from real value to binary and vice versa increase the computation time substantially. BCGA takes more time for decoding and shows poor accuracy in classification than RCGA and PSO. In RCGA (Kumar, Victoire et al. 2012), mediocre scalability is the main disadvantages as the number of parameters to be established grows quadratically with the size of the FCM model (number of nodes). This is because the genetic optimization applied to this modelling is time consuming especially when dealing with large number of variables. Mix of GA and SVM, GASVM (Chen and Yang 2012) faces with the high risk of over-fitting problems which it is happened because the number of genes greatly exceeds the number of samples.

Second; in terms of classification accuracy, resulted accuracy in intelligent feature selection and classification algorithms varies greatly either in different types of cancer or different datasets, also these methods are still unstable for high dimensional data, especially when the number of samples is relatively small,”overfitting problem (Nakatsu 2017) occurs. For example, based on our review regarding accuracy of some intelligent methods, we can conclude that, among the hybrid models which are applied in brain cancer MC-SVM (with feature selection) and ESVM have got the highest accuracy 100% and 96% respectively. In the leukaemia cancer after ESVM (Huang and Chang 2007) and IG-GA/KNN (Yang et al., 2010) that has gained 100% classification accuracy, MC-SVM (Huang and Chang 2007) (without feature selection) and GA/KNN (Yang et al., 2010) have ranked in the second stage with approximately 97% accuracy. In the case of lung cancer, MC-SVM (Huang and Chang 2007) (with and without feature selection) have raced very nearly at 96% which took second position again after ReliefF+NB (Alonso-González, Moro-Sancho et al. 2012) with 99.63% accuracy. As far as prostate cancer is considered the highest accuracy 100% of ESVM (Huang and Chang 2007) classifier followed by IG-GA/KNN (Yang et al., 2010) at 96%. In the colon datasets, GADP and GA/KNN (Lee and Leu 2011) have hit 100% accuracy. Finally, in the case of breast cancer KFDA and Bayesian+MCMC (Lee and Leu 2011) with 100% accuracy surpassed PSO-SVM at 99%. In sum, regarding average accuracy, ESVM got the highest average (98%) among four types of cancers: brain, lung, leukaemia and prostate in comparison with models such as IG-GA/KNN, MC-SVM (without feature selection) and GA/KNN. E.g., accuracy of GA/KNN was 96.24%, 95.18% and 93.67% for three types of cancers namely: lung, leukaemia and prostate, respectively. Hybridization of PSO model with other methods indicate that, although the accuracy of HPSOTS (Shen, Shi et al. 2008) and MObPSO (Mandal and Mukhopadhyay 2014) have lifted to 93.55% (in colon cancer) and 94.12% (in prostate cancer) respectively in comparison with the accuracy of pure PSO 90.33% (Shen, Shi et al. 2008) (in colon cancer), hybridization of PSO with SVM classifier (Lin, Ying et al. 2008) has risen the accuracy more significantly to 99.18% (in breast cancer). The entire aforementioned accuracy rates may change if the methods apply in other datasets, thus, this comparison cannot be generalized and other reviews can lead to different conclusion.

Third; regards time complexity, despite all the efforts which have been done to decrease execution time of methods on the CPU, it has remained a challenge in all the methods as far as we have reviewed. For example, although execution time in population-based optimization techniques' such as MObPSO is not efficient which takes (81.176 Sec), their time complexity is not higher than other comparative methods (Mandal and Mukhopadhyay 2014). Thus, computational cost is a big challenge for almost all intelligent algorithms which are run on CPU. Recently new attempts have been started to develop parallel feature selection and classification methods such as (Slavik, Zhu et al. 2009) and some efforts are focused on parallelization of intelligent optimization algorithms, such as parallel genetic algorithm on CPUs/computers to identify informative genes for classification (Liu, Iba et al. 2001, Sarkar, Sana et al. 2011), parallel Genetic algorithm on GPU (Li, Wang et al. 2007, Cano, Zafra et al. 2010, Pospichal, Jaros et al. 2010), parallel PSO on GPU (Zhou and Tan 2009, Mussi, Daolio et al. 2011, Kentzoglanakis and Poole 2012, Nobile, Besozzi et al. 2012, Nobile, Besozzi et al. 2013) and parallel processing of microarray data (Guzzi, Agapito et al. 2014). Membrane computing models have parallel structures. Many studies have been focused to simulate membrane computing on single processor (Gutiérrez-Naranjo, Pérez-Jiménez et al. 2006, Garcıa-Quismondo, Gutiérrez-Escudero et al. 2009, Păun, Perez-Jimenez et al. 2010). Clearly, this type of simulation only operates one task per a time unit until completing the assigned task and begin another task, therefor the most important potential of a membrane inspired system which is parallelism missing basically. Recently, remarkable efforts have been put in developing accelerated P systems, e.g., using of multiple CPUs (Peña and Castellanos 1993, Peña Camacho, Bravo García et al. 2011, Mazza, Ballarini et al. 2012), clustering of Computers (Peng, Jin et al. 2016) and using of field programming gateway array (FPGA) (Van Nguyen and Gioiosa 2010). The most important attempts to parallelize membrane computing models are being done via using of graphic processing units (GPUs) (Garcıa–Quismondo and Pérez–Jiménez , Cecilia, García et al. 2009, Cecilia, García et al. 2010, Dematté and Prandi 2010, Cabarle, Adorna et al. 2012, Martínez del Amor, Pérez Carrasco et al. 2013, Zhang, Wang et al. 2014). The other important objective of feature selection is to develop faster and more cost-effective models. In this regard, our proposed membrane-inspired feature selection method should perform with an efficient time in compare to previous methods that we aimed to settle via using potentials of membrane computing in parallel processing and nondeterminism. Based on our best knowledge there is not any parallel membrane-inspired feature selection and classification method.

Regarding classification method for membrane-inspired classifier, many statistic methods proposed such as weighted voting scheme (Golub, Slonim et al. 1999), nearest neighbor classification (Li, Weinberg et al. 2001),“least square and logistic regression (Mehmood and Ahmed 2016) and naive bayes approach (Fan, Poh et al. 2009). These approaches are used to build classifier to extract significant genes. The drawback of statistic-based classifiers is that they are not flexible enough in terms of different samples, in such a way that if any slight change in the expression of gene samples happens they will not be able to classify the genes correctly (Kumar, Victoire et al. 2012).

Regarding classification method for membrane-inspired classifier, many machine learning methods proposed in the previous works, such as Artificial Neural

Networks (Khan, Wei et al. 2001) and Support Vector Machine (Du, Jeng et al. 2016). These methods have been successful to classify microarray data. The drawback of these methods is that, although they have achieved good performance measures such as high accuracy under the examined specific dataset, interpretation of the results is not easy (Kumar, Victoire et al. 2012). These methods are well-known to ‘‘Black Box’’ method which focus on performance measures without providing deep understanding of how their operation matches with biological concept of gene extraction. In the study of (Kumar, Victoire et al. 2012), the performance of GSA (mix of GA and PSO) compared with three different approaches including: particle swarm optimization (PSO), real coded genetic algorithm (RCGA) and binary coded genetic algorithm (BCGA). The result indicates the proposed GSA outperforms the aforementioned approaches.

A common difficulty for all these techniques is the large number of genes (features) compared to the small sample size, which has a negative impact on their speed and accuracy. To overcome this limitation, feature selection techniques are applied to recognize differentially expressed genes from redundant genes and remove irrelevant genes. Feature selection can improve the accuracy and speed of classification systems by reducing dimensionality (Shang and Shen 2005). Some of the improvements are (Wang, An et al. 2015). According to (Nguyen and Rocke 2002), feature selection involves selecting a subset of novel features (i) to investigate the relationships between specific diseases and genes and (ii) to identify a compact set of discriminative genes to develop a pattern classifier with good generalizability and limited complexity.

(Hira and Gillies 2015) reviewed different feature selection and extraction methods and their classification accuracy in terms of the number of genes evaluated. The authors concluded that incorporation of prior knowledge from various biological sources increases the accuracy and reduces the computational complexity. However, their conclusions regarding robust feature selection methods were made without investigating experimental evidence for different methods or different microarray data sets, and without reviewing variations in classification accuracy for different combinations of feature selection methods and classifiers.

In general, three main methods are used for feature selection in microarray data sets: wrapper, filter, and hybrid methods (Langley, 1994). Wrappers, which are general purpose algorithms, search the feature space and then test the performance of subsets according to a learning algorithm. A classifier is required to evaluate the performance quality. Filter methods mostly use a feature ranking function that assigns a relevance score to each feature. A higher rank is allocated to features that are more relevant. Filter methods are independent of classification algorithms. Hybrid methods select features during an implicit process for learning of optimal parameters. As in wrapper methods, feature selection depends on a classification algorithm.

2.8 EMBEDDED METHOD

Embedded techniques e.g., (Mistry, Zhang et al. 2017) vary from other feature determination strategies in the way, feature selection, and classification (learning part) patronize. Filter techniques do not consolidate learning. Wrapper techniques utilize a learning machine to quantify the nature of subsets of features without consolidating information about the particular structure of the classification or regression function, and can along these lines be incorporated with any learning machine. As opposed to filtering and wrapper approaches, in embedded techniques the learning part and the feature selection part cannot be isolated - the structure of the class of functions under advisement assumes an essential role.

Feature selection can be comprehended as finding the feature subset of a specific size that prompts the biggest conceivable generalization or proportionately to the negligible risk. Each subset of features is demonstrated by a vector $\sigma \epsilon \{0,1\}^n$ of indicator variables, $\sigma_i \coloneqq 1$ showing that a feature is available in a subset and $\sigma_i \coloneqq 0$ demonstrating that that feature is missing *(i = 1,…, n)*.

The function G measures the execution of a trained classifier $f^*(\sigma)$ for a given σ. It is essential to comprehend that despite the fact that we compose G ($f^*, ., ., .$.) to signify that G relies on the classifying or regression function f^*, the capacity G does not rely on upon the structure of f^*; G can just access f^* as a black box, for instance in a cross-validation scheme. Additionally, G does not rely on upon the particular learner $\widetilde{T}$. As it were, $\widetilde{T}$ could be any off-the-shelf classification algorithm and G controls the pursuit through the space of feature subsets.

On“the off chance that we permit G to rely on upon the learner $\widetilde{T}$ and on parameters of f^* we get the Eq. (2.2), as:

$$\underset{\sigma \epsilon \{0,1\}^n}{inf} G\left(\alpha^*, \widetilde{T}, \sigma, X, Y\right) \;\; s.t. \;\; \begin{cases} s(\sigma) \leq \sigma_0 \\ \alpha^* = \widetilde{T}\,(\sigma, X, Y) \end{cases} \tag{2.2}$$

Some embedded techniques don't make utilization of a model choice criterion to assess a particular subset of features. Rather, they specifically utilize the learner $\widetilde{T}$. Expecting that many learning strategies $\widetilde{T}$ can be formulated as a streamlining issue, we will have Eq. (2.3) as:”

$$\alpha^* = \underset{\alpha \epsilon \Lambda}{argmin} \, T(\alpha, \sigma, X, Y) = \tilde{T}\,(\sigma, X, Y) \tag{2.3}$$

We can change the minimization issue for the unique instance of G = T as Eq. (2.4):

$$\underset{\alpha \epsilon \Lambda, \sigma \epsilon \{0,1\}^n}{inf} T(\alpha, \sigma, X, Y) \quad s.t. \quad s(\sigma) \leq \sigma_0 \tag{2.4}$$

Lamentably, both minimization issues (Eq. 2.2) and (Eq. 2.3) are difficult to solve. Existing embedded strategies roughly solve the minimization issue. One of the ways that embedded strategies take care of the problem as indicated by (Eq. 2.2) or (Eq. 2.3) is the techniques that iteratively include or expel features from the data to avariciously surmised a solution of minimization issue (Eq. 2.4).

These techniques are known as Forward-Backward Methods and can be gathered into three classes. The first category is Forward selection strategies, where, the method begins with one or a couple features chosen according to a method particular selection criterion. More features are iteratively included until a halting measure is met. Second class is Backward elimination strategies that this sort begins with all features and iteratively reject one feature or bunches of features. The third classification is Nested techniques that amid an iteration features can be included and in addition deleted from the information.

2.9 SVM FOR CANCER CLASSIFICATION

SVM“is a classification algorithm supported by factual learning hypothesis ((Huang, Chen et al. 2017). Because of the outrageous meager condition of microarray gene expression data, the tenuity of the input space is sufficiently high so that the cancer classification is as of now as linear dissociable assignment (Elyasigomari, Lee et al. 2017). It is superfluous and even futile to exchange it to a higher absolute feature space with a non-linear kernel. Thus, in present work, we take linear SVM (Burges 1998) as the fundamental classifier as Eq. (2.5).”

$$\text{Linear kernel } K\,(x, y) =< x, y > \tag{2.5}$$

Where x and y are points in a d-dimensional Euclidian space. For a linear kernel SVM, the margin width can be ascertained by Eq. (2.6) and Eq. (2.7).

$$w = \sum_{i=1}^{Ns} \alpha_i y_i x_i \tag{2.6}$$

$$\text{margin width} = {}^{2}/_{\|w\|} \tag{2.7}$$

Where *Ns* is the count of support vectors, which are characterized as the training samples with *0<α_i≤C*. SVM is accepted to be a prevalent model for sparse classification issues contrasted with different models (Elyasigomari, Lee et al. 2017). In any case, the sparseness condition of a microarray dataset is extreme to the point that even an SVM classifier cannot accomplish an acceptable execution. A preprocessing venture of gene selection is vital for more solid cancer classification.

2.10 BIOLOGICAL MODELING

From biological point, all the actions in biological-based systems happens in discrete and nondeterministic approach. However, in traditional methods such as Ordinary Differential Equations (ODE) approach procedures are handled in continuous and deterministic approach which totally ignores the real actions and reactions as they really are in biological systems (Chandren and Abdullah 2011). To adjust with biology, algorithms from computation point of view can be divided to deterministic and non-deterministic algorithms. In non-deterministic algorithms, the output cannot be predicted because there are multiple possible outputs for each input. It means different outcomes conclude via various routes. The potential of membrane computing to imitate the biological system's concept in nondeterministic approach (Currin, Korovin et al. 2017) indicates it is a suitable method to tackle with the limitation traditional approaches like ODE are facing with. From simulation perspective in P systems, the concept of non-deterministic approach got the same meaning as probability, it means in any time unit any set of rules which are proposed via P system have choice to be chosen or not chosen to execute. In sum, the privilege of building a membrane inspired method in compare with traditional methods such as ODE is its non-deterministic approach makes it very suitable method to imitate biological behavior of systems such as cells and genes (Kumar, Victoire et al. 2012).

Non-deterministic algorithms also can be divided to finite automata which are deterministic algorithms that simultaneously trace all possible paths of non-deterministic problem, and probabilistic algorithms that determine choices via random number generator. "Probability theory (Jaynes 2003) has been used to understand stochastic behavior of biological systems, and the mathematical analysis based on this theory provides complete description of properties for simple random systems. Stochastic simulation is the way to simulate the dynamics of a system by capturing the random phenomena to understand the model and to extract the many realizations from it in order to study them (Donnet and Robert 2012). In recent years, numbers of algorithms have been devised to deal with stochastic character of biological systems (Modchang, Nadkarni et al. 2010, Wu 2017). Some of them are, stochastic reaction-diffusion simulation with MesoRD (Hattne, Fange et al. 2005), stochastic simulation of chemical reactions with spatial resolution and single molecule (Kim, Nonaka et al. 2017) and Monte Carlo simulation methods for biological reaction-diffusion systems (Kerr, Bartol et al. 2008). However, such attempts focus more on the general behaviors of biological systems without taking into the structure of the system where the behaviors are taking place (Kumar, Victoire et al. 2012). Membrane computing can conceptualize the ideas and models of computation from the structure and behavior of living cell. For example, (Bakir, Ipate et al. 2014) extended simulation and verification platform for two classes of p system: kernel p system and stochastic p system. Moreover, the structure and the stochastic behaviors of biology systems modelled with membrane computing have been verified by using stochastic simulation strategy based on Gillespie algorithm (Gillespie 2000, Gillespie 2001) in the study of (Chandren and Abdullah 2011, Wu, Tian et al. 2016)."

In terms of discrete characteristic of biological systems, study of (Szekely and Burrage 2014) compares accuracy and computational effort in various stochastic methods as it is shown in Figure 2.4. Also, based on (Szekely and Burrage 2014), we have summarized explanations of how these methods are suitable based on molecular population in Table 2.4. It can be seen that discrete stochastic and spatial discrete stochastic methods can consider defining based on stochastic p system for a gene selection and classification method inspired by membrane computing.

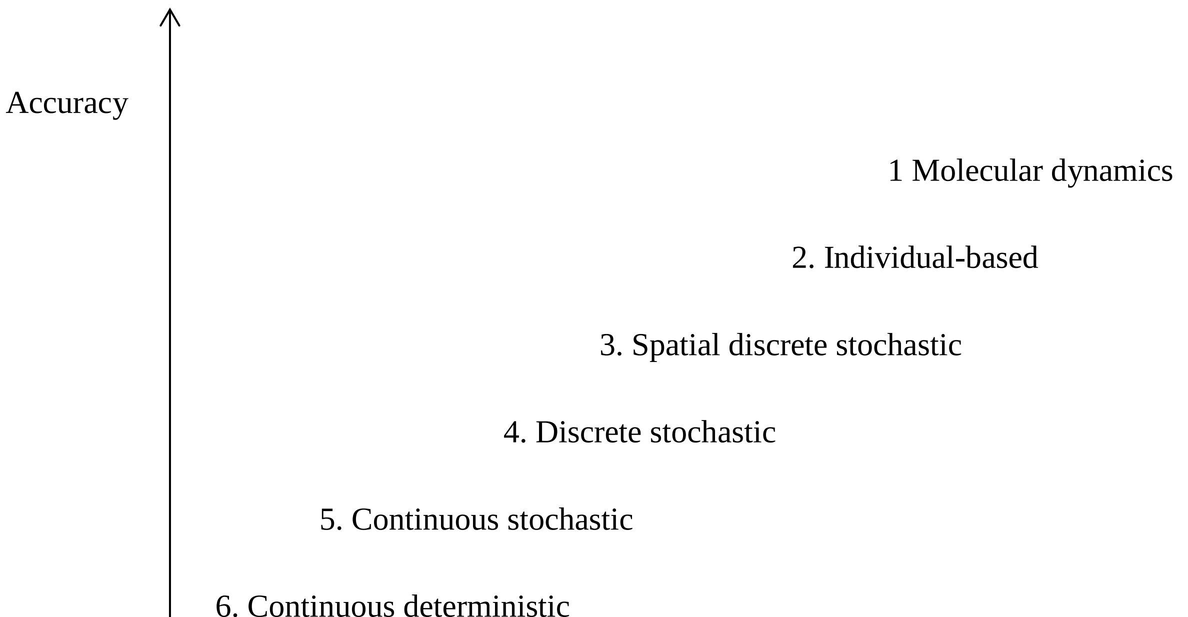


Figure 2.4 Accuracy versus computation in simulation methods (Szekely and Burrage 2014)

Table 2.4 Different methods based on molecular population (Szekely and Burrage 2014)

Molecular Population	Noise level	Suitable Method	Reason
Large	Low	6.continues deterministic	Continuous and deterministic — that is, their state variables are real numbers representing the concentrations of molecules and they do not include noise (Chen, Niepel et al. 2010). Such models are indeed useful for many problems, but they can only be regarded as accurate when we are interested in the mean dynamics of a large number of molecules, large enough that we need not worry about individual molecules but can approximate them as concentrations. This becomes viable when molecular populations are of the order of many hundreds or above. Above this population size, the fluctuations from intrinsic noise are averaged out and the deterministic approximation becomes increasingly valid. This is because intrinsic noise, as a rule of thumb, behaves as 1/ $\sqrt{x}$ where X is the number of molecules in the system.
Intermediate (relatively	high	5.continues stochastic	Continues stochastic

large)			(stochastic differential equations) (Talay 1994) can approximate molecular numbers as concentrations, but also include the effects of noise. They are similar in form to deterministic differential equations, but contain extra terms that represent noise (Gillespie 2000, Higham 2001).
Small	Very high	4. discrete stochastic	Intrinsic noise rapidly increases as molecular populations decrease, and it often becomes necessary to include the effects of stochasticity in biological models, especially for small systems with low populations of some molecular species, such as gene expression networks (McAdams and Arkin 1997).
Small and limited in compartments/membrane	Very high	3.spatial discrete stochastic	Intrinsic noise rapidly increases as molecular populations decrease

To be continued…

…continuation

* Molecular Population can be genes (parts of DNA copied into mRNA) or proteins(RNA)

There is excellent potential in the related area of spatial stochastic methods (Szekely and Burrage 2014). The limitation of non-spatial methods is that they can only be accurately applied to spatially homogeneous systems, but it is not suitable for the systems have biological basis. For instance, the membrane of the cell is an extremely heterogeneous environment, and even the cytoplasm contains many macromolecules that impede diffusion (Sturrock 2016).

2.11 PRELIMINARIES OF PROPOSED MEMBRANE-INSPIRED MULTI OBJECTIVE BINARY SWARM OPTIMIZATION

2.11.1 Biological Justification of MObPSO Method

In this part details of MObPSO justified by stochastic in system biology according to Table 2.5 and Table 2.6.

Table 2.5 Membrane-inspired feature selection and classification method

Feature selection and classification method	Biology (in vivo/vitro)	Models (in info)	Implementation

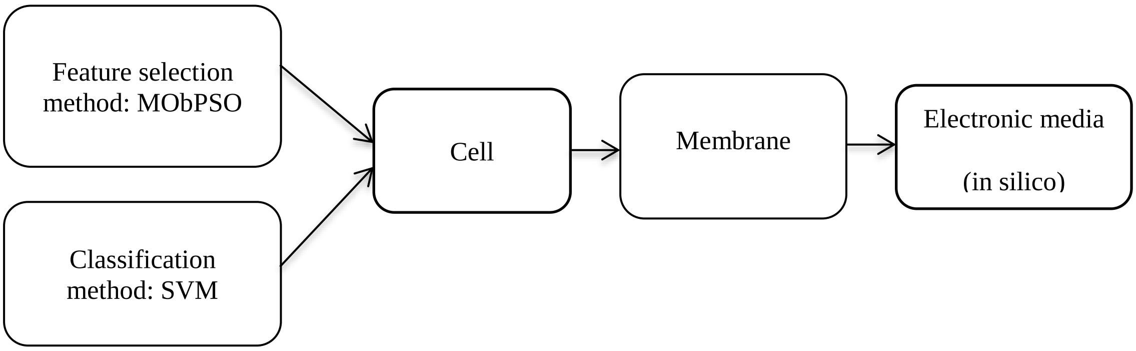


Table 2.6 Biological essence of membrane computing

Description of the basic components in a biological system	Description of the components in membrane computing	Biological justification of membrane computing
Compartment: A container enclosed by a membrane such as mitochondria, chloroplasts, cytoplasm, endoplasmic reticulum, nucleus, and Golgi apparatus. These organelles have their own chemical reactions	Compartment: The compartment is represented with a label or number using square brackets, $[\ ]_i$	The cell membrane is a biological wall that separates the interior of a cell from the outside environment. This cell membrane surrounds the living cells. It is semi-permeable which makes possible for elements to get inside the cell as well as getting out. In the way, it controls the passes of substances in and out of cells.

and species to carry out different metabolic activities.		
Species: A collection of chemically identical molecular structures, such as genes, proteins, ions, and molecules, that perform reactions to characterize certain behavior based on their concentrations.	System structure: The structure of the system (μ) is represented by considering the links between compartments.	Membranes in living cells are flexible structures. Through the membrane the cell gets nutrients and releases the elements that does not need anymore. Also, they are part of the chemical reaction occurring inside the cell. The plasma membrane also serves as the attachment point for the intracellular cytoskeleton and, if present, the extracellular cell wall. Permeability was essential when membrane computing was created. The permeability of a membrane is the ability for the molecules to pass through it. Permeability depends mainly on the electric charge of the elements in contact with it. Electrically neutral and small molecules pass the membrane easier than charged, large one.
		Formal paradigm: A region within a membrane can contain objects or other membrane. A P-system has an external membrane (also called skin membrane) and it also contains a hierarchical relation defined by the composition of the membranes.
Modifier: An activator or inhibitor of reaction such as an enzyme that enhances or inhibits the reaction without changing their concentration.	Objects: The species including the modifiers are assumed as objects (V) in the system.	Cellular components are stored in two basic categories: active and passive ones. The active components are molecular agents coded into an artificial genome or array chromosome: proteins, enzymes, receptors, etc. the passive components are signals, substrates, nutrients, metabolites, membrane elements, etc. which are not coded as such in the genome.
Reactant: A species that acts as a substance consumed in the course of a chemical reaction.	Initial Multiset: The multiset (ωi) is the combination of objects in compartment i at step 0.	
Reaction: A process that transforms one set of species acting as reactants to another set of species acting as products.	Reaction Ri is the reaction in compartment i.	Chemical reactions: In the living cells, there are some ongoing processes. Cells processes food and nutrients in a very characteristic way. Living cells get atoms or molecules and then react by producing other atoms and molecules. Living cells act and process nutrients according to certain rules that take place in a parallel and a non-deterministic manner. This model has been used to establish new machines that process

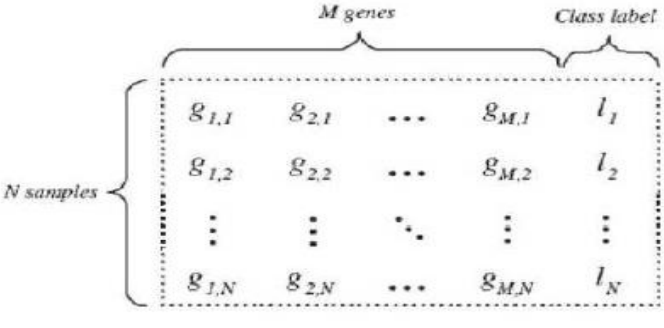
To be continued…

…continuation

		information in the same way. This part belongs to the molecular computation.
Product: A species formed during a chemical reaction.	Objects: the combination of objects which produce via reactions are acted on initial multiset of objects or latter subsequent of objects	As explained in objects and initial multisets.

a. Input Data Matrix (dt)

Microarray is a 2D array whose rows represent samples or experimental condition and columns represent genes. Originally in raw data besides genes one extra column can be viewed which corresponds to class label as shown in Figure 2.5. Most of the genes are not very significant to the corresponding class label, hence they are not useful for phenotype classification. Moreover because of large size of the microarray data, first through standard deviation and SNR sorting few genes are selected. These selected genes have low standard deviation and less noise and are further normalized to get ultimate data for processing.



$$\begin{array}{ccccc} g_{1,1} & g_{2,1} & \cdots & g_{M,1} & l_1 \\ g_{1,2} & g_{2,2} & \cdots & g_{M,2} & l_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ g_{1,N} & g_{2,N} & \cdots & g_{M,N} & l_N \end{array}$$

Figure 2.5 A microarray matrix with N samples and M genes

For example, by using BRB array tool for filtering and normalization, as it is shown in Figure 2.6 a) before normalization, 2.6 b) and 2.6 c) after normalization.

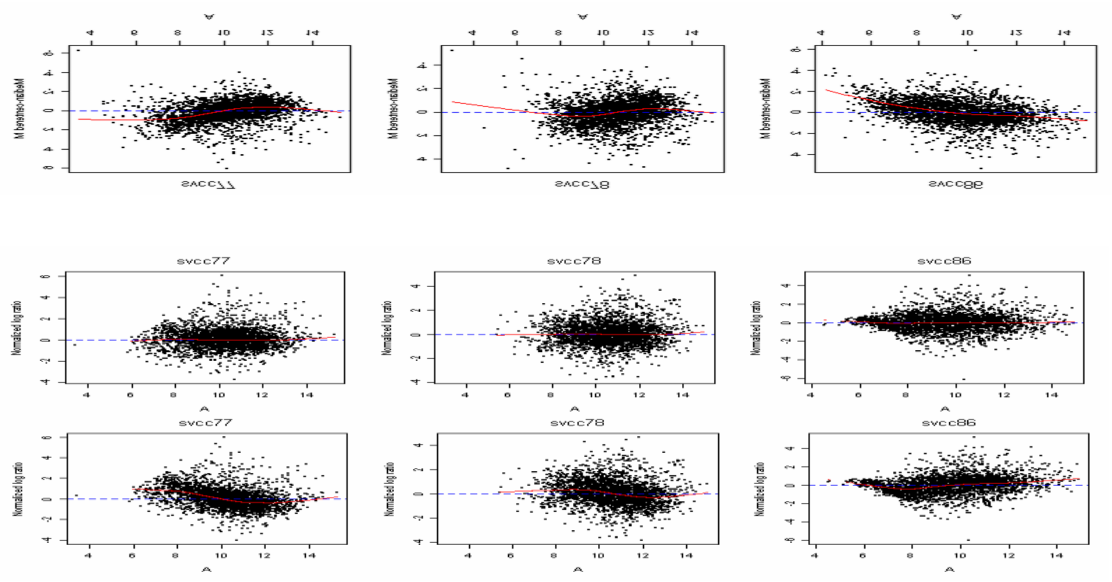


Figure 2.6 a) Before normalization, b) Global median normalization and c) Global loess normalization

b. Heterogeneity

Heterogeneity is a key property of biological systems at all scales: from the molecular level, all the way up to the population level. We can classify heterogeneity as Figure 2.7 in three main sources: genetic (nature), environmental (nurture) and stochastic (chance).

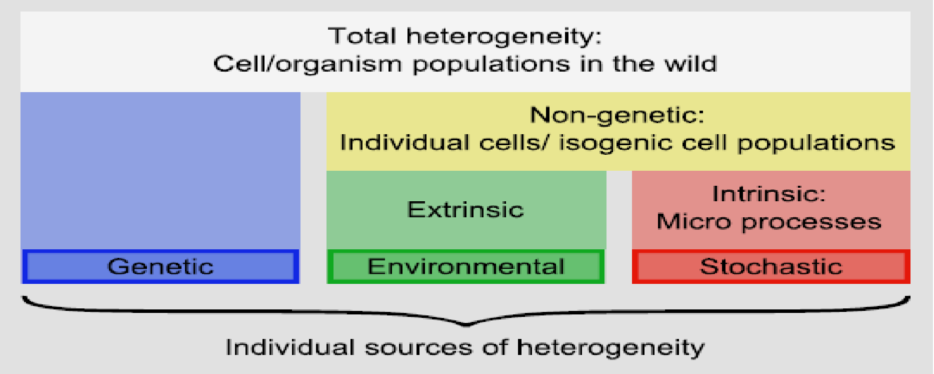

Total heterogeneity:
Cell/organism populations in the wild
Non-genetic:
Individual cells/ isogenic cell populations
Extrinsic
Intrinsic:
Micro processes
Genetic
Environmental
Stochastic
Individual sources of heterogeneity

Figure 2.7 Sources of heterogeneity

About genetic, recent work has shown that there are a surprising number of cases of convergent (phenotypic) evolution with a corresponding convergence in genotype. Environmental or extrinsic examples of each of these factors could be the weather an animal experiences in a particular year; the amount of water or oxygen in the blood of an animal; the pH or nutrient level in which a cell population is propagated; numbers and positions within each cell of shared gene expression machinery such as ribosomes; and cell cycle stage or cell age, respectively. Clearly, not all of these sources will be considered for the same system, and they must be chosen phenomenologically. For instance, if we were interested in animal populations we would look only at the external environment of an animal, whereas if we were interested in levels of protein expression of a cell population, we might look at differences in individual cells such as ribosome number and cell cycle stage. As always, the choice of what to include in the model lies with the modeler. Stochastic (Intrinsic or noise or chance) arises from random thermal fluctuations at the level of individual molecules. It affects the DNA, RNA, protein and other chemical molecules inside cells in many ways, most notably by ensuring that their reactions occur randomly, as does their movement (via Brownian motion). In contrast, extrinsic heterogeneity arises from other, outside, sources and affects all genes inside a cell equally.

It is possible to separate the contributions of intrinsic and extrinsic noise in a gene expression network inside a single cell (Figure 2.8). However, in the presence of both intrinsic and extrinsic noise, the expression of the two proteins would be uncorrelated, as intrinsic noise affects the expression of each protein differently. The respective noise contributions can also be visualized on a plot of the expression level of green versus red fluorescent proteins.

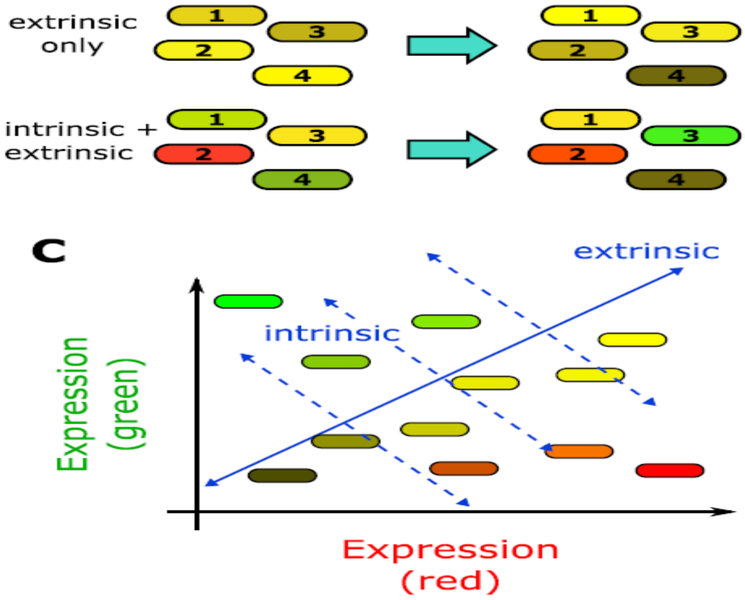


Figure 2.8 Contribution of noise type

c. Swarm Initialization and Particle Representation

Initially the first part of the candidate solutions is randomly chosen values between 0 and 1, and the next part of the candidate solutions are randomly selected genes from the data set. After the initial particles are generated randomly, their corresponding fitness values are calculated. Then the velocity of each cell and cluster are initialized to zero. In the second part, instead of velocity of each particle dimension, velocity of each gene-cluster is taken for moving a gene-cluster of a particle to same direction in the search space. The inputs of the technique for example are (swarm size=15, upper bound of gene cluster size=15, lower bound of the gene cluster size=3 and weighting factors c1 and c2 which are cognitive and social parameters respectively are set to 2. The number of iterations is taken as 50 for each dataset and the threshold of the padding cell is taken as 0.5).

Each particle has two parts; the first part contains n padding cells and next part contains n cluster centres or genes. The first n padding cells contain values between 0 and 1, and the last n cells of the particle consist of n genes from the dataset. Basically, each gene represents the centre of a gene-cluster. Initially the genes are chosen randomly from dataset f_1, f_2, f_3, …, f_g where g is the number of genes present in the data matrix. Each gene contains s sample values. Since each gene has the dimension s, the size of the last part of the particle is *n*s*. Thus, the total length of a particle is *(n+n*s)*. If the value of i^{th} padding cell is greater than a specific threshold then the gene represented by the i^{th} cell of the second part of the particle is selected for fitness Computation. The particle encoding scheme has been demonstrated in Figure 2.9 where a full particle is shown and where only the last n part of the particle is depicted.

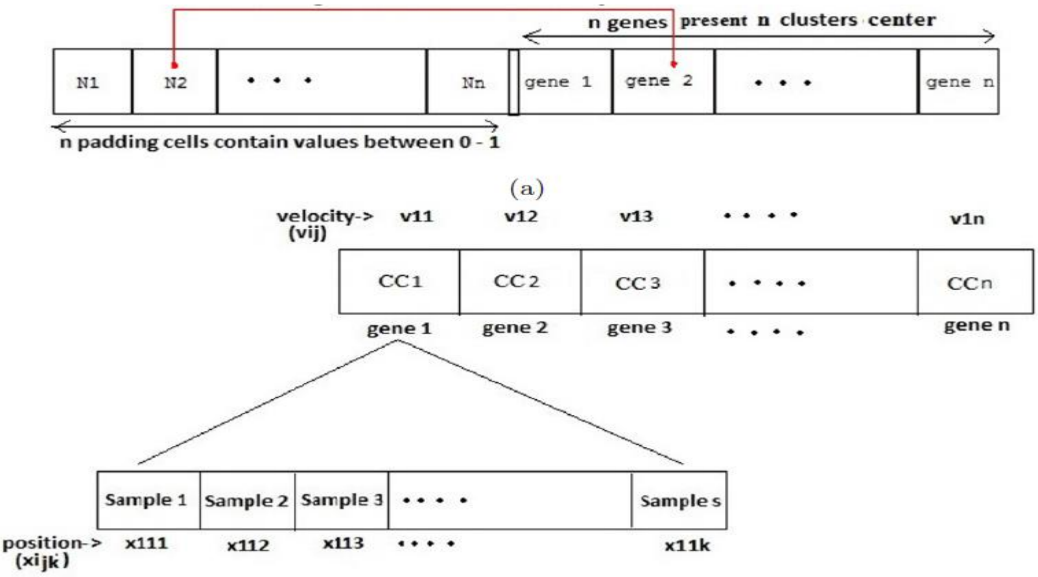


Figure 2.9 Particle encoding

d. Origin of Input Data Matrix in Biology

We will refer to reactions occurring inside a cell (Figure 2.10) volume between different types of biochemical molecules (or, interchangeably, particles) with various reaction rates. These will generally be genes (for our purposes, sections of DNA that are copied into mRNA and initiate a gene expression network), RNA or proteins, as well as chemical compounds. As we want to model feature selection in membrane and feature selection means gene selection, the type of particle which will be our concern is gene.

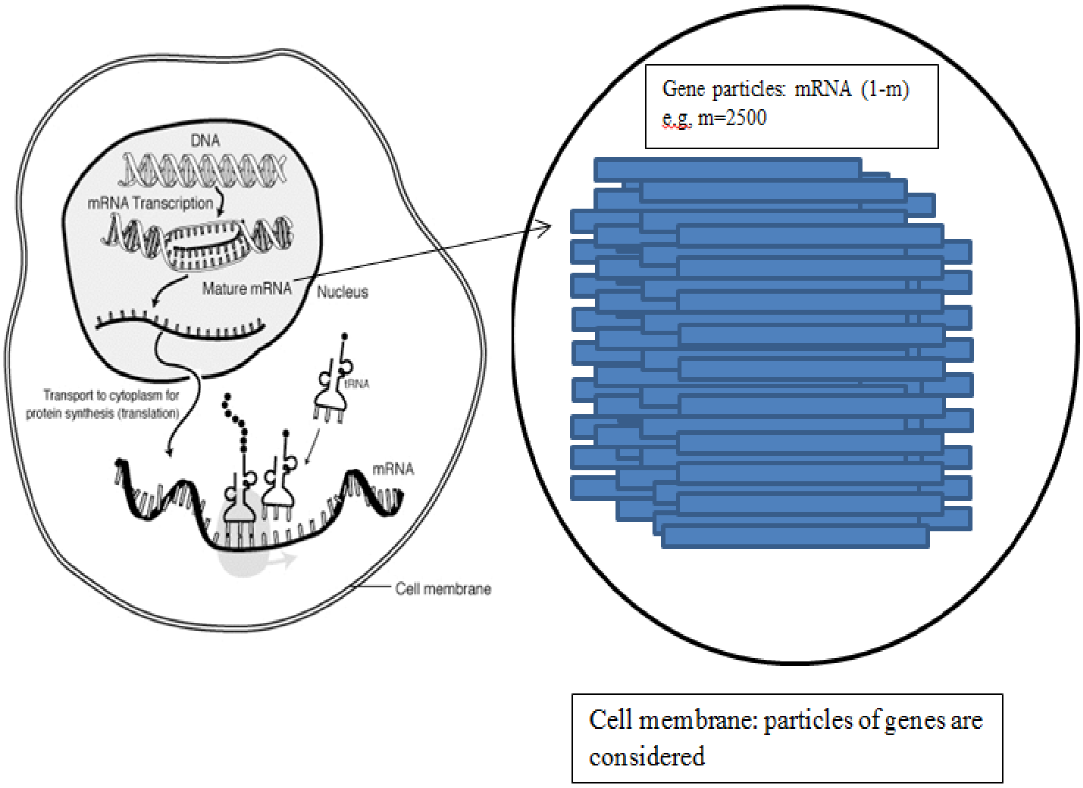

DNA
mRNA Transcription
Mature mRNA
Nucleus
Transport to cytoplasm for protein synthesis (translation)
tRNA
mRNA
Cell membrane
Gene particles: mRNA (1-m)
e.g. m=2500
Cell membrane: particles of genes are considered

Figure 2.10 Cell membrane

Cancer is a phenotypic complexity which affects genes, proteins, pathways and regulatory networks. The research about identifying the important genes which are responsible for various types of cancer is still in progress. In this context, important genes refer to the gene marker which indicates change in expression or state of protein that correlates with the risk or progression of a disease, or with the susceptibility of the disease to a given treatment. However, extracting these marker genes from a huge set of genes is a major problem. There are many approaches for detecting these informative genes. Most of the approaches can find a set of redundant marker genes. Motivated by this fact a multi-objective optimization method has been proposed which can find small set of non-redundant disease related genes which have high sensitivity, specificity and accuracy at the same time.

When samples of the microarray dataset represent normal (benign) and cancer (malignant) tissue, classifying theses samples is called binary classification. Otherwise when samples represent various subtypes of cancer then classification is called multiclass cancer classification. In both cases genes with significantly different expression in two different classes (normal and tumor or two different subtypes of cancer) are called differentially expressed genes. Therefore, these genes are used as indicator or marker of disease and are also called biomarker. In real life, biomarker prediction is very important because it helps in disease prevention, early diagnosis, drug target identification etc.

e. Dissimilarity and relevance

The goal is to find non-redundant and relevant features from a data matrix that means the resultant features are non-correlated as well as relevant. So, the problem should be defined in such a way that the correlated and irrelevant features are not selected. In MObPSO, the problem is formulated as a problem of densest subgraph finding problem from a weighted undirected graph. The structure of the data matrix can be viewed as a two-dimensional matrix; the rows imply instances and columns imply attributes or features; one extra column is used for representing the corresponding class labels of the instances. Some similarity/dissimilarity measures include correlation coefficient, least square regression error and maximal information compression index etc. Using one of these similarity measures the symmetric matrix is generated which is termed as a similarity matrix. Let the data set has n features, $F = \{f_1, f_2, f_3, \dots , f_n\}$. Calculating pairwise similarity between features of the feature set F we generate the $(n * n)$ symmetric similarity matrix where both n rows and n columns correspond to n features. Therefore, from this similarity matrix a weighted complete graph G can be formed. Each node represents a feature, so the vertex set of the graph G is $V = \{f_1, f_2, f_3, \dots , f_n\}$, i.e. the graph contains total n number of nodes. The values present in intersection of row i and column j in the similarity matrix *Sm* represents the weight of the edge between node f_i and f_j. As each feature has some similarity value with every other feature (present in similarity symmetric matrix *Sm*), hence the graph G is a complete graph. Figure 2.11. demonstrates the process of conversion from data matrix to feature graph.

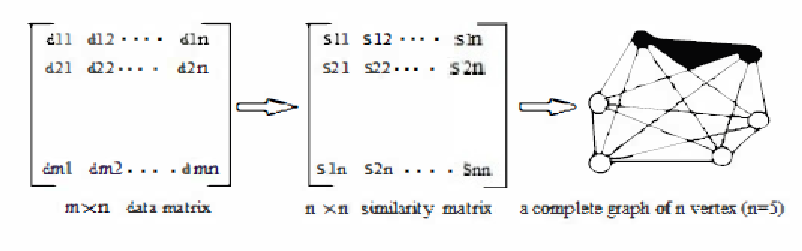


Figure 2.11 Data matrix, similarity matrix and graph formulation

The similarity matrix (edge weight) is calculated for the data matrix using correlation coefficient Eq. (3.1) between each pair of gene.

For the graph G, lower edge weight means that the features connected by that edge are more dissimilar and larger the node weight means that the features are more relevant. Thus, finding the densest subgraph (g) from graph G is equal to finding the most non-redundant and relevant feature set because the features (nodes) contained by the subgraph (g) will have minimum average edge weight (similarity) and maximum average node weight (SNR) Eq. (3.2)."

f. SNR in Biology

There is considerable heterogeneity at every scale of biological systems. The first two sources are now well-known, but until recently the effects of intrinsic noise have generally been ignored in biology, conceptually as well as in them a thematical and computational methods that have traditionally been used. We define noise level here as the coefficient of variation of the abundance of a molecule, that is, the standard deviation of its distribution divided by its mean (SNR=MEAN/S.D). Roughly, the above rule of thumb arises because molecular reactions are random in birth–death processes (Figure 2.12).

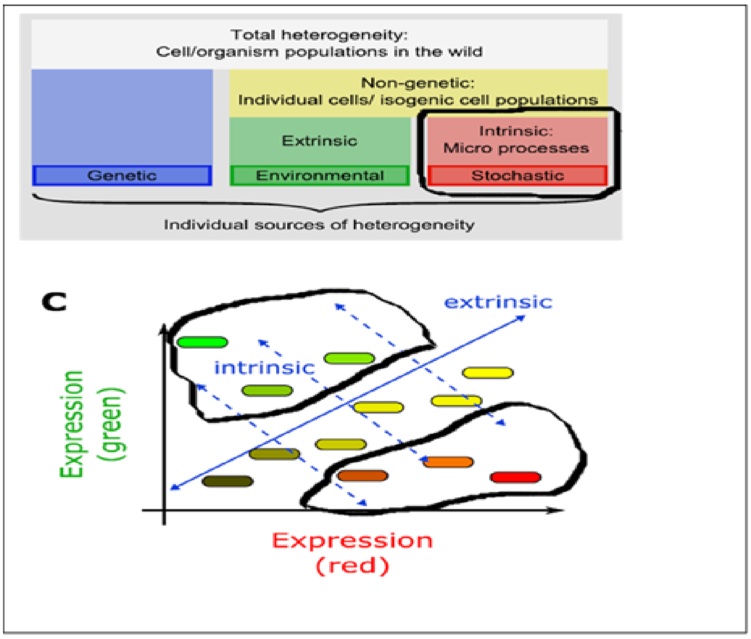


Figure 2.12 Origin of SNR

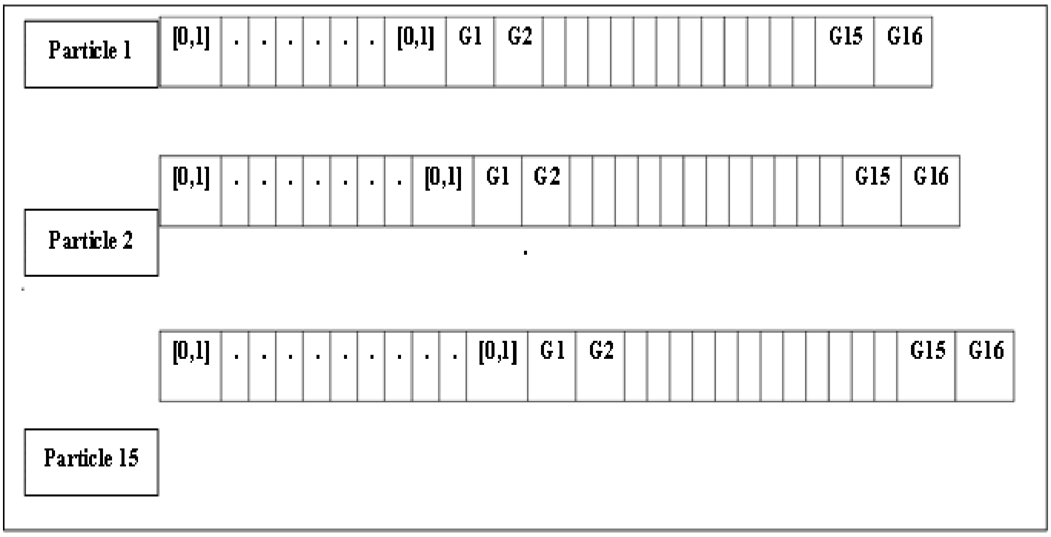
According to the Figure 2.12, Stochastic (intrinsic, noise or chance) arises from random thermal fluctuations at the level of individual molecules. It affects the DNA, RNA, protein and other chemical molecules inside cells in many ways, most notably by ensuring that their reactions occur randomly, as does their movement (via Brownian motion). In contrast, extrinsic heterogeneity arises from other, outside, sources and affects all genes inside a cell equally. Thus, in computation methods when we calculate SNR means ratio of intrinsic noise not extrinsic.

g. MObPSO Updating Values

According to the Figures 2.13 and 2.14, initialization starts with t=0 and swarm size (S) as the number of candidate solutions (particles) assumed as 15 cells. N which is the number of genes in candidate solution (particles) is 16. Maximum number of chosen genes are 16 and minimum number of chosen genes are 3 as c1 and c2 respectively. The number of iterations is 50 and padding threshold is 0.5

Randomly chosen genes in each particle; in each particle N cells contain values between 0 to 1 and remaining n cells represent n cluster centers or genes.

Based on the number generated by random generator [0, 1] if padding cell (i) > threshold then i^{th} gene is selected for fitness computation.



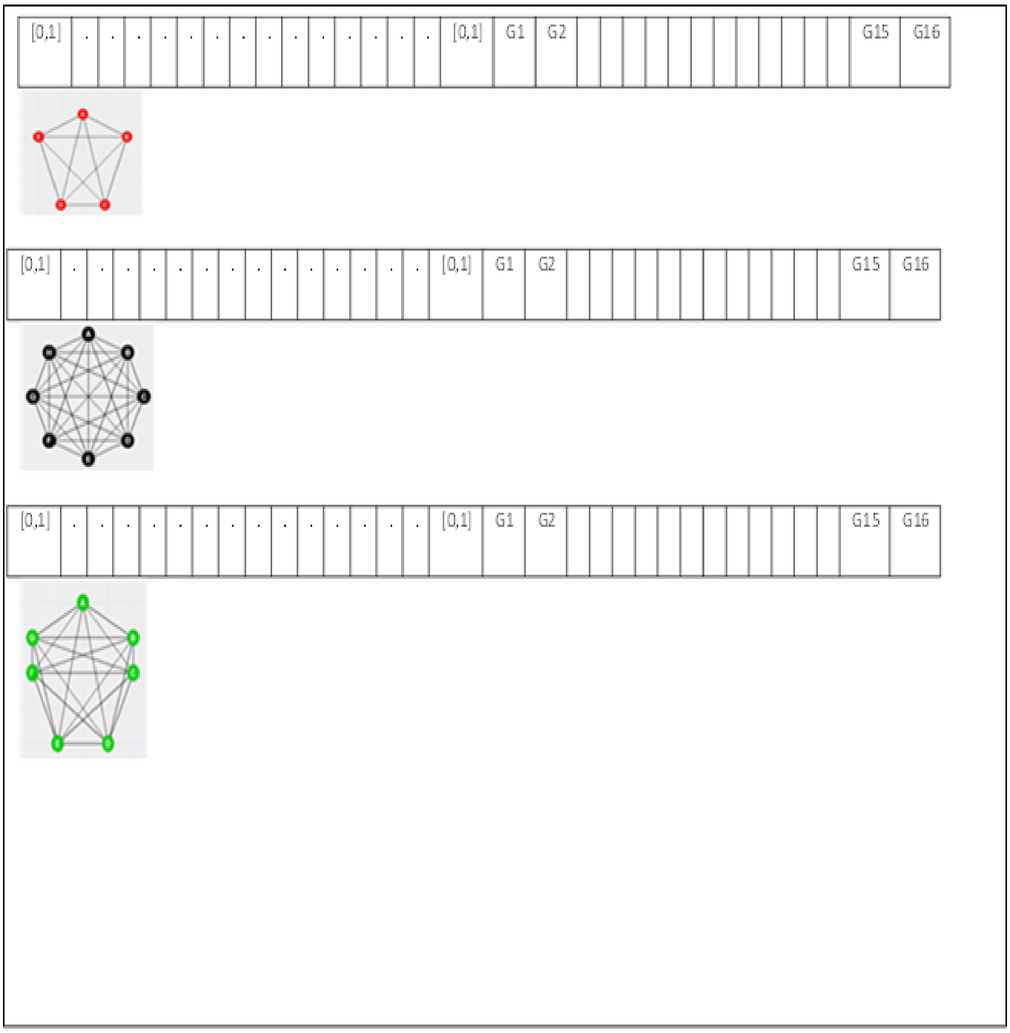


Figure 2.13 Gene value and complete graph

[set of edges as *g1,2-g1,5-g1,7- g1,10-g2,1-g2,5-g2,7-g2,10-g5,1-g5,2-g5,7-g5,10-g7,1-g7,2-g7,5-g7,10-g10,1-g10,2-g10,5-g10,7*]= Avg_ncorr = $\frac{\sum_{i=1}^{|v|}\sum_{j=1}^{|v|} \mathrm{ew}_{ij}}{\frac{|v|.(|v|-1)}{2}}$, ncorr= 1-corr= $|1-\sigma\,(x,y)|$ where $\sigma(x,y)=\frac{cov\,(x,y)}{\sqrt{var(x)var(y)}}$

e.g.,[set of nodes as *g1*, *g2*, *g5*, *g7*, *g10*]=[SNR as *snrg1*,*snrg2*, *snrg5*,*snrg7*,*snrg10*]= Avg_snr = $\frac{\sum_{i=1}^{|v|} vw_i}{|v|}$

For each particle from 1 to 15 fitness value will calculate as follow

f1=avg_ncorr, if high= means selected genes in particles have min correlation

f2=avg_snr, if high= means selected genes in particle have strong relevance

For all 15 particles at *t=0*, *{f=f1-f2, f p=1, f p=2, …. f p=15}*= Max f p=pbest=gbest

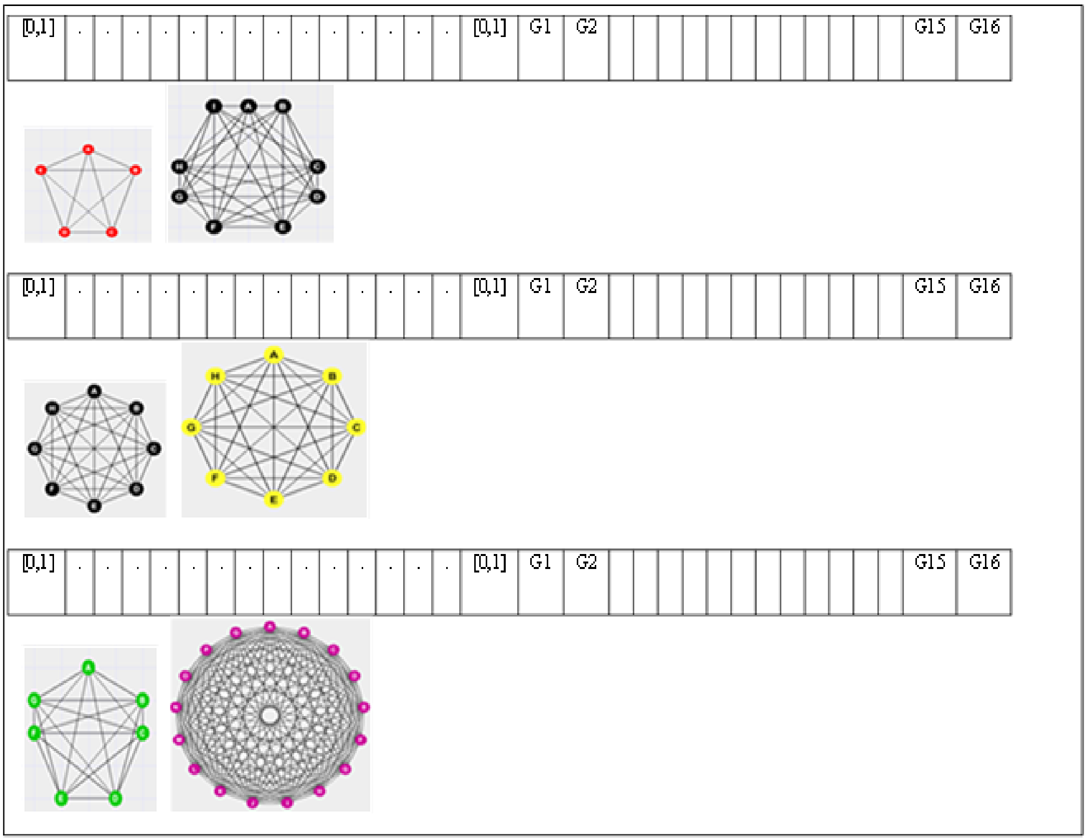


Figure 2.14 Second iteration of complete graph

i. Subgraph

In each iteration, velocity formula will calculate for all C number of genes to decide whether this gene will contribute in the next iteration or not, if x=1 the gene will be chosen for next iteration. After calculating new *Xs* for all C number of genes in each particle, fitness value for all N number of particles will calculate.

Update pbest (local best) and gbest (global best), new fs in each particle will compare with the pbest of last iteration, if new f of particle dominates last pbest, this particle will use this f as pbest for new iteration otherwise pbest of last iteration still is the pbest of this particle for next iteration. Moreover, new fs in N particles will compare with the gbest of last iteration and Max one will choose as new gbest so gbest will update.

2.12 PARALLEL PROCESSING

Typical microarray experiments output the express values for a large number of genes (e.g., more than 20,000). These genes impose significant challenges to any tools which intend to interpret the interactions between genes or link the correlations between the genes and diseases. For filter or wrapper gene selection approaches, the selection process is usually time consuming, especially for high dimensional large size datasets. From a high-performance computing perspective, one of the cheapest ways to speed up computationally expensive algorithms is parallelizing them to execute on cluster-based supercomputers (Raghavan and Chandrasekaran 2016). Cluster computers are becoming the primary means of supercomputing due to their great and improving cost effectiveness.”

A cluster supercomputer consists of a number node, each of which is typically a stand-alone computer with independent memory, hard disk, and operating systems. The clusters are usually connected in a Local Area Network (LAN) environment based on IP network technology, so all clusters can communicate with each other through high speed switches with Gigabit speeds. Designing algorithms to run in such a distributed fashion can be challenging, given that the program instances are running independently and communicate infrequently and at high cost. An application that can be perfectly distributed will execute N times faster on an N node cluster than on a single machine and is thus said to have linear speedup.

Since the model of P systems was presented, many simulators and software applications have been produced. The majority of these simulators have been developed under sequential architectures using languages such as Java, CLIPS, Prolog or C. However, all of sequential P systems simulators are inefficient on time of execution. They serialize the natural parallelism of P systems, and therefore, the performance is dramatically decreased.

We are witnessing the consolidation of the parallel architectures in the newest generation of processors. The last generation of CMP (Chip Multi Processor) processors from both Intel and AMD contains up to 8 cores per die. Moreover, these processors are still organized in clusters of computers, which are extremely expensive and only available for organizations that have enough resources to buy and maintain them. However, other parallel architectures are being consolidated as an alternative computational model. Among these emergent parallel architectures, the newest version of programmable GPUs provides a compelling alternative to the traditional parallel environments such as cluster of computers, delivering extremely high floating-point performance and also a massively parallel framework for scientific applications which fit their architectural idiosyncrasies. The graphics processing units (GPUs) are a kind of computing devices with high parallelism on numerical operations, where massively parallel processors can support several thousands of concurrent threads. The computational power of GPUs has turned them into attractive platforms for general-purpose scientific and engineering applications, especially for tackling large scale numerical computing problems (Ruiz, Ujaldón et al. 2007).

GPUs can support several thousand of concurrent threads providing a massively parallel environment. Current NVIDIA Corporation's GPUs, for example, contain up to 240 scalar processing elements per chip (Graefe 2006), they are programmed using C and CUDA (Havran 2000, Lauterbach, Garland et al. 2009), and they have low cost compared with a cluster of computers.

Clustering of Computers for P system (Peng, Jin et al. 2016) developing on parallel hardware architectures and using of field programming gateway array (FPGA) (Van Nguyen and Gioiosa 2010). The most important attempts to parallelize membrane computing models are being done via using of graphic processing units (GPUs) (Garcıa–Quismondo and Pérez–Jiménez , Cecilia, García et al. 2009, Cecilia, García et al. 2010, Dematté and Prandi 2010, Cabarle, Adorna et al. 2012, Martínez del Amor, Pérez Carrasco et al. 2013, Zhang, Wang et al. 2014). All of these efforts have demonstrated that a parallel architecture is better positioned in performance than traditional CPUs to simulate P systems, due to the inherently parallel nature of them, and specifically GPUs obtain very good preliminary results simulating P systems.

Briefly, a GPU consists of hundreds of blocks, and each block can support several thousands of concurrent threads. A GPU should work with the help of host CPU. Data can be transferred between threads in the same block through the shared memory in the block. But, the transferred data should be very little due to the small size of the shared memory. Data cannot be directly transferred between threads in different blocks, but only through the host. Figure 2.13 depicts the structure of a GPU. Under the GPU implementation presented here, the m cells will work in parallel instead of one by one during the four operations. The parallelism of the m cells is achieved based on an idea as follows; a thread of a GPU does the work of a cell of membrane algorithms, so the m cells can work in parallel through the concurrent threads. This means that GPU has to create the same number of threads as that of cells used by membrane algorithms during the implementation on GPU. Note that, in the GPU implementation procedure of membrane algorithms, a synchronization function has been used after each of the four operations.

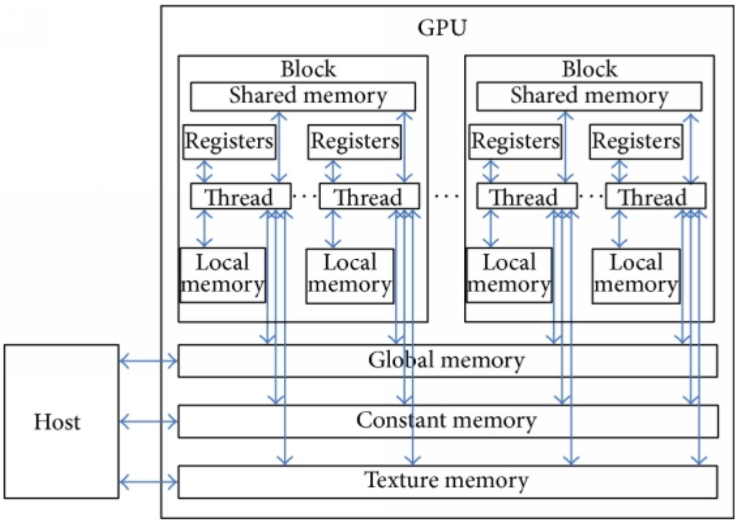


Figure 2.13 a) Structure of a GPU, b) the CPU and GPU implementation procedure of membrane algorithms (Zhang, Wang et al. 2014)

Up to now, it has not been possible to have implementations neither in vivo nor in vitro of P systems. The only way to analyze and execute these devices is through simulators. Therefore, P systems simulators are tools that help the researchers to extract results from a model without the need of having a real implementation.

Since the model of P systems was presented, many simulators and software applications have been produced [2,3]. The majority of these simulators has been developed under sequential architectures using languages such as Java, CLIPS, Prolog or C. However, all of sequential P systems simulators are inefficient on time of execution. They serialize the natural parallelism of P systems, and therefore, the performance is dramatically decreased.

In the study of (Zhang, Wang et al. 2014) the implementation of membrane algorithms on a parallel computing device GPU was carried out by using the case study of a family of P systems that provides an efficient and uniform solution to the SAT problem.

Although the GPU implementation of membrane algorithms has shown a good performance in terms of runtime, many problems remain to be solved for the GPU implementation presented in this work (Zhang, Wang et al. 2014). Among these problems, an interesting one is to further reduce the runtime of GPU implementation of membrane algorithms. In the presented GPU implementation, the data transfer between host and GPU will be performed a large number of times, which takes a lot of runtime. Therefore, a possible solution is to reduce the number of data transfer times between host CPU and GPU. It is conjectured that the runtime of GPU implementation can be greatly reduced by improving the GPU implementation procedure such that only a small number of date transfer times are performed between host and GPU.

2.13 CANCER DATASET

Two types of available cancer dataset from different technological point of view called DNA and RNA data set will be reviewed in this section.

2.13.1 DNA Microarray

All cells have a nucleus, and inside this nucleus there is DNA, which encodes the ‘‘program’’ for future organisms (Feng, Zheng et al. 2017). DNA has coding and non-coding segments. The coding segments, also known as genes, specify the structure of proteins, which do the essential work in every organism. Genes make proteins in two steps as DNA is transcribed into mRNA and then mRNA is translated into proteins. Figure 2.14 displays the general process of acquiring the gene expression data from a DNA microarray. These gene expression profiles can be used as inputs to large-scale data analysis, for example, to increase our understanding of normal and diseased states.

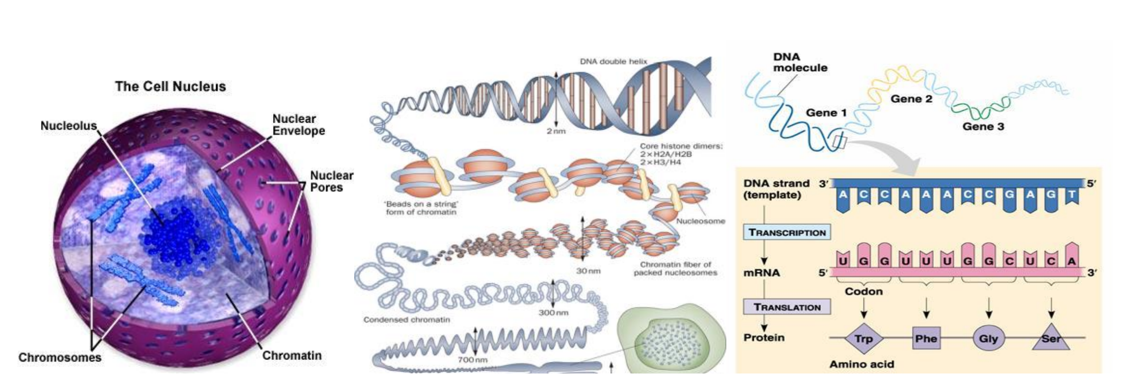


Figure 2.14 DNA origin

Microarrays offer an efficient method of gathering data that can be used to determine the expression pattern of thousands of genes. The mRNA expression pattern from different tissues in normal and diseases states could reveal which genes and environmental conditions can lead to disease. The experimental steps of typical microarray began with extraction of mRNA from a tissues sample or probe. The mRNA is then labeled with fluorescent nucleotides, eventually yielding fluorescent (typically red) cDNA. The sample later is incubated with similarly processed cDNA reference (typically green). The labeled probe and reference are then mixed and applied to the surface of DNA microarrays, allowing fluorescent sequences in the probe-reference mix to attach to the cDNA adherent to the glass slide. The attraction of labeled cDNA from the probe and reference for a particular spot on microarray depends on the extent to which the sequences in the mix (probe -reference) complement the DNA affixed to the slide. A perfect complement, in which a nucleotide sequence on a strand of cDNA exactly matches a DNA sequence affixed to the slide, is known as hybridization. There are two types of hybridization which are called two-channel and one channel. Hybridization in two-channel microarrays is by a pair of samples that the label of first one is fluorescent Cy3; wavelength of 570 nm (green), and the second one by fluorescent Cy5; wavelength of 670 nm (red). These two samples hybridize to one microarray. After that the microarray will be scanned for fluorescence intensity and gene expression will be identified by a ratio of Cy3/ Cy5. In contrast, in one-channel microarrays solely a single sample which is Cy3 will be hybridized to one microarray. Therefore, one-channel microarray reflects the abundance levels of a gene transcript while two channel microarrays reflects the relative abundance between two samples. The process of microarray experiment is illustrated in Figure 2.15.

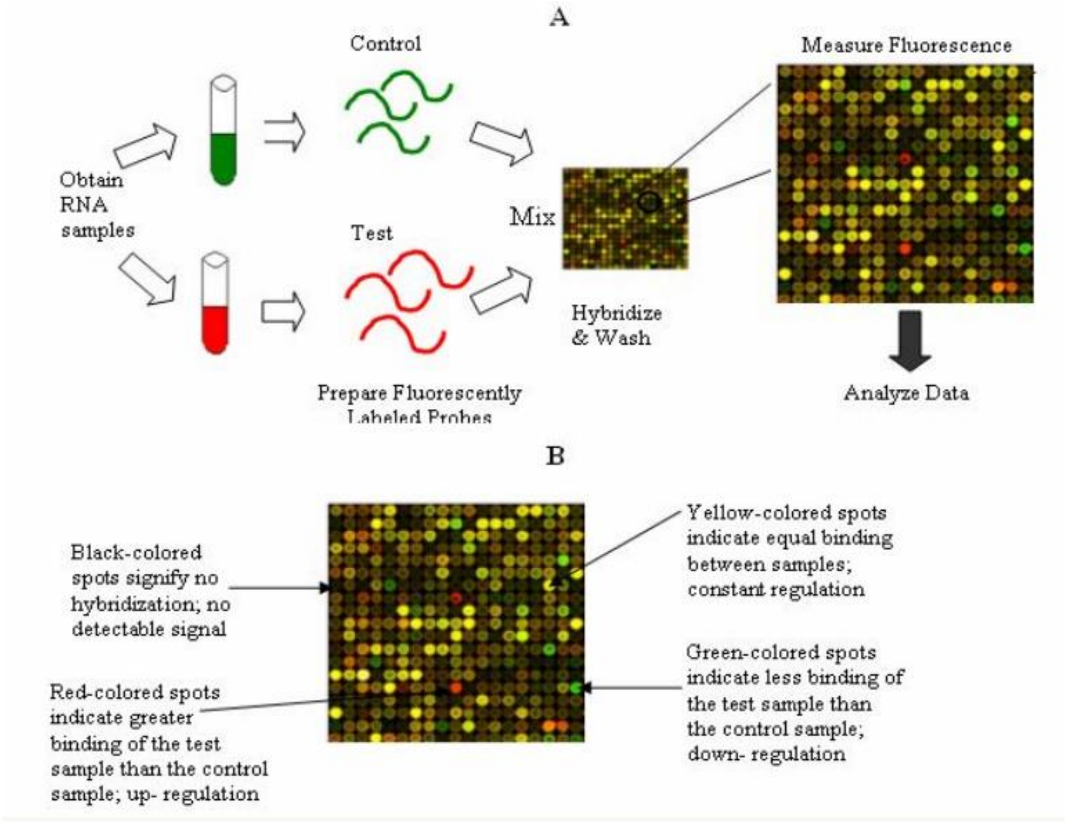

A
Control
Measure Fluorescence
Obtain RNA samples
Mix
Test
Hybridize & Wash
Prepare Fluorescently Labeled Probes
Analyze Data
B
Black-colored spots signify no hybridization; no detectable signal
Red-colored spots indicate greater binding of the test sample than the control sample; up- regulation
Yellow-colored spots indicate equal binding between samples; constant regulation
Green-colored spots indicate less binding of the test sample than the control sample; down- regulation

Figure 2.15 Process of microarray experiment

2.13.2 RNA

Recently, RNA-seq technology is introduced in the field of gene expression. RNA-seq refers to the use of next-generation sequencing (NGS) technologies to sequence cDNA in order to get information about a sample’s RNA content. In terms of power and accuracy some experimental studies are carried out to compare microarrays and RNA-Seq. While some of these studies conclude that RNA-Seq lead to more accurate results ('t Hoen, Ariyurek et al. 2008), some others contradict this conclusion (Willenbrock, Salomon et al. 2009, McIntyre, Lopiano et al. 2011). In fact, to select one of these expression methods, the most important thing is the target of specific study. It means both of microarray and RNA-seq have advantages based on the target of gene expression. For example, the microarray technology has several advantages over RNA-seq (Guo, Sheng et al. 2013); one is the lesser complexity and the other is cost and time efficiency in analyzing than RNA-seq. In contrast, the most prominent feature of RNA-seq over microarray is the detection range of RNA-seq which is not limited to a set of predetermined probes unlike the microarray technology, so RNA-seq is capable of identifying new genes (Guo, Sheng et al. 2013). The study of (Robinson, Wang et al. 2015) conclude that there is not a big difference regarding both significant genes and effect size between RNA-Seq and microarray in high-intensity genes, therefore, the decision about selecting appropriate technology will be made on other factors such as cost. However, the RNA-Seq technology in expression estimation of low-intensity genes shows bigger variation, lower statistical power and higher uncertainty. Microarrays show systematic biases that necessitate cross-platform in low-expressed genes, while unlike the RNA-seq their estimation in technical replication is more consistent. Moreover, the study of (Guo, Sheng et al. 2013) founded high correlation between affymetrix one channel microarray and RNA-seq and they concluded that there is very good concordance between affymetrix one channel and RNA-seq. Indeed, one-channel affymetrix microarray reflects the abundance level of a gene in contrast with two channels which reflect the relative abundance between two samples.

2.14 DISCUSSION

As it is reviewed, traditional methods of feature selection and classification such as filter and wrapper methods are not efficient due to the discussed drawbacks and new researches are focused on more efficient methods like embedded methods and graph methods. Moreover, according to the, the evolutionary algorithms such as GA, KNN and combination of these two methods still facing with the problems as 1) slow convergence, 2) lacks of rank based fitness function and 3) being a time-consuming approach 4) various classification accuracy of the proposed methods in different datasets and 5) overfitting.

In addition to the aforementioned deficiencies of traditional feature selection and classification methods to tackle with the problems of stable accuracy and time efficiency, another problem arises due to the lack of imitating biological process as they really are as it is reviewed. Since, cancer issue and its origin backs to the cell and