Novel Pattern Mining Techniques for Genome-wide Association Studies

Hoang Son Pham

▶ **To cite this version:**

Hoang Son Pham. Novel Pattern Mining Techniques for Genome-wide Association Studies. Bioinformatics [q-bio.QM]. IRISA, equipe GENSCALE, 2017. English. NNT : . tel-01672442

HAL Id: tel-01672442
https://inria.hal.science/tel-01672442

Submitted on 25 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ANNÉE (2017)

UNIVERSITÉ DE RENNES 1

UNIVERSITE BRETAGNE LOIRE

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Bretagne Loire

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Informatique

École doctorale MATHSTIC

présentée par

Hoang Son PHAM

préparée à l'unité de recherche IRISA – UMR6074
Institut de Recherche en Informatique et Système Aléatoires

Nouvelles techniques d'extraction de motifs pour l'étude d'association à l'échelle du génome

Thèse soutenue à Rennes
le 22 December 2017

devant le jury composé de :

Maguelonne TEISSEIRE
Professeure à IRSTEA / *Rapporteur*

Nikolaj TATTI
Chercheur à Aalto University School of Science, Helsinki - Finland / *Rapporteur*

Jacques NICOLAS
Directeur de Recherche à INRIA / *Examinateur*

Agnès BRAUD
Maitre de Conférence à l'Université de Strasbourg / *Examinatrice*

Dominique LAVENIER
Directeur de Recherche à CNRS / *Directeur de thèse*

Alexandre TERMIER
Professeur à l'Université Rennes 1 / *Co-directeur de thèse*

Great achievers are driven, not so much by the pursuit of success, but by the fear of failure.

Larry Ellison, Co-founder of ORACLE

Acknowledgments

Contents

Résumé de la thèse

La modification d'un simple nucléotide, à une position très précise dans le génome, peut accroitre ou décroitre le risque, pour un individu donné, de déclarer une maladie. Pour découvrir ces modifications, des études d'association sur un génome complet (Genome Wide Association Study ou GWAS) sont effectuées. Ces analyses comparent les points de polymorphisme (Single Nucleotide Polymorphism ou SNP) de deux groupes d'individus: un groupe affecté par la maladie et un groupe composé d'individus sains. Un SNP est associé à une maladie s'il apparait plus fréquemment dans le groupe des malades que dans le groupe sain. La détection des SNPs aident à mieux cibler les traitements et à prévenir les risques. Les analyses GWAS sont particulièrement utiles dans le cas de maladies complexes comme l'asthme, le cancer, le diabète, les maladies cardiaques ou les maladies mentales.

Le problème est que la plupart des maladies ne sont pas causées par une mutation unique, mais souvent par plusieurs modifications localisées dans plusieurs gènes (i.e. une combinaison de plusieurs SNPs). Par exemple, des maladies génétiques telles que la bipolarité, la schizophrénie, le diabète de type 2 ou quelques cancers, sont polygéniques et montrent une forte hétérogénéité génétique. Ainsi, des patients présentant des symptômes identiques peuvent avoir des profils génétiques différents et peuvent donc répondre différemment aux mêmes médicaments.

Beaucoup de stratégies ont été explorées pour détecter les interactions entre variants génétiques. Plusieurs méthodes sont basées sur des approches statistiques telles que la régression logistique ou les modèles de Bayes. D'autres adoptent des techniques dapprentissage comme les SVM (Support Vector Machines), les réseaux de neurones, les arbres de décision ou les modèles arborescents aléatoires. Ces stratégies sont cependant limitées à l'analyse de petits jeux de données et détectent au mieux des interactions entre 2 SNPs.

Pour pallier ces limitations, diverses solutions basées sur la recherche de patterns discriminatifs ont été investiguées. La recherche de patterns discriminatifs a pour objectif d'extraire des ensemble de SNPs (des patterns) qui apparaissent plus fréquemment dans une classe que dans une autre. Il a été montré que ces entités sont extrêmement pertinentes dans une large gamme d'applications. Plus

spécifiquement, en bio-informatique, cette stratégie a été appliquée à l'identification de combinaisons de SNPs, à l'expression différentielle de gènes ou à la découverte de motifs phosphorylés.

La recherche de patterns discriminatifs possède l'avantage majeur de manipuler efficacement la recherche de combinaison de SNPs. Par contre, cette stratégie rencontre un certain nombre de limitations qui freinent significativement son utilisation dans le cadre d'étude génomiques à grande échelle. Nous listons maintenant quelques-unes de ces limitations et indiquons les défis associés:

1. **Mesure de la force d'association**. Il existe une grande variété de mesures statistiques pour évaluer la force d'association entre les patterns biologiques et les maladies. Déterminer quelle mesure de qualité est la mieux adaptée pour à la fois entériner la découverte d'un motif biologique et guider le processus algorithmique vers sa découverte est un défi à part entière. De plus, pour chaque mesure, il faut généralement choisir (empiriquement) un seuil permettant d'atteindre un haut niveau de qualité. En pratique, c'est une tâche particulièrement difficile. La raison principale est que si le seuil n'est pas strict, un grand nombre de patterns inintéressants est généré. A l'opposé, avec un seuil strict beaucoup de patterns significatifs peuvent ne pas être retenus. Il faut donc définir des stratégies beaucoup plus flexibles pour résoudre cet antagonisme.

2. **Efficacité des calculs**. La recherche de combinaisons d'un nombre de SNPs conséquent accroit fortement la complexité des calculs: le nombre de possibilités augmente exponentiellement avec le nombre de SNPs considérés dans un pattern. Les approches "force-brute" peuvent généralement analyser un petit nombre de SNPs (quelques centaines) tandis que des approches "heuristiques", qui en manipulent beaucoup plus, peuvent ne pas détecter des combinaisons pertinentes. Le défi, ici, est de mettre en place de nouvelles méthodes algorithmiques avec un élagage efficace de l'espace de recherche.

3. **Tests d'hypothèses multiples**. Cette limitation représente un défi encore plus important. Les algorithmes usuels génèrent énormément de patterns. Beaucoup d'entre eux sont découverts par chance. Un grand nombre de tests d'hypothèses est donc nécessaire pour corriger la signification statistique des résultats. Cette tâche est excessivement coûteuse en temps de calcul. Une manière d'attaquer le problème est d'intégrer directement les tests statistiques au sein même du processus de recherche de patterns.

4. **Visualisation des patterns**. Les patterns discriminatifs représentent le résultat de la fouille, mais doivent être validés par des experts. La plupart du

temps, les outils se bornent à générer une liste (trop) importante de patterns, liste qui inclut souvent des éléments redondants. L'interprétation est alors difficile, d'autant plus que le résultat est fourni textuellement, généralement sans éléments contextuels. Il apparait donc nécessaire de restituer les résultats dans un cadre plus confortable pour les experts du domaine, notamment par le biais d'outils graphiques qui permettent rapidement de positionner l'essentiel de l'information.

Le travail de thèse présenté dans ce manuscrit est une contribution à ces différents défis. Plus spécifiquement, nous proposons les solutions suivantes:

Premièrement, pour mieux évaluer la force d'association entre combinaisons de SNPs et maladies génétiques, une stratégie d'analyse flexible est proposée. Elle se base sur la technique des "skypatterns" qui exploite une combinaison de mesures pour juger de la pertinence d'un pattern.

Deuxièmement, pour tenter de résoudre les problèmes d'efficacité et de tests d'hypothèses multiples, nous proposons un nouvel algorithme, appelé SSDPS (pour *Statistically Significant Discriminative Patterns Search*), qui extrait des patterns discriminants à partir d'un jeu de données constitué de 2 classes. Plus précisément, l'algorithme SSDPS recherche des patterns qui satisfont à la fois des scores discriminatifs et des intervalles de confiance. Ces patterns sont définis comme patterns statistiquement significatifs et discriminants. L'algorithme SSDPS se base sur une stratégie de recherche dans laquelle les propriétés anti-monotones des mesures de risque et d'intervalles de confiance sont avantageusement exploitées. Ces propriétés permettent d'élaguer très efficacement l'espace de recherche. De plus, cet algorithme permet de découvrir des ensembles complets de patterns discriminatifs avec un seuil de fréquence très bas. Il utilise également des stratégies heuristiques pour seulement extraire les patterns les plus grands. Des expérimentations sur des jeux de données réels montrent que l'algorithme SSDPS peut effectivement découvrir des combinaisons intéressantes de SNPs en très peu de temps. Beaucoup de ces combinaisons contiennent des SNPs qui sont connus pour être associés à des maladies.

Troisièmement, pour aider à mieux interpréter les résultats de l'algorithme SSDPS, nous avons développé un outil graphique interactif, appelé *SNPvisual*, qui visualise et positionne les patterns de SNPs directement sur le génome. Cet outils intègre également d'autres informations permettant de resituer les patterns dans un contexte biologique.

Bien que ces travaux de thèse se concentrent sur l'étude d'association sur l'ensemble d'un génome, d'autres tâches bio-informatiques telles que la découverte d'expression génique, la recherche de motifs de phosphorylation ou la détection de motifs de régulation, peuvent tirer parti de ces recherches. Il faut noter que le problème de la recherche de combinaisons de SNPs associées à une maladie a été très largement

étudié par le biais d'approches statistiques, d'apprentissage ou de fouille de données. Néanmoins, nous estimons que les solutions proposées dans ce travail de thèse restent originales. Elles apportent un ensemble de techniques complémentaires par rapport à l'état de l'art actuel.

Le manuscrit est structuré en 5 chapitres dont nous donnons rapidement, pour chacun d'eux, le contenu.

Chapitre 1: Etat de l'art sur la recherche de patterns discriminatif en bio-informatique.

Dans ce chapitre, un état de l'art des techniques de recherche de patterns discriminatifs et de leurs applications en bio-informatique est présenté. Une définition précise du problème est d'abord introduite. Puis quelques mesures statistiques standard pour évaluer la puissance de discrimination ainsi que des méthodes de correction de la significativité statistique sont présentées. Nous poursuivons en détaillant quelques algorithmes du domaine avec leur application en bio-informatique. Nous terminons par exposer les défis et les motivations de nos travaux de recherche.

Chapitre 2: Identification de combinaisons de variants génétique avec des "Skypatterns".

Ce chapitre décrit une méthode pour identifier des combinaisons de variants génétiques associées à une maladie avec la technique Skypattern. Cette dernière utilise une combinaison de mesures pour évaluer l'importance de ces combinaisons. Après une introduction sur les mesures de forces d'association et de l'approche Skypattern, nous présentons plusieurs expérimentations conduites sur des jeux de données réels qui démontrent l'efficacité de cette méthode.

Chapitre 3: Recherche de patterns discriminatifs et statistiquement significatifs dans les données génomiques

Ce chapitre présente en détail l'algorithme SSDPS développé dans cette thèse pour extraire des combinaisons de variants génétiques. Les mesures de risque et les méthodes de tests statistiques sont d'abord présentées. Ensuite, la stratégie de recherche basée sur les propriétés anti-monotones des mesures de risques et des intervalles de confiance pour effectuer un élagage efficace de l'espace de recherche est expliquée. Une approche en 2 étapes est proposée: sélection de SNPs candidats, puis recherche de combinaisons. Diverses expérimentations sont effectuées, à la fois sur des jeux de données synthétiques et sur des jeux de données réels. Elles permettent dévaluer globalement les performances de l'algorithme SSDPS.

Chapitre 4: Visualisation des SNPs

Ce chapitre présente l'implémentation d'un outil graphique qui supporte les différentes étapes d'une analyse GWAS. Une vue globale de l'ensemble du logiciel est d'abord exposée. Les différentes fonctions du logiciel sont ensuite décrites.

Chapitre 5: Conclusions et perspectives

Ce chapitre conclut le manuscrit. Il résume les principales contributions et expose les futures directions de recherche.

Thesis Introduction

Single Nucleotide Polymorphism (SNP), a single base pair changes at key positions in the genome, may increase or decrease an individual's risk of getting a disease or benefitting from a particular therapy [1, 2]. To discover SNPs associated with a disease, Genome-wide association studies (GWAS) compare the SNPs of two groups: case group consists of patients with a disease and control group consists of healthy people without the disease. A SNP may be associated with the disease if its occurs more frequently in the case group than in the control group. Once new genetic associations are identified, they can be used to develop better strategies to detect, treat and prevent the disease. GWAS studies are particularly useful in finding SNPs that contribute to complex diseases such as asthma, cancer, diabetes, heart disease and mental illnesses [3].

The problem is that most diseases are not caused by single genetic variations but by variations in many interacting genes (i.e. combinations of SNPs rather than single SNPs) [4]. For example, common genetic disorders (such as bipolar disorder, schizophrenia, type 2 diabetes and various cancer types) are polygenic and show genetic heterogeneity, i.e. the patients have the same phenotype (disease), but their genetic profiles may be different, and they may thus respond differently to different drugs. Thus, discovering high-order SNP combinations associated with interesting phenotype is an important task.

Many approaches have been investigated for detecting the interactions of genetic variants. Some methods use statistical approaches such as Logistic Regression [5] or Bayes model [6], while others adopt machine learning techniques such as support vector machine [7], neural networks [8], decision trees [9] or random forests [10]. These approaches have been effectively applied to discover SNPs interactions in GWAS. However, they are used to tackle only small biological datasets and detect only single or two-locus interactions [11, 12].

To address these limitations, various solutions based on discriminative pattern mining have been investigated [13]. Discriminative pattern mining aims to find patterns (sets of SNPs) which occur more frequently in one class than in the other class. It has been demonstrated that discriminative patterns are very valuable in

a wide range of applications [14, 15]. Discriminative pattern discovery algorithms have been widely applied to tackle different bioinformatics tasks such as identifying SNP combinations [16], mining differential gene expressions [17] or discovering phosphorylation motifs [18].

Traditional discriminative pattern mining techniques have advantages of efficiently handling SNP combinations search. They also have several limitations that prevent them from effectively tackling genomic data due to its unique characteristics. Below are several key challenges that have to be taken into consideration:

1. **Association strength measure**. There exists a variety of statistical measures for evaluating the association strength between biological patterns and diseases. Determining which quality measures are more adapted both for assessing the discovered biological patterns and guiding the search process is a challenge. In addition, for each measure, one has to choose a suitable threshold to get the highest quality result. In practice, it is a very difficult task. The reason is that if the threshold is not strict, a huge number of less interesting patterns are generated. Oppositely, many valuable patterns may be missed by strict thresholds. This calls for more flexible methods to evaluate the association strength between genetic variant combinations and diseases.

2. **Computational efficiency**. The need of searching for high-order SNP combinations leads to increased computational complexity, since the number of possible patterns increases exponentially with the number of SNPs. To search for SNP combinations from high-dimensional datasets, brute-force approaches can handle only a relatively small number of SNPs (tens or hundreds), while heuristic approaches risk missing informative combinations. This challenge calls for novel algorithmic approaches with effective search space pruning strategies.

3. **Multiple hypothesis testing**. Beside the computational problem, multiple hypothesis testing is an even more serious challenge. Existing algorithms often generate a large number of patterns. Many of them could be discovered due to random chance. Thus, a huge number of hypothesis tests are needed to correct the statistical significance of results. This task is very time-consuming. This calls for approaches which integrate statistical tests in the pattern mining process to directly discover statistically significant discriminative patterns.

4. **Interesting patterns visualization**. Discriminative patterns are often used to present result to an expert who will give a decision based on this result. However, existing methods usually generate a large number of the patterns which include many redundant ones. In addition, most of algorithms present

the patterns in the form of long textual lists. This is impractical in many specific biological tasks since the generated patterns are complicated to interpret. In addition, it is difficult for the experts to understand the knowledge that is related to analysis data. This calls for an interactive graphical tool that allows to visualize the discriminative patterns.

To address the above challenges, this thesis aims to advance the state of the art of discriminative pattern mining techniques and apply them to discover genetic variant combinations associated with diseases. In particular, the following solutions have been proposed:

First, to overcome the challenge of evaluating the association strength between SNP combinations and diseases, a flexible evaluation method has been proposed. This method is based on the *skypattern* technique which allows combination of measures to be used to assess the interestingness of a pattern in a threshold-free manner. Experiments on several real variant datasets demonstrate that the proposed method effectively identifies the risk genetic variant combinations related to diseases.

Second, to address the computational efficiency and multiple hypothesis testing problems, we proposed a novel algorithm, named *SSDPS*, that discovers discriminative patterns in two-class datasets. More precisely, the SSDPS algorithm searches patterns satisfying both discriminative scores (equivalent to risk scores) and confidence intervals thresholds. These patterns are defined as *statistically significant discriminative patterns*. The SSDPS algorithm is based on a search strategy in which risk measures and confidence intervals can be used as anti-monotonic properties. These properties allow the algorithm to efficiently prune the search space. In addition, the algorithm can discover a complete set of discriminative patterns with a very low frequency threshold or use heuristic strategies to mine only the largest patterns. Experiments on real SNP datasets: Age-Related Macular Degeneration, Breast Cancer and Type 2 Diabetes show that the SSDPS algorithm can effectively discover interesting SNP combinations in a short execution time. Many of them contain SNPs which are already known as associated with diseases.

Third, to pursue the enhancement of interesting discriminative patterns visualization, we implemented an interactive graphical tool, named *SNPvisual*, to visualize the discriminative patterns in the form of genetic variant combinations in a real chromosome panel. This tool provides various interactive functions to visualize SNP combinations with other related biological information in different genetic variant datasets. This is an efficient and easy-to-use genetic-analysis tool that supports biologists in their search for the relations between genetic variant combinations and phenotype.

Although these solutions are focused on GWAS, other bioinformatics tasks such as gene expression discovery, phosphorylation motif mining and regulatory motif

combinations detection can benefit from these proposed techniques. We also note that the problem of discovering SNP combinations associated with diseases has been tackled with a broad range of work in statistics, machine learning and data mining. Nonetheless, we believe that the solutions which are proposed in this thesis are unique. They provide a complementary set of techniques to discover biological patterns that other techniques were not designed for.

The rest of this dissertation is structured as follows:

Chapter 1: Literature review: discriminative pattern mining for bioinfomatics

In this chapter, a comprehensive review of discriminative pattern mining techniques and its applications to bioinformatics is formally presented. First, a uniform definition of discriminative pattern mining problems is introduced. After that some popular statistical measures for evaluating the discriminative power and statistical significance correction methods are presented. Then, various discriminative pattern mining algorithms with different search strategies and their applications in bioinformatics are detailed. At the end, the remaining challenges which motivate us to propose new efficient approaches, and the thesis contributions are exposed.

Chapter 2: Identifying genetic variant combinations using Skypattern

This chapter presents a method to identify genetic variant combinations associated with diseases by the using skypattern technique. This technique allows combinations of measures to be used to evaluate the importance of genetic variant combinations. First, the background of association strength measures and skypattern technique is introduced. Subsequently, various experiments on different real genetic variant datasets are conducted to demonstrate the effectiveness of the proposed method. Finally, the conclusion of the chapter with a summary and future research directions is given.

Chapter 3: Searching for statistically significant discriminative pattern in genomic data.

This chapter presents in details the SSDPS algorithm which is used to discover multiple SNPs combinations in large genetic variant datasets. First, the background of risk measures and statistical significance testing methods is presented. Afterward, the search strategy that allows risk measures and confidence intervals to be used as anti-monotonic properties to effectively prune the search space is explained. Subsequently, a two-step framework (selecting candidate SNP genotypes step and searching combinations step) is presented to search high-order SNP combinations associated with diseases. Various experiments on both synthetic and real genetic variant datasets are conducted to assess the efficiency of the SSDPS algorithm. At the end, summary of contributions and perspectives of this chapter are given.

Chapter 4: SNP visualization.

This chapter presents the implementation of a graphical tool that supports all

steps of GWAS analysis. First, an overview of the software architecture is introduced. After that the different methods to tackle each step of the software are presented. The main visualization principles which are used to design the graphical tool are discussed. At the end, the visualization results and conclusion of this chapter are given.

Chapter 5: Conclusions and Perspectives

This chapter concludes the thesis with summary of contributions, limitations and future research directions.

Chapter 1

Literature review: discriminative pattern mining for bioinfomatics

Discriminative pattern mining is a powerful task in data mining and machine learning. This task aims to find patterns which occur with different frequencies in class-label datasets. Recently, this technique has been widely applied to tackle bioinformatics problems. This chapter presents the current state of the art discriminative pattern mining techniques and its applications to bioinformatics.

1.1 Introduction

Recently, discriminative pattern mining techniques have been widely applied to tackle bioinformatics problems [19, 15, 13, 14]. They provide efficient methods to detect biologically significant patterns in various biological data. The important applications of discriminative pattern discovery in bioinformatics include identifying high-order SNP combinations [20, 16, 21, 22, 23], searching differential genes expressions [24, 25, 26, 27, 17], detecting phosphorylation motifs [28, 29, 30, 18], discovering regulatory motif combinations [31, 32, 33] and other applications [34, 35, 36, 37]. The sheer volume of biological data increases: constant improvement of discriminative pattern mining algorithms are required to cope with this increase in volume (especially number of genetic variants). In addition, with scientific progresses, the expectations of biologists evolve (and their available time dwindles). Thus discriminative pattern discovery algorithms have to be adapted to take that into account.

There are some existing studies that summarize the recent advances on discriminative pattern mining techniques in the literature [38, 15, 13]. However, to deeply

understand these techniques, this chapter provides a complementary study to discuss the properties, techniques, challenges and applications of available discriminative pattern mining algorithms in bioinformatics.

The rest of this chapter is organized as follows: Section 1.2 presents the definition and the problem of discriminative pattern discovery. Section 1.3 introduces some popular quality measures and statistical significance correction methods which are used to evaluate the interestingness and the statistical significance of patterns. Section 1.4 focuses on various discriminative pattern mining algorithms with different target objectives and search strategies. Section 1.5 illustrates the effectiveness of adopting discriminative pattern mining techniques to handle a variety of applications in bioinformatics. Section 1.6 concludes this chapter with the thesis's research directions and its contributions in the fields of data mining and bioinformatics.

1.2 Preliminaries

Frequent itemset mining is an important task of data mining. This task aims at finding all set of items occurring frequently in a transaction dataset [39, 40, 41, 42]. A typical example of frequent patterns from a dataset of supermarket transactions could be the products that are often purchased together, such as beer and chips or bread and milk. Furthermore, there exists numerous datasets with multiple classes in the real world such as biological datasets with two groups of individuals: patients with a disease and healthy people without the disease, cancer data with different subtypes or marketing data with various classes of customers. Discovering patterns which are discriminative between different classes has also become an essential work. Such patterns are of great value for classifier construction [19, 38] and very interesting in a wide range of applications such as medicine [15], bioinformatics [13, 14] and marketing [43]. For example, in bioinformatics, detecting groups of genetic variants which occur more frequently in the group of individuals which are effected by a disease than in the healthy individuals is an important task. These genetic variant groups can be used to develop better strategies to detect, treat and prevent the disease.

To address this issue, discriminative pattern mining [44], an extension of frequent itemset mining, is investigated to discover patterns in a dataset with multiple classes. This approach aims to find a set of patterns which have differences of frequency across classes. Research on discriminative patterns evolves rapidly under several terms such as emerging patterns [45], jumping emerging patterns [46] and contrast sets [47]. According to these studies, emerging pattern mining detects the set of patterns whose support is significantly larger in one class than in the others. A jumping emerging pattern, a special type of emerging pattern, is defined as a pattern

which is present in one class but absent in the others. Similarly, contrast set mining aims at seeking patterns that have different levels of frequency in different groups of individuals. Overall, although different names are used for these patterns, they are similar in essence. Accordingly, we refer to all these patterns as discriminative patterns. In this section, the main definitions and the problem of discriminative pattern discovery are introduced.

Let $I = \{i_1, i_2, ..., i_m\}$ be a set of m items and C be a set of s labels. A subset $p = \{i_1, i_2, ..., i_k\} \subseteq I$ is called *itemset*, *pattern* or $k - pattern$ if it consists of k items. A pair $t_i = (x_i, y_i)$ where $x_i \subseteq I$ and $y_i \in C$ is called a *transaction*. A multiset of n transactions, denoted T, over I can be termed as a *transaction dataset*, denoted D. Let D_i be a subset of transactions corresponding to the class c_i. We have $D = D_1 \cup D_2 \cup ... \cup D_s$.

Given $p \subseteq I$, a set of transactions in D that contains p is denoted by $D(p)$. Similarly, a set of transactions in D_i that contains p is denoted by $D_i(p)$.

The *support* of pattern p over D is defined by:

$$sup(p, D) = \frac{|D(p)|}{|D|}$$

The support of pattern p over D_i is defined by:

$$sup(p, D_i) = \frac{|D_i(p)|}{|D_i|}$$

where $|.|$ denotes the cardinality of a set.

Definition 1.1 (Frequent pattern) Given a minimum frequency threshold α ($0 \leq \alpha \leq 1$), a pattern $p \subseteq I$ is frequent in D if its support value over D is no less than α: $sup(p, D) \geq \alpha$.

For illustration purpose, Fig. 1.1 presents a simple transaction dataset which contains two classes, each with 10 transactions (rows) and 15 items (columns). In this dataset, 4 example patterns can be observed: $p_1 = \{i_1, i_2, i_3\}$, $p_2 = \{i_5, i_6, i_7\}$, $p_3 = \{i_9, i_{10}\}$, and $p_4 = \{i_{12}, i_{13}, i_{14}\}$.

The supports of p_1, p_2, p_3, p_4 in D_1 are: $sup(p_1, D_1) = 0.6$, $sup(p_2, D_1) = 0.4$, $sup(p_3, D_1) = 0.2$, $sup(p_4, D_1) = 0.7$. Suppose $\alpha = 0.3$ is the minimum frequency threshold. p_1, p_2, p_4 are frequent in D_1 however p_3 is not frequent in D_1.

Local discriminative pattern mining problem: To evaluate the importance of a pattern in class-labeled datasets, algorithms often adopt some statistical measures such as growth rate [45], support difference [47] or mutual information [48]. These measures are defined over the supports of a pattern in the classes. For example, the growth rate of supports of pattern p in classes D_i and D_j, denoted GR, is defined by:

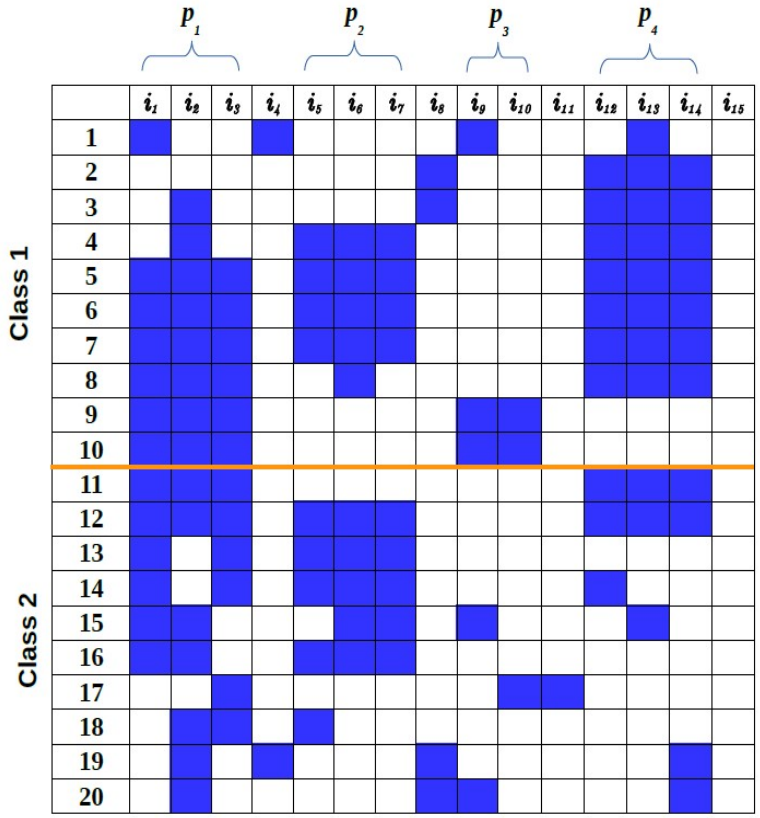


Figure 1.1: An example of a two-class labeled dataset

$$GR(p, D_i, D_j) = \frac{sup(p, D_i)}{sup(p, D_j)}$$

Definition 1.2 (Discriminative pattern) Let $f(p, D)$ be the discriminative measure function. Given a minimum discriminative threshold β, pattern p is discriminative if its discriminative power is no less than β: $f(p, D) \geq \beta$.

Taking again the data in Fig 1.1 for example. The GR values of p_1, p_2, p_3, p_4 are: $GR(p_1, D_1, D_2) = 3$, $GR(p_2, D_1, D_2) = 1$, $GR(p_3, D_1, D_2) = \infty$, $GR(p_4, D_1, D_2) = 3.5$. Suppose $\beta = 2$ is the minimum discriminative threshold. p_1, p_3, p_4 are discriminative patterns since their GR values are larger than β.

Given a dataset D with s classes and a minimum discriminative threshold β, the local discriminative pattern mining problem is to find a set of patterns, denoted R_{local}, which satisfies:

$$R_{local} = \{p \subseteq I | compare(f(p, D), \beta) \quad is \quad true\}$$

where f is the discriminative measure function. *compare* is true if the discriminative power of p satisfies the comparison constraints (such as $<, >, \leq, \geq, =, \neq$).

Global discriminative pattern mining problem: For the global case, some constraints are added on the set of output patterns in order to remove (most of) redundancies and increase interest of the patterns output.

Definition 1.3 (Pattern set) A pattern set, denoted g, is a subset of the powerset of R_{local}, $g \subseteq 2^{R_{local}}$.

In the global context, we want to find $g \subseteq 2^{R_{local}}$, preferably such as $|g| << |2^{R_{local}}|$. In order to do so, g should not have redundant patterns.

One approach is to consider *closed patterns* which are defined as follows:

Definition 1.4 (Closed pattern) Given a dataset D, a pattern p is closed in D if there doesn't exist any pattern q which contains p and has the same support as p in D.

The closed pattern approach can be used to remove a certain kind of redundancy of individual patterns. However, the number of generated patterns is still high.

More radical approaches to pattern set mining consider a scoring function $f : 2^{R_{local}} \longmapsto \mathbb{R}$ which gives a better score to smaller and more interesting pattern sets. The problem then becomes an optimization problem: find $g \subseteq 2^{R_{local}}$ that gives optimal f value. The problem is *NP-hard* thus finding good approximations is the only possibility.

In pattern set mining, finding suitable scoring functions is difficult. There exists various functions to evaluate the discriminative power of pattern sets such as *accuracy*, *w_accuracy* and *Laplace* [49]. However, these measures do not guarantee that the selected pattern sets are statistically significant.

Statistically significant discriminative pattern problem: In discriminative pattern mining, many patterns are tested for statistical significance, denoted *p_value*, by using Pearson's chi square test [50] or Fisher's extract test [51]. With a large number of tests, false positive errors may occur. Thus, we need to control this type of error by using hypothesis testing methods. Some methods such as Bonferroni's correction [52], Tarone's testability criterion [53], or Westfall-Young permutation procedure [54] are often used to correct the *significance level*, denoted δ. A pattern is statistically significant if its *p_value* is less than δ.

Definition 1.5 (Statistically significant discriminative pattern) Given a corrected significance level δ, a discriminative pattern p is statistically significant discriminative if its p_value is lower than δ.

Statistically significant discriminative pattern mining aims to find all patterns, denoted R_{stat}, whose p_value is below a corrected significance level.

$$R_{stat} = \{p \subseteq I | z(p, D) < \delta) \; and \; p \in R_{local} \; for \; some \; \beta, f, compare\}$$

where z is a function testing statistical significance.

1.3 Quality functions

To rank and select the patterns according to their potential interest to the users, several quality measures have been proposed. An appropriate measure allows the algorithms to reduce the search space as well as retrieve high quality results. In this section, we present some popular functions which are used to measure the interestingness of pattern in local and global levels. In addition, major statistical significance correction methods are also discussed.

1.3.1 Local measures

To evaluate the importance of discriminative patterns, the algorithms adopt some statistical measures which are generally defined based on the relative support of pattern in different classes. These measures can be defined either simply as the difference or ratio of the two supports [45, 47] or other variations, such as χ^2 [49] and mutual information [55]. As discussed in [19, 13, 48, 56], there exists a wide range of measures for evaluating the discriminative power of a pattern at the local level. In this section, measures for a pattern in two-class datasets are presented. These functions can be extended for multiple classes problems as discussed in [47].

Let D be a two-class dataset: $D = D_1 \cup D_2$. The presence and absence of a pattern in D_1 and D_2 can be tabulated by a contingency table as Table 1.1. A list of widely used measures for discriminative power are shown in Table 1.2. According to the properties of these statistical measures, a pattern with a higher value is considered as more discriminating.

Weighted Relative Accuracy ($WRAcc$) and generalization quotient (q_g) are widely used measures for subgroups discovery [62, 63, 57]. According to [15], algorithms employing the $WRAcc$ as the quality measure perform well compared with other algorithms. The reason is that this measure considers both the unusualness of the patterns and the size of the subgroups.

Table 1.1: Contingency table of a pattern in two-class dataset

	Presence	**Absence**	**Row total**
D_1	t_{11}	t_{12}	$\|D_1\| = t_{11} + t_{12}$
D_2	t_{21}	t_{22}	$\|D_2\| = t_{21} + t_{22}$
Column total	t_1	t_2	$\|D\| = \|D_1\| + \|D_2\|$

Table 1.2: Local discriminative power measures

No	**Name**	**Equation**	**Ref.**
1	Weighted Relative Accuracy	$WRAcc(p, D_1, D_2) = \frac{t_{11}+t_{21}}{\|D_1\|+\|D_2\|}(\frac{t_{11}}{t_{11}+t_{21}} - \frac{\|D_1\|}{\|D_1\|+\|D_2\|})$	[57]
2	Generalization quotient	$q_g(p, D_1, D_2) = \frac{t_{11}}{t_{21}+g}$, g is a user-defined parameter	[58]
3	Difference support	$DS(p, D_1, D_2) = \|sup(p, D_1) - sup(p, D_2)\|$	[47]
4	Growth rate	$GR(p, D_1, D_2) = \frac{sup(p,D_1)}{sup(p,D_2)}$	[45]
5	Odds ratio	$OR(p, D_1, D_2) = \frac{t_{11}t_{22}}{t_{12}t_{21}}$	[59]
6	Chi square	$\chi^2 = \sum_{i=1}^{i=2}\sum_{j=1}^{j=2}\frac{(t_{ij}-E_{ij})^2}{E_{ij}}, E_{ij} = \frac{\sum_{q=1}^{q=2} t_{iq}\sum_{q=1}^{q=2} t_{qj}}{\|D\|}$	[60]
7	Mutual information	$MI(p, D_1, D_2) = \sum_{i=1}^{i=2}\sum_{j=1}^{j=2}\frac{t_{ij}}{\|D\|} log\frac{t_{ij}/\|D\|}{t_i\|D_j\|/\|D\|^2}$	[55]
8	Information gain	$IG(p, D_1, D_2) = sup(p, D_1)(log\frac{sup(p,D_1)}{sup(p,D)} - log\frac{\|D_1\|}{\|D\|})$	[60]
9	Gini index	$GI(p, D_1, D_2) = \frac{1}{2}\sum_{i=1}^{2} sup(p, D_i)(1 - sup(p, D_1))$	[61]
10	supMaxPair	$supMaxPair(p, D_1, D_2) = sup(p, D_1) - max_{\alpha\subseteq p}(sup(\alpha, D_2))$, $(\|\alpha\| = 2)$	[24]

Other measures such as difference support (DS) [47] and growth rate (GR) [45] measure the discriminative power of a pattern based on its supports in different classes. In particular, DS measures the difference of supports between the two classes. On the other hand, GR measures the ratio of supports between the two groups. These measures are also demonstrated as equivalent with risk ratio (RR) and absolute risk reduction (ARR) which are often used in GWAS to evaluate the association strength of biological patterns with an interesting group of individuals [59]. Similarly, odds ratio (OR) is often adopted to evaluate the discriminative power of patterns [57]. It calculates the ratio of odds of a pattern in one class to that in the other class. This measure is also known as a gold standard for measuring the association strength in GWAS [64]. In practice, the combination of DS, GR, and OR is efficiently applied to assess the importance of risk factor patterns. For example, by using a combination of these measures, it has been shown that the task of cancer classification is performed more accurately than with an approach based on Naive Bayesian classifier [59].

Another important group of measures such as Chi square (χ^2), Mutual information (MI), Information gain (IG), and Gini index (GI) are used to evaluate the significant difference of frequencies of a pattern in two classes [48, 61]. More specifically, χ^2 is used to determine whether there is a significant difference between the frequencies of a pattern in two groups of subjects. MI and IG are functions based on information theory. They measure the difference of frequencies of a pattern between two classes [48]. Similarly, Gini index (GI) is used to measure the inequality of a pattern in two classes. These measures can be used in branch-and-bound [65, 66] and constraint programming algorithms [60] to discover discriminative patterns.

On the other hand, $supMaxpair$ [24] is an extension of DS. They form a family of monotonous interestingness measures for discriminative power. It can be used to prune the search space in an Apriori framework and mine discriminative patterns with very low frequency in high dimensional and dense datasets.

As discussed above, there is a wide range of measures. However, using these measures for discovering discriminative biological patterns remains challenging.

The first problem relates to the strategies for mining discriminative patterns. None of these statistical metrics are anti-monotone [47, 61]. It means that the discriminative power of a pattern is not correlated to the discriminative power of its sub-patterns. This considerably limits the opportunities for pruning the search space, compared to a traditional pattern mining setting.

Second, for each measure, users have to choose an appropriate threshold to evaluate the significance of patterns. This is an extremely difficult task, and incorrect choice of thresholds may have an important impact. If the thresholds are too loose, the pattern mining algorithms will generate many patterns of limited interest. On

the other hand, some interesting patterns will be lost if the thresholds are too restrictive.

Third, biological patterns are more complicated than general patterns since they are related to natural properties. Thus, it is difficult to directly use discriminative power measures to assess the importance of biological patterns. In fact, researchers often use additional techniques or combine some measures to evaluate the statistical significance of biological patterns. In practice, one usually combines it with other measures to evaluate the interesting of biological patterns. Specifically, OR and χ^2 are used in [24] to evaluate the significance of co-occurrence for genes expressions. Similarly, OR, χ^2 and p_value are adopted in [16] to evaluate association strength between high-order SNP combinations and disease. In addition, the combination of OR, RR, and ARR is used in [59] to evaluate the significance of risk factor patterns.

1.3.2 Global measures

Instead of evaluating individual discriminative patterns, the discriminative pattern set mining techniques use global constraints to assess the set of patterns [49].

Many functions exist to measure the interestingness of a pattern set. Some popular measures are listed in Table 1.3. To illustrate these measures, we use the following notations: g is a pattern set consisting of s patterns where p_i is the i^{th} pattern in this set. $D(p)$ and $D_k(p)$ are the set of transactions that contain pattern p in D and D_k respectively. Similarly, we use $D(g)$ and $D_k(g)$ to denote the set of transactions that contain pattern set g in D and D_k respectively. A pattern set can be interpreted as a disjunction of the individual patterns. Thus $D(g)$ and $D_k(g)$ can be computed by taking the union over the individual transaction sets.

$$D(g) = D(p_1) \cup D(p_2) \cup ... \cup D(p_s)$$

$$D_k(g) = D_k(p_1) \cup D_k(p_2) \cup ... \cup D_k(p_s)$$

itemsOverlap is used to measure the similarity of discriminative patterns with regard to the set of items that are included in these discriminative patterns while *transOverlap* is applied to calculate the similarity between the transaction sets that contain the discriminative patterns [67]. The discriminative pattern sets with smaller *itemsOverlap* or *transOverlap* values are better because they contain fewer and less redundant discriminative patterns.

For example, given a set of 5 discriminative patterns: $p_1 = \{a, b, c, d\}$, $p_2 = \{a, b, c, f, g\}$, $p_3 = \{a, b, d, h\}$, $p_4 = \{d, g, i, j, k\}$, $p_5 = \{i, j, k, h\}$.

Consider the following pattern sets: $g_1 = \{p_1, p_2, p_3\}$, $g_2 = \{p_4, p_5\}$, $g_3 = \{p_1, p_4\}$.

Table 1.3: Global discriminative quality measures

No	Name	Equation	Ref.
1	Item overlap	$itemsOverlap(g) = \frac{2}{s(s+1)} \sum_{i=1}^{s} \sum_{j=i+1}^{s} \lvert p_i \cap p_j \rvert$	[67]
2	Transaction overlap	$transOverlap(g) = \frac{2}{s(s+1)} \sum_{i=1}^{s} \sum_{j=i+1}^{s} \lvert D(p_i) \cap D(p_j) \rvert$	[67]
3	Area	$tile(p_s, D) = \{(T, i_k) \lvert T \in D(p_i), i_k \in p_s\}$ $area(g) = \lvert tile(p_1, D) \cup tile(p_2, D) \cup ... \cup (p_s, D) \rvert$	[49, 68]
4	Overall coverage	$COV(g) = \frac{\lvert D_1(g) \cup D_2(g) \rvert}{\lvert D \rvert}$	[15]
5	Accuracy	$accuracy(g) = \lvert D_1(g) \rvert - \lvert D_2(g) \rvert$	[49] [69]
6	Weighted accuracy	$w\_accuracy(g) = \frac{\lvert D_1(g) \rvert}{\lvert D_1 \rvert} - \frac{\lvert D_2(g) \rvert}{\lvert D_2 \rvert}$	[49]
7	Laplace	$Laplace(g) = \frac{\lvert D_1(g) \rvert + 1}{\lvert D_1(g) \rvert + \lvert D_2(g) \rvert + 2}$	[49]

The *itemOverlap* values of these pattern sets are:
$itemOverlap(g_1) = \frac{2}{3(3+1)} (3 + 3 + 2) = 1.45$,
$itemOverlap(g_2) = \frac{2}{2(2+1)} (3) = 1$,
$itemOverlap(g_3) = \frac{2}{2(2+1)} (1) = \frac{1}{3}$.
In this case, g_3 is better than g_1 and g_2 since it contains fewer redundant discriminative patterns.

For the other functions, the goal is to maximize the score returned by the function. The *area* of a pattern set is estimated by counting all the *tiles* covered by the individual patterns [49, 68]. More specifically, a *tile* of a pattern is the set of all tuples $(t, i) \in D$ $(t \in T, i \in I)$ that are covered by the pattern. The area of a single pattern is the number of tuples that are covered in the *tile*: $area(p) = \lvert tile(p) \rvert \leq \lvert I \rvert . \lvert T \rvert$. Overall coverage ($COV$) is defined as the fraction of transactions covered by a pattern set [15]. This measure gives us the proportion of transactions that are covered by the discovered pattern set. On the other hand, *accuracy* is defined as the dif-

Table 1.4: Statistical significance correction methods

No	P_value correction methods	Ref.
1	Bonferroni's correction	[47, 71]
2	Tarone's testability criterion	[72, 73, 70]
3	Westfall-Young permutation	[33, 31]
4	LAMP	[32, 74]

ference of number of transactions that are covered by a pattern set in two classes. Similarly, weighted accuracy (*w_accuracy*) and *Laplace* are also computed based on the number of transactions that are covered by a pattern set.

There are many measures to assess the global interestingness of a pattern set. These functions can be used to evaluate the redundancy, diversity or discrimination of pattern sets. In practice, providing the constraints guaranteeing that the patterns are globally significant is a difficult task.

1.3.3 Multiple hypothesis testing

To test the statistical significance (*p_value*) of a discovered discriminative pattern, different mathematical methods such as Fisher's exact test or Pearson's chi-square test are used. In practice, discriminative pattern mining algorithms often generate a large number of patterns due to the high-dimension of dataset as well as the combinatorial nature of the task. Accordingly, a huge amount of statistical tests is performed to evaluate the significance of discovered patterns. With the large number of tests, the probability of false discovery increases. Thus the calibration of significance level in each test is required to control the total error rate of false positives by multiple testing correction procedure [70]. A list of the popular *p_value* statistical significance correction procedures is given in Table 1.4.

Bonferroni correction procedure [52] is a simple and widely used theoretical approach. Given I items, we must perform $M = 2^I - 1$ association tests, one for each possible pattern, to measure the significance of patterns. Let α be a significance level. Bonferroni correction controls the probability of at least one false discovery, called family wise error rate (FWER), by adjusting the significance level to $\delta = \alpha/M$. A pattern is statistically significant if its *p_value* is below the adjusted significance level δ. When all possible patterns are checked, the number of tests increases exponentially due to the amount of items, and δ becomes a very small value. But computing all M tests is very time-consuming.

To overcome this problem, Tarone's testability criterion [73], an improvement of Bonferroni correction procedure, is proposed. The key of this strategy is that only a subset of M tests, called *testable hypotheses*, can reach the significance level. Thus

instead of checking all M tests, one can safely prune the tests which are not testable hypotheses without affecting the probability of reporting FWER. This method is successfully applied in data mining to find significant combination of transcript factors in gene regulatory network [32] and search significant subgraphs [70].

Another strategy is the Westfall-Young permutation procedure [54] which generates a null distribution from thousands of randomly permuted datasets, and determines δ based on the distribution. According to [31], this method has higher detection power than Bonferroni correction and its improvements [75, 76]. Westfall-Young procedure has been successfully used in data mining to find statistically significant discriminative patterns [31, 33].

Recently, a novel method for *p_value* correction, named LAMP [32] has been proposed. It is based on frequent itemset mining to exclude meaningless infrequent itemsets which never reach the significance level. This method adjusts *p_value* much more accurately and is less cost-consuming than Bonferroni's test procedure [32, 74].

Among these procedures, Tarone's testability criterion, Westfall-Young and LAMP can be directly used in the pattern mining process [31, 74, 33, 72] to find the statistically significant discriminative patterns in one stage. These studies are the first approaches that successfully combined *p_value* correction procedures into the pattern mining process.

1.4 Algorithms and software frameworks

Many algorithms and software frameworks have been investigated to efficiently discover discriminative patterns. The various strategies can be classified into several categories such as local discriminative pattern mining [24, 62, 77], global discriminative pattern mining or discriminative pattern set mining [78, 69, 79, 49, 80] and statistically significant discriminative pattern mining [71, 31, 32, 74, 33, 72]. In the local discriminative pattern mining context, every pattern is separately evaluated under no consideration of the relationships between each other. The disadvantage of this approach is that it outputs a lot of patterns of mixed interest (many redundancies). For the global case, some constraints are added on the set of output patterns in order to remove (most of) redundancies and increase interest of the patterns output. A difficulty is to provide the constraints guaranteeing that the patterns are interesting. Thus, adding statistical constraints which come with guarantees well understood by the biologists (and many other practitioners) is essential. To address this problem, statistically significant discriminative pattern mining is proposed. This approach aims to detect patterns which are at the same time discriminative and statistically significant by using statistical significance correction procedures.

Table 1.5: Local discriminative pattern mining algorithms

Search strategy	**Algorithms**	**Quality measure**	**Ref.**
Exhaustive	Apriori-SD	$WRAcc$	[63]
	SMP	$supMaxPair$	[24]
	DDMiner	IG	[44]
Top-ranking	CIMCP	χ^2, IG, GI, $Fisher\ score$	[60]
	SSDP	q_g, $WRAcc$, DS	[81, 82]
	Top-K Minimal Jumping Emerging Patterns	GR	[83]
Heuristics	SD	q_g	[58]
	CN2-SD	$WRAcc$	[57]
	SDIGA	$WRAcc$	[43]
	NMEEF-SD	$WRAcc$	[84]
	GAR-SD	$support$, $confidence$, $significance$	[85]

In general, to discover discriminative patterns, the algorithms perform the search on a dataset and use some statistical measures which are discussed in the previous section to rank and select the interesting patterns. In recent years, a wide variety of software frameworks and algorithms have been investigated to tackle this issue. It is out of the scope of this section to present exhaustively the algorithms. Rather, we present a selection of state-of-the-art algorithms that have been successfully used in bioinformatics. In particular, local and global discriminative pattern mining with some popular search strategies such as exhaustive, top-ranked and heuristic are firstly discussed. Next, some successful approaches for directly mining statistically significant discriminative patterns are presented.

1.4.1 Local discriminatve pattern mining

Local discriminative pattern mining algorithms aim to find and evaluate individual patterns separately. They often adopt some quality functions which are displayed in Table 1.2 to measure the significance of patterns. If the score of a pattern satisfies a given threshold it is considered as a discriminative pattern. A large number of algorithms and software frameworks have been developed for this task. Table 1.5 and Table 1.6 show some popular algorithms and software respectively.

Depending on search strategies, local discriminative pattern mining algorithms can be classified into 3 groups: exhaustive, heuristic and top-ranked.

Table 1.6: Software frameworks for local discriminative pattern mining

Software	**License**	**Source**
VIKAMINE	GNU	http://www.vikamine.org
Orange	GPL	https://orange.biolab.si/
Cortana	Free download	http://datamining.liacs.nl/cortana.html
KEEL	GPLv3	http://www.keel.es
RapidMiner	Commercial	https://rapidminer.com

Exhaustive search aims to discover the complete set of discriminative patterns which satisfy a given threshold. These algorithms usually adopt some classic search strategies such as breadth-first search (BFS) and depth-first search (DFS) [42]. To find patterns with size of k, a BFS algorithm such as Apriori [39] starts with patterns of size $k = 1$. Then the patterns of larger size are created based on the set of patterns which are generated in the previous step, i.e. patterns of size k are generated from the set of patterns of size $k-1$. This approach requires a lot of memory for intermediate computation, and usually runs out of memory. In addition, it cannot benefit from some important optimization techniques such as dataset reduction. On the other hand, DFS algorithms start to search patterns with individual items. For a selected item i, all patterns that contain i are recursively generated. This process is repeated for all items. This DFS strategy is usually the basis of efficient algorithms such as FP-Growth [40], LCM [86].

These search strategies are widely applied in discriminative pattern mining. For example, Apriori-SD algorithm [63] discovers discriminative patterns based on Apriori framework. First, it finds all patterns satisfying the support threshold. Then patterns are evaluated for discriminative scores and selected according to their $WRAcc$ score in a post-processing step. Similarly, SMP algorithm [24] uses $supMaxPair$, a monotonic measure, in an Apriori framework to exhaustively mine discriminative patterns with low support. On the other hand, SD-Map [62], a fast algorithm for exhaustive discriminative pattern discovery, applies FP-growth method to detect association rules with adaptations for the discriminative pattern mining task. Similarly, BSD [87], an algorithm for fast discovery of relevant subgroups, exhaustively discovers discriminative patterns based on a branch-and-bound strategy.

Among these approaches, algorithms based on DFS are more efficient than those based on BFS. DFS approaches allow efficient pruning strategies to be applied to reduce the search space. In addition, DFS can be employed for directly mining discriminative patterns. For example, the algorithm in [88] performs a recursive search on a search tree to discover discriminative patterns which satisfy a minimum support and information gain value thresholds. On the other hand, DDPMine [44] performs a

branch-and-bound search for directly mining discriminative patterns without generating the complete pattern set. These approaches outperform the two-step methods [65, 89, 90]: first generate a set of frequent patterns, then apply a statistical measure to evaluate and select the discriminative patterns.

The exhaustive algorithms guarantee to discover a complete set of discriminative patterns in a given dataset. However, they can not be used for high-dimension datasets such as biological datasets which usually have some hundred thousands of items. To overcome this limitation, heuristic methods have been investigated to discover a good enough but not necessary optimal result. This approach trades off between execution time and the quality of the pattern set.

The common approach for heuristic search of discriminative patterns is beam search [57, 58]. To conduct the mining of discriminative patterns, the beam search algorithms use an initial number of discriminative patterns which is determined by a beam size parameter. For each iteration, new patterns are generated from the set of candidates which have been selected in the previous step. A typical algorithm which uses beam search strategy is CN2-SD [57]. To start the search, CN2-SD considers the highest quality items (according to their discriminative power and minimal support threshold) as singleton discriminative patterns. For each iteration, the least relevant patterns are replaced by the most relevant ones with larger sizes. The search stops when the patterns in the current beam can not be replaced by more relevant patterns.

On the other hand, other algorithms based on genetic fuzzy systems have also been developed to tackle the task of heuristic discriminative patterns discovery [43, 84, 85]. These approaches are designed to find the most important rules of the subgroups on various quality measures based on evolutionary computing. In comparison with beam search, these approaches are more efficient [91]. They can be employed to tackle high dimensional datasets [81].

Another search strategy in local discriminative pattern mining is top-ranking. The idea is, given a small integer k, to output only the first k patterns according to statistical significance.

According to the literature, the first approach for discovering top-k discriminative pattern has been studied in [83]. It enumerates top-k discriminative patterns, called top-k minimal jumping emerging patterns, based on CP-Tree [92]. To prune the search space, the algorithm uses the minimal support of patterns.

Another approach for top-k discriminative pattern mining is based on constraint programming [60]. In this study, the task of mining top-k discriminative patterns is modeled as a constraint programming problem. Based on specific properties of statistical measures such as Fisher score, information gain, Gini index, or χ^2 the algorithm discovers k patterns with regard to a given constraint.

Last, approaches based on evolutionary algorithms have also been investigated to mine top-k discriminative patterns [82, 81]. These approaches have been successfully applied to discover top-k discriminative patterns in high dimension datasets which are difficult to analyze with traditional methods.

Overall, local discriminative pattern mining has been tackled with various search strategies, and has demonstrated its usefulness in many applications. However, the major drawback of this approach is the amount of generated patterns which is often very large. It is complicated to use directly the patterns without post processing. Moreover, many redundant discriminative patterns are included in the result since the algorithms evaluate patterns independently.

1.4.2 Global discriminative pattern mining

As discussed in the previous section, most existing local discriminative pattern mining algorithms often face the problem of generating a huge number of patterns which include many redundant ones. These patterns might be covered by a similar set of transactions or include an equivalent set of items. In practice, discovering all these redundant patterns is time-consuming. In addition, it is complicated to interpret the results. Thus, searching a condensed and non-redundant pattern set is a critical task. This task is often referred as global discriminative patterns or discriminative pattern set mining in data mining.

The objective of global discriminative pattern mining is to keep only some representative discriminative patterns of all the equivalent ones to reduce the degree of redundancy and increase the ease of understanding of discriminative patterns in real-world applications. Instead of evaluating individual patterns separately, global discriminative pattern mining algorithms often adopt global functions which are shown in Table 1.3 or also use local functions which are illustrated in Table 1.2, to rank and select a set of global interesting patterns according to the whole pattern set. Some popular global discriminative pattern mining algorithms with their quality measures are listed in Table 1.7.

The global discriminative patterns can be discovered by different ways. The first approach is in two steps: in the first step, all discriminative patterns which satisfy given constraints, are discovered by a local discriminative pattern mining algorithm. Then, in the second step, patterns are post-processed to find discriminative pattern sets. The post-processing can be conducted by exhaustive or approximate search depending on the user's objective. However, with a huge amount of generated patterns, greedy search is often performed to compute the pattern sets [69, 96].

The other strategy for discovering global discriminative patterns is to perform heuristic search. The most popular approach for this strategy is beam search [78, 93]. This technique can be briefly described as follow: the algorithm finds the

Table 1.7: Global discriminative pattern mining algorithms

Search strategy	**Algorithm**	**Measure**	**Ref.**
Heuristic	GSD	$WRACC$	[93]
	SSDS	$WRACC$	[78]
	BSD	q_g	[87]
	CDPM	$support$, DS	[80]
Top-ranking	Delta-relevant patterns mining	$WRACC$	[94]
	RP-growth	χ^2, IG	[95]
	K-pattern set mining under constraints	$accuracy$, $w\_accuracy$, $Laplace$	[49]

most interesting (local) discriminative patterns, and then candidate pattern sets are selected according to current beam which contains these patterns. Subsequently, based on the overall statistical significance, the most significant pattern sets are selected. This process is continued until no more candidate pattern sets can be discovered. Pattern sets with a strong discriminative power are obtained in the final result.

Additionally, the problem of discovering pattern sets can be formulated as a global optimization problem with user-specified significant constraints. For instance, the approach in [49] imposes significant constraints on the whole pattern set to find k-pattern sets with the strongest discriminative power in one step.

Moreover, other definitions and strategies have also been investigated to tackle the problem of redundant patterns. The simplest method employed is the *closeness* constraint to find closed discriminative patterns [96, 97]. The idea of these approaches is equivalent to closed frequent itemsets mining which is widely used in data mining [98].

Some other studies proposed *relevant pattern* concept to mine non-redundant patterns in class labels datasets [87, 94, 95, 99]. The relevance between two discriminative patterns is defined based on the relationship of the sets of transactions that contain these discriminative patterns in the two classes. For example, let $D_1(p)$ and $D_2(p)$ be the sets of transactions that contain pattern p in D_1 and D_2 respectively. Similarly, let $D_1(q)$ and $D_2(q)$ be the sets of transactions that contain pattern q in D_1 and D_2. Pattern p is relevant with another pattern q if $D_1(q) \subseteq D_1(p)$ and $D_2(p) \subseteq D_2(q)$. Or we can also say that q is *irrelevant* with respect to p. If q is irrelevant with respect to p, then the power of a discriminative pattern q is lower or equal to the power of discriminative pattern p for any quality function satisfying a

given set of axioms [87]. Thus, the relevant pattern mining algorithms discover non-redundant patterns by filtering out irrelevant patterns [99, 95]. These methods can be applied to find a set of non-redundant pattern set. However, these approaches do not consider the relationship between the discriminative power of a pattern and scores of its subsets. Thus, they can not remove redundant patterns caused by their subsets.

To address this problem, a new concept, named *conditional discriminative pattern*, has been investigated [80]. A conditional discriminative pattern is defined based on the discriminative power of a pattern and the discriminative power of its subsets. Specifically, let p be a k-pattern then p has $2^k - 2$ possible sub-patterns. Each sub-pattern is covered by an equivalent transaction set. *Local significance* of k-pattern is defined as the smallest value of discriminative powers of (k-1)patterns. *Global significance* is defined as the discriminative power of p which is computed based on its original transaction set. A k-pattern is conditional discriminative if it satisfies significance thresholds on both local and global levels. For example, suppose p is a 2-pattern which has two sub-patterns: a and b; α and β are the global and local significance thresholds respectively. p is conditional discriminative pattern if its discriminative power is not less than α and discriminative power of a and discriminative power of b are not less than β. To discover conditional discriminative patterns, the algorithm builds data on a tree structure, then adopts DFS strategy to traverse and produce patterns which satisfy both local and global significance thresholds. Experimental results show that an approach based on conditional discriminative pattern efficiently eliminates redundant patterns whose discriminative power mainly comes from their sub-patterns.

In short, compared with the local discriminative patterns discovery the task of global discriminative patterns mining is more complicated and time-consuming. The exhaustive search is infeasible since the number of sets of local discriminative patterns is enormous. Thus, to trade off between the performance and the quality of non-redundant pattern set most approaches adopt heuristic strategies.

1.4.3 Statistically significant discriminative pattern mining

Beside the computation and redundancy problems, a perhaps even more important challenge in discriminative pattern mining algorithms is multiple hypothesis testing. The available algorithms often discover a huge number of patterns which should be tested for statistical significance, p_value. For example, given I items, we must perform $2^I - 1$ association tests, one for each possible pattern, to measure the significance of the pattern set. With the enormous number of tests, the probability of some patterns deemed to be significantly associated with class membership by mistake is high. This probability is referred as *false positives* or *family wise error*

Table 1.8: Statistically significant discriminative pattern mining algorithms

Algorithm	**Multiple hyphothesis testing**	**Ref.**
FastWY	Westfall-Young procedure	[31]
Westfall-Young light	Westfall-Young procedure	[33]
LAMP	LAMP	[32]
New LAMP	LAMP	[74]
FACS	Tarone's testability criterion	[72]

rate (FWER). Thus calibrating the significance level in each test is required to control the total error rate of false positives by multiple testing correction procedures which are given in Table 1.4. This task is time-consuming since a large number of validations are computed.

Searching for statistically significant itemsets has been widely studied [71, 100, 101]. A naive approach to discover statistically significant patterns is a two-step strategy: first find all discriminative patterns which satisfy a given threshold, then conduct permutation test to choose the significance level δ and select patterns which have p_value lower than this threshold [71]. This method is effectively employed to find a set of statistically significant discriminative patterns. However, it can deal with only very small datasets since the number of tests (patterns) scales combinatorially.

To address this problem, some methods allowing to combine discovering patterns and multiple hypothesis testing in one stage are proposed [31, 32, 74, 33, 72]. These algorithms are summarized in Table 1.8.

Among them, FastWY [31] is an early approach which takes the dependence between test statistics in pattern mining into account. In this study, Fisher's exact test and Westfall-Young procedure are used to test the statistic and correct the significance level respectively. FastWY discovers all statistically significant discriminative patterns in three steps. Step 1: estimate a null distribution. Step 2: calculate the adjusted significance level δ. Step 3: generate patterns whose p_values are lower than δ. To find the adjusted significance level δ for keeping $FWER \leq \alpha$, FastWY uses randomly permuted datasets. Specifically, for each permuted dataset, FastWY uses a branch-and-bound algorithm combined with the lower bound and monotonicity properties of p_value to discover the minimum p_value among all of the patterns. When a minimum p_value is retrieved, it can be used as the adjusted significance level for multiple testing. FastWY is not efficient since it uses randomly permuted datasets to find adjusted significance level. This task has to be repeated N times if N permutations are required to calculate the FWER. To address this limitation, Westfall-Young light algorithm [33] uses an incremental search strategy to find all

significant patterns in one instead of N times without any extra memory requirements. Similar to FastWY, Westfall-Young light uses Westfall-Young procedure in a branch-and-bound algorithm to find and correct the significance of patterns. Experimental results show that Westfall-Young light algorithm outperforms FastWY in both execution time and memory usage.

Recently, a novel method for *p_value* correction, named $LAMP$, based on frequent itemset mining to exclude meaninglessly infrequent itemsets which never reach the significance level, has been proposed. This method adjusts *p_value* much more accurately and is less cost-consuming than Bonferroni's test procedure [32]. LAMP is adopted in [74] to directly discover statistically significant discriminative patterns. In this study, LAMP condition is demonstrated as a kind of monotonic function. This property is efficiently used in frequent itemset mining to explore all statistically significant discriminative patterns satisfying a given threshold function. This algorithm allows to discover statistically significant discriminative patterns in a short time even for very large-scale databases.

The above approaches evaluate the statistical significance of patterns by Pearson's chi square test or Fisher's extract test which does not consider the conditional association between discriminative patterns and the target class. As a result, many false discoveries might occur due to unaccounted confounding effects. To address this problem, a novel algorithm, named FACS [72], which applied Tarone's testability criterion in Cochran-Mantel-Haenszel (CMH) test [102] is proposed. In this study, Tarone's testability criterion is employed to correct the statistical significance level. The CMH test is used to test conditional association between discriminative pattern and class label. This is the first algorithm that bridges the gap between Tarone's testability criterion and the CMH test. FACS includes two main steps: compute Tarone's corrected significance threshold δ_{tar} and retrieve all patterns whose *p_values* (estimated by CMH test) are below δ_{tar}. To compute δ_{tar}, FACS uses a branch-and-bound algorithm which allows to directly apply Tarone's testability criterion to the CMH. Experimental results show that this approach outperforms the state-of-the-art significant discriminative patterns mining such as LAMP [32] and BONF-CMH [74].

In short, the approaches that combine test statistic in pattern mining have been successfully applied for discovering statistically significant discriminative patterns. They not only generate a limited number of patterns but also correct the significance level of results.

Table 1.9: An example of SNPs dataset

Individual	**SNP**					**Label**
	SNP1	SNP2	SNP3	SNP4	SNP5	
1	AT	GC	AT	GC	AG	Case
2	AT	GC	AA	CC	AG	
3	AT	CC	AT	GG	AG	
4	AA	GG	AA	GC	AA	Control
5	AT	GG	AT	GC	AA	
6	TT	GC	AA	CC	GG	

1.5 Applications in bioinformatics

Discriminative pattern mining is very important for bioinformatics. Although this thesis is focused on GWAS, the discriminative pattern discovery techniques which are investigated in this thesis are also effectively applied to other bioinformatics problems. In this section, we present some major bioinformatics problems which have been successfully tackled by discriminative pattern mining techniques. In particular, the tasks of identifying high-order SNP combinations, discovering differential gene expressions, detecting phosphorylation motifs and mining regulatory motif combinations are focused.

1.5.1 High-order SNP combinations identifying

Single-nucleotide polymorphism (SNP) is a variation in a single nucleotide that occurs at a specific position in the genome [1]. These SNPs may be associated with the increase or decrease of an individual's risk of getting a disease or benefitting from a particular therapy. To find SNPs associated with a disease, genome-wide association studies (GWAS) compare the SNPs of two groups: case group consists of patients which are affected by a disease and control group consists of healthy people without a disease. Single SNPs or combinations of SNPs are correlated with a disease if they occur more frequently in the case group than in the control group. Once new genetic associations are identified, they can be used to develop better strategies to detect, treat and prevent the disease [2]. Thus identifying SNP combinations associated with diseases is very important task in bioinformatics. Table 1.9 shows a simple SNP dataset which includes 6 individuals of two groups (case and control), each individual contains 5 SNPs.

Many approaches have been investigated for detecting the interactions of genetic variants. Some methods uses statistical models such as Logistic Regression [5], Bayes model [6] while others adopt machine learning techniques such as support vector

machine [7], neural networks [8], decision trees [9], and random forests [10]. These approaches have been effectively applied to discover SNPs interactions in GWAS. However, they are used to tackle only small biological datasets and detect only single or two-locus interactions [11, 12].

To address these limitations, various solutions based on discriminative pattern mining have been investigated. Local pattern mining algorithms can be directly applied to find high-order SNP combinations associated with disease [16, 21, 103, 104]. These studies adopt exhaustive search strategy to discover all possible discriminative patterns. In addition, to assess the association strength between SNP combinations and disease, local quality measures such as OR, χ^2 are used. Experimental results show that these approaches can discover many interesting SNP patterns. However, the exhaustive search strategies which are used in these studies are only suitable to deal with small variant datasets with some hundreds or thousands of SNPs.

To work with larger SNP datasets, other methods use heuristic strategy to discover a good enough but not necessarily optimal result [105, 21, 20]. Although the performances of these methods are better than the exhaustive search approaches, risk of missing interesting SNP combinations is increased.

In addition, common step-wise approaches [23, 106, 107, 108] have also been investigated to conduct the mining of SNP patterns. These methods include two steps: filtering step and searching step. During the first step, the interesting SNPs with regard to some conditions are selected. Then these SNP candidates are used in the second step to find combinations using specific discriminative pattern mining algorithms. Experimental results demonstrate that these approaches are efficient. Many interesting SNP patterns are discovered in an acceptable execution time.

In short, searching high-order SNP combinations in large genetic variant datasets is a challenge. The exhaustive search is infeasible while heuristic and step-wise approaches increase risk of missing importance patterns. Thus, methods capable of efficiently searching high quality SNP combinations should be considered to pursue this issue.

1.5.2 Differential gene expressions discovering

Discovering and visualizing differential gene expression groups plays an important role in bioinformatics [25, 109, 110]. These compounds of gene expressions can be used to build disease diagnosis or treatment systems [26, 27, 17, 111]. With the development of AND chip technologies, thousands of gene expressions can be measured in an experiment. Thus, searching combinations of gene expressions in high-dimension datasets is a computational challenge.

The gene expression data is presented as a matrix in which rows correspond to the set of genes and columns present the normal cells or disease cells. The value

Table 1.10: An example of gene expression dataset

Genes	**Cell types**					
	Cancer	Cancer	Cancer	Normal	Normal	Normal
gene_1	0.1	0.2	0.3	0.7	0.9	0.3
gene_2	0.1	0.2	0.3	0.4	0.5	0.6
gene_3	-0.70	-1.1	-0.2	-.90	-0.55	-0.32
gene_4	3.25	4.15	5.25	0.50	0.75	0.83
gene_5	-2.05	1.1	-2.2	4.0	-5.5	0.3
gene_6	1.0	1.1	1.2	1.0	1.5	1.3

at position (i, j) presents the expression level of equivalent gene i^{th} and cell j^{th}. A simple example of gene expression data is shown in Table 1.10.

The task of discovering differentially expressed genes requires to find groups of genes that are constrained to specific intervals of gene expression levels. Such patterns occur highly frequently in one class of cells but less in another class of cells or are only present in one class of cells but do not occur in the other cells. For example, gen_1 and gen_2 have values of gene expression ranging from 0.1 to 0.3 and they occur 100% in the cancer cells but are absent in the normal cells. In this situation, the combination of gen_1 and gen_2 is considered as an interesting gene expression pattern.

The problem of identifying differentially expressed genes can be handled by discriminative pattern discovery methods [25, 26, 27, 17]. In order to perform this task, the gene expression dataset is transformed into the input of a discriminative pattern mining algorithm by considering each gene as an item and each cell type as a transaction. More importantly, discretization methods have to be used to partition gene expression levels into a number of suitable intervals [25, 26, 17]. The reason is that gene expression are continuous values thus they cannot be used directly in discriminative pattern mining algorithms. To test the significance of gene combinations associated with the interesting class of cells, one can directly adopt the local quality measures. For example, to measure the discriminative power of gene expression combinations, the studies in [25, 26, 17] directly use GR while [27] adopts p_value which is computed by Fisher's exact test.

In brief, discriminative pattern mining algorithms can be adopted to search differential gene expression combinations. However, these methods have to discretize the value of gene expression level to be suitable for the available discriminative pattern mining algorithms. This problem calls for approaches which can analyze continuous value.

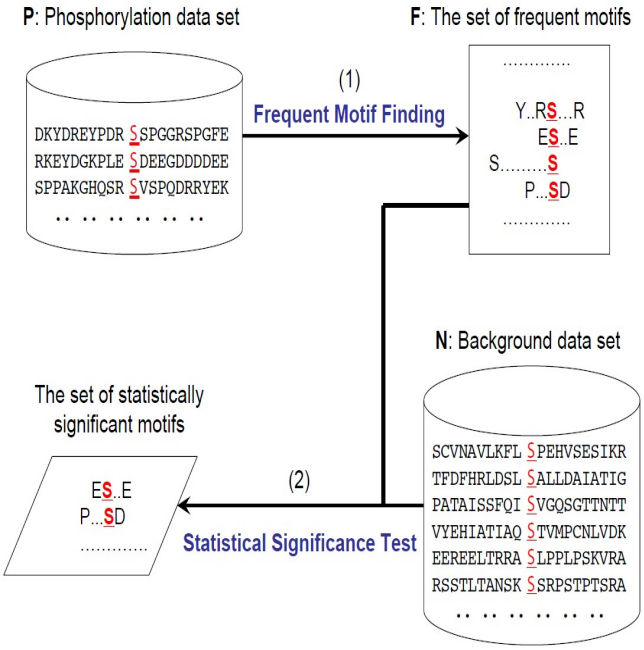


Figure 1.2: Phosphorylation motifs discovery process [29]

1.5.3 Phosphorylation motifs detection

The goal of discovery of phosphorylation motifs is to find a set of motifs that occur more frequently in the phosphorylated peptide set, called foreground (P), than in the unphosphorylated peptide set, called background (N) [28]. An example of phosphorylation motifs discovery is demonstrated in Fig 1.2. According to this objective, the discovery of motif combinations is equivalent to discriminative patterns mining. Upon this issue, the motif datasets can be considered as the inputs of discriminative pattern discovery methods where the property of (un)phosphorylation are considered as class labels; the given peptides set correspond to the transactions; and the phosphorylation motifs as the set of items. The interesting motif combinations with statistical significance can represent the differences between these two classes, which are equivalent to the discriminative patterns.

A wide range of effective approaches has been proposed for phosphorylation motif discovery [28, 29, 30, 18, 112, 113]. Among them, Motif-All [29] and C-Motif [18] are the two studies which use discriminative pattern mining techniques to tackle the problem of discovering phosphorylation motif combinations. Motif-All [29] uses two-step approach for discovering statistically significant motifs. In the first step, Motif-

All uses the support enumeration to mine a set of frequent motifs as candidates in the foreground. Then it adopts statistical significance measure to rank and select the significant ones in the second step. On the other hand, to avoid two-step limitations C-Motif [18] conducts these two tasks in a single step to directly generates the significant motifs. These approaches use local quality measure functions to rank the statistical significance of discovered phosphorylation motifs. Particularly, OR and GR (i.e risk ratio) are used in Motif-All and C-Motif respectively. Experimental results show that these approaches outperform other alternative methods such as MoDL[28], MMFPh [30], Motif-X [112] and F-Motif [113].

Although there exist some successful methods to discover phosphorylation motifs, the number of generated patterns is still high. It is difficult for biologist to interpret the results. Further research with computation and statistics perspective will be needed to reduce the execution time and amount of reported motif combinations.

1.5.4 Regulatory motif combinations mining

Transcription factors (TFs) is a critical component of the cellular machinery [32]. Usually some TFs work together to enable cells to respond to various signals. Similar to other biological pattern discoveries, the detection of multiple TFs combinations is not only computationally challenging but also extremely unlikely because of multiple testing correction. With k motifs taken into consideration, the number of tests for all combinations increases exponentially to k. It is well known that false positives may occur due to the multiple hypothesis tests.

Fig 1.3 illustrates an example of regulation motif combination in N genes. In this example, each gene has one expression level. For a given motif, N genes are partitioned into two groups, regulatory or unregulatory, depending on the p_value of motifs. A combination of motifs is compounded by all its motif members. For example, Motif 1,2,3 is the combination of three motifs: Motif 1, Motif 2 and Motif 3. The statistical significance of a motif combination is evaluated by p_value which is computed from Fisher's exact test. If its p_value is below a given threshold, it is considered as a regulatory motif.

Recently, to discover regulatory motif combinations, various statistically significant discriminative pattern mining algorithms have been proposed [31, 32, 74, 33, 72]. These approaches effectively discover many significant regulatory motif combinations. They not only generate limited number of patterns but also retrieve a set of motif combinations which satisfy the corrected statistical significance level.

For example, FastWY [31] performs experiments on two datasets: yeast dataset consists of 102 motifs in 5,988 genes and human dataset consists of 397 motifs in 11,610 genes. Experimental results show that FastWY efficiently finds statistically significant motif combinations. In particular, in the yeast dataset, it discovers 12

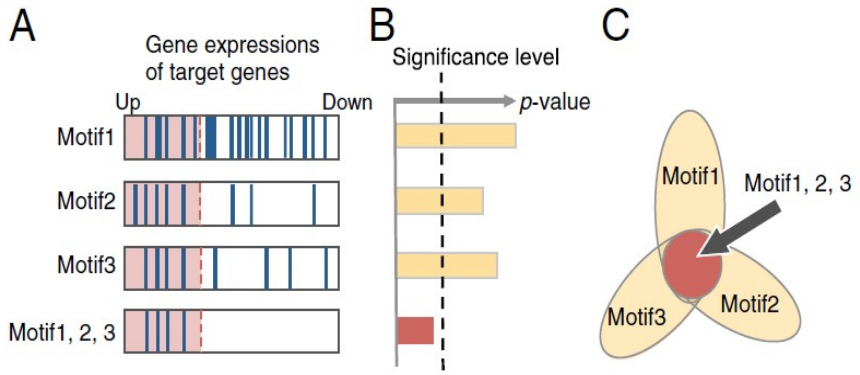


Figure 1.3: An example of regulation motif combination [32]. (A) Three individual motifs and a combination of three motifs with gene expression levels. (B) p_value of the motifs. (C) Combination of three motifs.

patterns which satisfy adjusted significance level $\delta = 0.000580$. The largest pattern contains 4 motifs. In the human dataset, it discovers 7 patterns which satisfy adjusted significance level $\delta = 1.781 \cdot 10^7$. The largest pattern contains 8 motifs. In general, FastWY discovers high statistically significant motif combinations. However, the running time is still high. For example, with human dataset, FastWY spends approximate 100,000 seconds to discover 7 statistically significant motif combinations.

On the other hand, FACS [72] applies Tarone's testability criterion to the CMH test to discover statistically significant motif combinations. Experiment on breast cancer dataset which includes 12,773 genes and 397 motifs shows that FACS is more efficient than other approaches. For example, in comparison with LAMP, the performance of FACS is better. In addition, FACS generates only 26 statistically significant motif combinations. This number of patterns is approximate 3% proportion of motif patterns which are discovered by LAMP.

Briefly, statistically significant discriminative pattern discovery algorithms are efficient methods for dealing with searching patterns in biological datasets. They generate limited number and high statistically significant patterns. However, the computation cost is still high.

1.6 Conclusion

In this chapter, a comprehensive study of discriminative pattern mining techniques and its applications in bioinformatics has been presented. We introduce a uniform definition of discriminative pattern mining with various search objectives such as local, global and statistically significant discriminative patterns. In addition, some popular statistical measures for discriminative power and p_value correction are also discussed.

Discriminative pattern mining techniques have been applied to tackle GWAS which is the most important task of bioinformatics. However, there are some remaining challenges that prevent them to directly handle large SNP datasets. These challenges include association strength measure, high-order SNP combinations searching, statistical significance testing and interesting SNP combinations visualization. To address these challenges, this thesis advances state-of-the-art of discriminative pattern mining techniques to discover genetic variant combinations associated with interesting phenotype.

Chapter 2

Identifying Genetic Variant Combinations Using Skypatterns

This chapter presents the method to identify genetic variant combinations associated with diseases by using the skypattern technique. This technique allows combinations of measures to be used to evaluate the importance of genetic variant combinations without having to select a given measure and a fixed threshold.

2.1 Introduction

Local discriminative pattern mining algorithms have been applied to discover genetic variant combinations associated with diseases [16, 103]. These algorithms directly use local quality measures to evaluate the association strength between SNP combinations and diseases. However, a wide range of statistical discriminative power measures are available. Selecting the most appropriate measures in biological situations remains a major challenge. In addition, for each measure users have to indicate an appropriate threshold to evaluate the importance of patterns, which is an extremely difficult task, specific to each particular biological datasets. The reason is that when the thresholds are not strict, the pattern mining algorithms generate many patterns of limited interest. On the other hand, some interesting patterns may be lost if the constraints are too restrictive.

To address these challenges, we propose to use the skypattern technique, which is based on a Pareto-dominance relation between set of measures, to evaluate the association strength of variant combinations and diseases. Skypattern technique has been introduced by [114]. This technique allows multi-criteria decision to be taken

in a threshold free manner to evaluate the importance of patterns. Given a set of patterns, each pattern is evaluated by a set of measures. Skypatterns are patterns which have dominance over the other patterns. Skypatterns are highly interesting since they not only receive a global evaluation from the set of measures, but also do not require any thresholds on the measures.

This chapter is organized as follows. Next section focuses on skypattern techniques which are used to evaluate the interestingness of a pattern. Then various experiments are conducted to illustrate the efficiency of the skypattern technique in identifying genetic variant combinations associated with diseases. In the last section, a summary of the results and future research directions are given.

2.2 Skypatterns

Pattern mining techniques use threshold-based or top-k-ranking strategy to select the interesting patterns. However, it is difficult to choose an appropriate threshold or a k value in most practical situations. To solve this problem, [114] proposed to use skyline queries to mine skyline patterns (or skypatterns) in a threshold-free manner. The idea is that each pattern is evaluated by a set of measures. Pattern x is evaluated better than pattern y if x dominates y. It means that x has at least one measure better than y, and the other measures of x must be not worse than the measures of y. A traditional example for this problem is retail transaction data in which each transaction corresponds to a client invoice; and every item in the transaction is a product bought by the client. Individual patterns are evaluated by some criteria such as frequency, size and price respectively. A user selecting a set of patterns may consider a pattern with high frequency, large size and low price. In this case, we say that pattern x dominates another pattern y if $x.frequency \geq y.frequency$, $x.size \geq y.size$, $x.price \geq y.price$, where at least one inequality is strict. The general definitions of skypatterns are stated as follows:

We consider D and I as defined in Chapter 1. An individual pattern is evaluated by a set of k measures $M = \{m_1, m_2, ..., m_k\}$.

Definition 2.1 (Dominance) Given a set of measures M, a pattern p dominates another pattern q with respect to M, denoted by $p \succ_M q$, iff $\forall m \in M$, $m(p) \geq m(q)$ and $\exists m \in M$ such that $m(p) > m(q)$.

Definition 2.2 (Skypattern and skypattern operator) Given a set of patterns P, each pattern is evaluated by a set of measures M. A skypattern with respect to M is a pattern not dominated in M. The skypattern operator, which is denoted by $Sky(M)$, returns all the skypatterns with respect to M.

Table 2.1: Example of transaction dataset

Transactions	i_1	i_2	i_3	i_4	i_5	i_6
t_1	1	1	1	1	0	1
t_2	1	1	1	1	1	0
t_3	1	1	0	0	0	0
t_4	0	0	0	1	0	0
t_5	1	0	1	0	0	0
t_6	0	0	0	0	1	0

Table 2.2: Skypatterns with respect to the set of measures $M = \{freq, size\}$

Patterns	$freq$	$size$
i_1	4	1
i_1i_2	3	2
i_1i_3	3	2
$i_1i_2i_3i_4$	2	4

$$Sky(M) = \{p \in P| \nexists q \in P : q \succ_M p\}$$

Given a set of measures M, the skypattern mining problem is thus to evaluate the query $Sky(M)$ over 2^I patterns.

For example, Table 2.1 presents a transaction dataset including 6 transactions denoted by $t_1, ..., t_6$ which are described by 6 items $i_1, ..., i_6$. Each individual pattern is evaluated by a set of measures M including:

- m_1: $freq(p)$ is the frequency of pattern p.
- m_2: $size(p)$ is cardinality of pattern p.
- m_3: $area(p) = freq(p) * size(p)$.

Considering pattern $i_1i_2i_3i_4$ for example, we have $freq(i_1i_2i_3i_4) = 2$, $size(i_1i_2i_3i_4) = 4$ and $area(i_1i_2i_3i_4) = 8$.

Suppose using $M = \{freq, size\}$ as a set of measures, pattern $i_1i_2i_3i_4$ dominates pattern $i_1i_2i_3$ since $freq(i_1i_2i_3i_4) = freq(i_1i_2i_3)$ and $size(i_1i_2i_3i_4) > size(i_1i_2i_3)$.

Skypattern operator with respect to M generates a set of skypatterns which is shown in Table 2.2. Graphical presentation of $Sky(M)$ is illustrated in Fig 2.1. The shaded area is called the dominated area since it cannot contain any skypatterns.

Mining skypatterns with respect to the set of measures is a computational challenge. This process can be done with an exhaustive search strategy: i.e., first discover all patterns, then run domination tests with respect to the set of measures to find skypatterns. In practice, this approach is not feasible since the collection of patterns

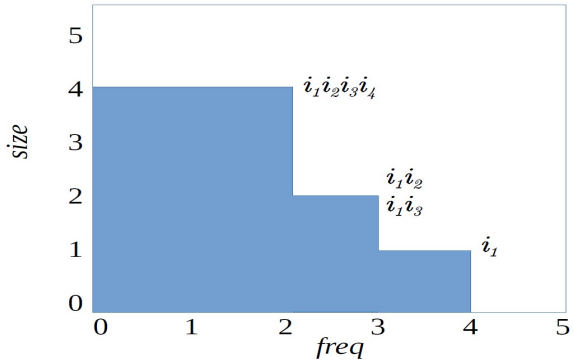


Figure 2.1: Graphical presentation of $Sky(M)$

is often very large to be manageable. Obviously, to limit the size of the collection, constraints might be introduced. However, the consistency of the result may be lost (i.e., some skypatterns may not be produced) and the thresholding problem would remain.

2.3 Skypatterns cube

In practice, selecting the most appropriate set of measures to evaluate the importance of patterns is a difficult task since users may not know exactly the role of each measure. Nevertheless, users can keep all the potential measures; then add or remove a measure to look how the skypattern set changes. To explore the different sets of measures, [115] proposes the notion of *skypattern cube*. The skypattern cube is a lattice over all subsets of measures where each node of the lattice corresponds to a subset of measures and its skypattern set. Based on this structure, users can have a better understanding about the role of measures by observing the new skypatterns or the ones which disappear when adding or removing a measure in two neighboring nodes. Additionally, different subsets of measures may lead to the same set of skypatterns and thus be shown as equivalent. This helps users to classify the measure subsets effectively. The definition of the skypattern cube is given as follows:

Definition 2.3 (Skypattern cube) Given a set of measures M, the skypattern cube with respect to M, denoted by $SkyCube(M)$, consists of $2^{|M|} - 1$ skypattern sets which are generated by $Sky(M_u)$, for all $M_u \subseteq M$.

$$SkyCube(M) = \{(M_u, Sky(M_u)) | M_u \subseteq M, M_u \neq \emptyset\}$$

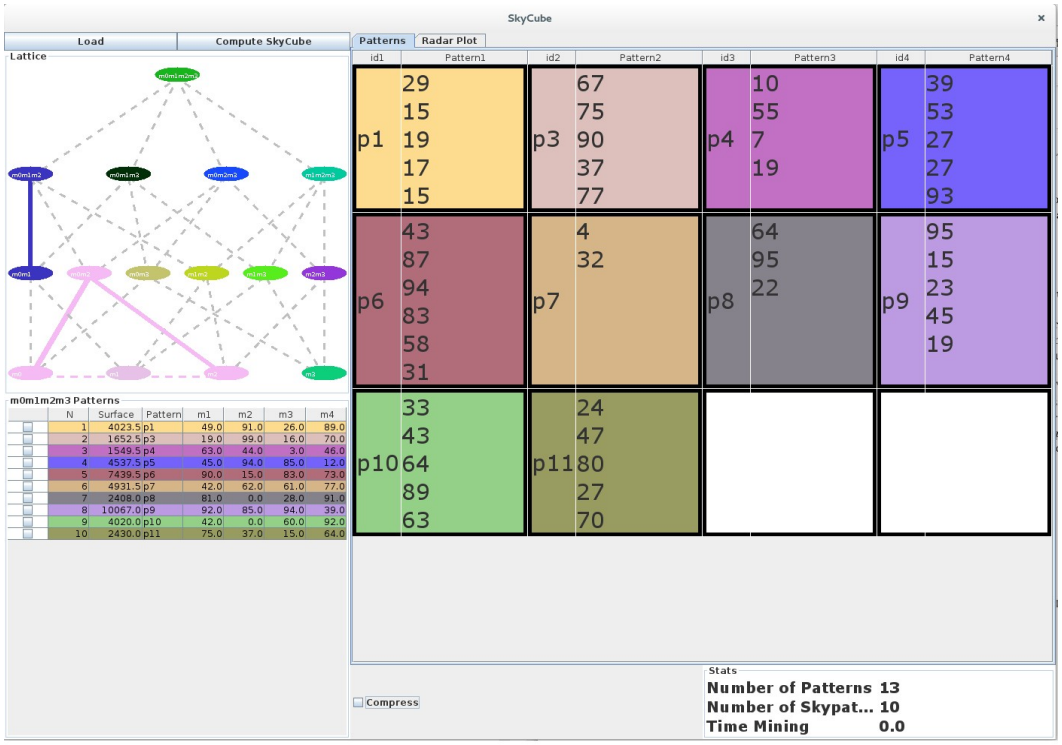


Figure 2.2: Full lattice association to 4 measures

To compute skypattern cube, the *SkyCube* software can be used [115]. This software discovers and presents skypatterns in a lattice structure which enable users to perform various queries effectively and to discover the most interesting skypattern sets. For example, Fig 2.2 illustrates the relative lattice which is generated by SkyCube. This lattice presents all subsets of 4 measures and their equivalent skypatterns. Users can choose a specific subset of measures to view the related skypatterns.

Whole skypattern cube may generate skypatterns that are redundant. For example, a skypattern p can be present in many different nodes. Thus, we use the compression function of the SkyCube to keep only the *proper skypatterns* of each node. A proper skypattern is a skypattern that is not derived from its child nodes. For example p is a proper skypattern for $\{m_1, m_2\}$ if p is not a skypattern for $\{m_1\}$ nor $\{m_2\}$. In some cases a node may not have proper skypatterns, so it disappears from the compressed SkyCube. For example, Fig 2.3 shows the relative compressed lattice of a set of 4 measures. The lattice shows only the nodes which generate

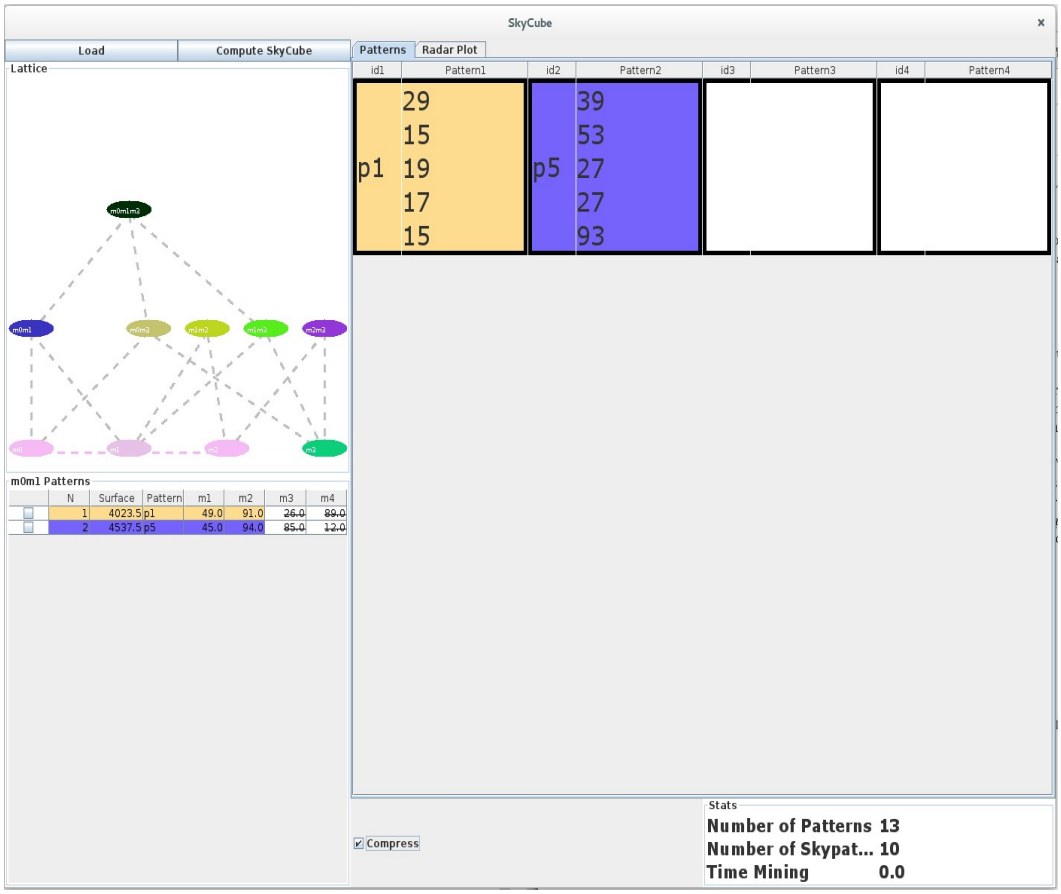


Figure 2.3: Compressed lattice association to 4 measures

proper skypatterns.

2.4 Experiments

In this section, we use skypattern technique to identify SNP combinations associated with diseases. To evaluate the efficiency of skypattern technique, various genetic variant datasets and association strength measures are used.

Table 2.3: Seven common diseases datasets

No	**Diseases**	**Genes**	**Chromosome**	**SNPs**
1	Bipolar disorder (BD)	PALB2	16	rs420259
2	Coronary artery disease (CAD)	CDKN2A	9	rs1333049
3	Crohn's disease (CD)	BSN	3	rs9858542
4	Hypertension (HT)	RYR2	1	rs2820037
5	Rheumatoid arthritis (RA)	PTPN22	1	rs6679677
6	Type 1 diabetes (T1D)	KIAA0350	16	rs12708716
7	Type 2 diabetes (T2D)	TCF7L2	10	rs4506565

2.4.1 Datasets

In this study, we use 7 real case-control genetic variant datasets which are provided by Wellcome Trust Sanger Institute (http://www.sanger.ac.uk/). Based on the results of [116], the strong SNP signals associated with diseases are showed in Table 2.3. Since discovering all SNP combinations in large case-control dataset is time-consuming, for each dataset we select 100 SNPs including SNP related to disease on a particular chromosome.

The purpose of the experiments is to evaluate the effectiveness of skypatterns with respect to the set of measures. The effectiveness of a measure (or set of measures) is assessed based on the number of interesting SNP genotype combinations that it found. According to the literature there is no report related to SNP combinations association with these diseases. In this study we suppose that the interesting SNP genotype combinations are patterns containing at least one of the SNPs related to diseases reported by the literature.

2.4.2 Mining skypatterns strategy

Exhaustively mining SNP genotype combinations and calculating skypattern cube are computationally challenging. Thus, in these experiments, the size of combinations are limited to three SNP genotypes. To discover SNP combinations and compute skypatterns the following steps are conducted.

First, we use a brute-force strategy to mine all SNP genotype combinations of size 3. This process guarantees that all 3-SNP genotype combinations are taken into consideration. The exhaustive search strategy generates a very large number of patterns. Thus, to reduce the size of pattern sets, we filter the less interesting ones. In particular, the patterns having support in case group $< 10\%$ and support

Table 2.4: Discriminative power measures

No	Measures	Denoted
1	Difference support	DS
2	Growth rate	GR
3	Odds ratio	OR
4	Chi square	X^2
5	Weighted Relative Accuracy	$WRAcc$
6	Mutual information	MI
7	Information gain	IG
8	SupMaxPair	$SupMaxPair$

in control group $> 50\%$ are removed. After reducing, each set of patterns consists of approximately 500,000 patterns.

Then we use 8 measures which are shown in Table 2.4 to evaluate the importance of SNP genotype combinations. These are popular measures for evaluating the discriminative power of patterns in two-class datasets. They are often adopted to evaluate the association strength between biological patterns and interesting phenotype.

Finally, SkyCube software is used to find skypatterns over these SNP genotype combinations.

2.4.3 Results

2.4.3.1 Individual measures results

Firstly, we evaluate the effectiveness of 8 individual measures for identifying the SNP genotype combinations related to diseases. This result is used as a baseline to compare and evaluate the effectiveness of skypatterns over patterns evaluated by individual measures. For each measure, we select the top 100 patterns which have the highest discriminative power for analysis. The effectiveness of individual measures is assessed based on the number of patterns containing the SNP genotype associated to disease in this set. Table 2.5 shows the number of interesting SNP genotype combinations which are identified by individual measures in 7 datasets.

The most effective measure is X^2 which can discover interesting SNP genotype combinations in all datasets. The highest effectiveness of X^2 is for RA disease. However, in the other datasets, the efficiency of X^2 decreases. In contrast, the group of measures including DS, $WRAcc$, $SupMaxPair$ is the least effective. These measures can only detect variants related to disease in some datasets. The other measures such as GR, OR, MI and IG give a higher effectiveness. The notable methods

Table 2.5: Number of risk patterns identified by individual metrics

No	Measures	BD	CAD	CD	HT	RA	T1D	T2D
1	DS	0	10	0	20	97	0	0
2	GR	4	33	0	21	91	10	9
3	OR	5	53	0	19	91	10	9
4	X^2	2	27	16	21	100	4	47
5	$WRAcc$	0	10	0	22	98	0	0
6	MI	3	9	0	18	90	16	11
7	IG	4	53	0	22	98	8	0
8	$SupMaxPair$	0	0	0	53	0	10	0

Table 2.6: The highest effectiveness of two-measure sets

Measures	BD	CAD	CD	HT	RA	T1D	T2D
$\{GR, SupMaxPair\}$	0/12	10/15	13/21	12/15	10/14	6/14	7/24
$\{OR, SupMaxPair\}$	0/14	8/13	13/21	8/11	10/15	4/9	2/20
$\{MI, SupMaxPair\}$	0/35	25/56	16/36	17/51	10/43	6/24	18/51

in this group are OR and GR. Both of them discover risk variant combinations in 6/7 datasets. In short, there is no best measure for all datasets. However, each measure effectively identifies risk variant combinations in a particular dataset.

2.4.3.2 Skypattern results

We then analyze the skypatterns generated from SkyCube. According to the subsets of measures which generate proper skypatterns, we analyze the skypattern sets with respect to the combinations of 2 to 4 measures.

Firstly, we consider the skypattern sets with respect to 2 measures. Based on the number of interesting SNP genotype combinations found in each skypattern set, the most effective 2-measure combination is $\{GR, SupMaxPair\}$. The lowest effective methods is $\{DS, WRAcc\}$. Table 2.6 presents the most effective 2-measure combinations. Note that, in this table, the effectiveness of the measure combinations are presented by the number of risk patterns per total skypatterns.

These 2-measure combinations can identify many risk variant combinations in their equivalent skypattern sets. Considering CD dataset for example, these measure compounds can detect interesting SNP genotype combinations effectively. Particularly, the ratio of skypatterns containing risk variant over the total of skypatterns of $\{GR, SupMaxPair\}$, $\{OR, SupMaxPair\}$, and $\{MI, SupMaxPair\}$ are 13/21, 13/21, and 16/36 respectively. Notably, for this dataset most individual measures

Table 2.7: The highest effectiveness of three-measure sets

Measures	**BD**	**CAD**	**CD**	**HT**	**RA**	**T1D**	**T2D**
$\{OR, MI, SupMaxPair\}$	1/54	29/40	3/20	45/123	14/39	2/6	9/28
$\{X^2, MI, SupMaxPair\}$	1/218	126/198	3/36	60/230	46/71	13/29	37/136
$\{WRAcc, MI, SupMaxPair\}$	1/130	61/102	10/135	88/266	69/100	31/111	13/103

Table 2.8: The effectiveness comparison of $\{GR, SupMaxPair\}$ and $\{OR, MI, SupMaxPair\}$

Measures	**BD**	**CAD**	**CD**	**HT**	**RA**	**T1D**	**T2D**
$\{GR, SupMaxPair\}$	0	0.67	**0.62**	**0.8**	**0.71**	**0.43**	0.29
$\{OR, MI, SupMaxPair\}$	**0.02**	**0.73**	0.15	0.37	0.36	0.33	**0.32**

cannot detect risk variant combinations in the top of 100 patterns, except X^2.

Similarly, the result of the most effective 3-measure combinations is presented in Table 2.7. According to this result, $\{OR, MI, SupMaxPair\}$ is the most effective 3-measure combination. It identifies risk variant groups in all datasets. The highest effectiveness is for CAD with 29 out of 40 skypatterns containing risk SNP genotype. However, this combination is less efficient in BD where there is only 1 skypattern including risk variant over 54 skypatterns.

In comparison with 2-measure combinations, the set of measures $\{OR, MI, SupMaxPair\}$ is less effective. For example, with 7 datasets, there are 4 out of 7 datasets in which the combination of $\{GR, SupMaxPair\}$ is better than $\{OR, MI, SupMaxPair\}$. Table 2.8 presents the comparison of $\{GR, SupMaxPair\}$ and $\{OR, MI, SupMaxPair\}$. Note that, to compare easily we used the ratio (the number of interesting patterns per total number of skypatterns) to present the effectiveness of measure combinations.

The combination of two or three measures can effectively discover the groups of variants associated to diseases. However, it is less effective when we use 4-measure combinations. Particularly, these 4-measure combinations can only identify risk SNP genotype combinations in 2 out of 7 datasets including CAD and T2D. In the other remaining datasets, there is no risk variant combinations detected although the number of generated skypatterns are high. Especially, the SkyCube doesn't generate any proper skypattern sets which corresponds to the combination of more than 4 measures.

Table 2.9: The comparison between 2-measure combinations and X^2

Measures	**BD**	**CAD**	**CD**	**HT**	**RA**	**T1D**	**T2D**
X^2	0/12	1/15	1/21	2/15	**14/14**	0/14	6/24
$\{GR$, $SupMaxPair\}$	0/12	**10/15**	**13/21**	**12/15**	10/14	**6/14**	**7/24**
X^2	**1/14**	1/13	1/21	2/11	**15/15**	0/9	**6/20**
$\{OR$, $SupMaxPair\}$	0/14	**8/13**	**13/21**	**8/11**	10/15	**4/9**	2/20
X^2	**1/35**	10/56	5/36	13/51	**43/43**	0/24	**24/51**
$\{MI$, $SupMaxPair\}$	0/35	**25/56**	**16/36**	**17/51**	10/43	**6/24**	18/51

2.4.3.3 Individual measures and skypatterns comparison

In order to confirm the effectiveness of measure combinations over individual measures, we compare them with X^2 which is the most efficient individual metrics. For fair comparison, the number of highest X^2 patterns is reselected. For each dataset, we select the top-k patterns in descending order of X^2 where k is the number of skypatterns which are generated from the combination of measures in that dataset. This comparison is fair as it considers in both cases the k first patterns that an analyst will examine. The efficiency of one method is evaluated better than the other if its pattern set contains a higher number of risk SNP genotype combinations. The comparison between 2-measure combinations and X^2 is showed in Table 2.9.

According to this result, the skypatterns with respect to $\{GR, SupMaxPair\}$ contain more interesting SNP genotype combinations than X^2 does. Specifically, there are 5 out of 7 datasets in which $\{GR, SupMaxPair\}$ is better than X^2. They are equally efficient in BD; and less effective than X^2 in RA. Similarly, the effectiveness of $\{OR, SupMaxPair\}$ and $\{MI, SupMaxPair\}$ are also better in average than X^2. To be more specific, in 4 out of 7 datasets these methods are better than X^2, but they are worse than X^2 in the 3 remaining datasets (BD, HT, T2D).

In addition, the set of measures $\{OR, MI, SupMaxPair\}$ is more effective than X^2. Specifically, there are 5 out of 7 datasets in which $\{OR, MI, SupMaxPair\}$ is better than X^2; one is equal; and another one is less efficient than X^2. Table 2.10 illustrates the comparison of $\{OR, MI, SupMaxPair\}$ and X^2.

To sum up, according to the results, using combination of measures is more effective than using individual measures. Particularly, X^2 is the most effective individual measure, whereas, $\{GR, SupMaxPair\}$ and $\{OR, MI, SupMaxPair\}$ are the most effective for two and three measure combinations. In comparison with X^2, both of

Table 2.10: The comparison between {OR, MI, SupMaxPair} and X^2

Measures	**BD**	**CAD**	**CD**	**HT**	**RA**	**T1D**	**T2D**
X^2	1/54	6/40	1/20	25/123	**39/39**	0/6	8/28
$\{OR, MI,$ $SupMaxPair\}$	1/54	**29/40**	**3/20**	**45/123**	14/39	**2/6**	**9/28**

$\{GR, SupMaxPair\}$ and $\{OR, MI, SupMaxPair\}$ are more efficient than X^2. The set of measures $\{OR, MI, SupMaxPair\}$ is less effective than $\{GR, SupMaxPair\}$ slightly. The compound of 2 or 3 measures are effective but the combination of 4 measures or higher are not useful in our setting.

2.5 Conclusion

In this chapter we proposed to use the skypattern technique to identify the groups of genetic variants associated with diseases. The experiments on various SNP datasets demonstrate that the proposed method is promising. The skypatterns with respect to the set of two or three statistical measures can effectively detect SNP genotype combinations related to diseases. In comparison with X^2, the most effective individual method, the set of two or three measures give a higher efficiency. However, it is not necessary to use more than 3-measures combinations since they do not generate proper skypatterns effectively.

The skypattern technique has a good potential to evaluate the association strength between SNP combinations and diseases. However, mining skypatterns with regard to multiple measures is a time-consuming task. Thus, to use this technique for larger genetic variant datasets further research is required.

Chapter 3

Searching for Statistically Significant Discriminative Patterns in Genomic Data

This chapter presents an efficient algorithm to search statistically significant discriminative patterns in two-class datasets. It has been efficiently applied in a two-step framework to discover high-order SNP combinations associated with diseases.

3.1 Introduction

Using the skypattern technique to evaluate the association strength of SNP combinations and diseases is an interesting approach. It has been demonstrated that relevant SNP combinations associated with diseases can be identified by using groups of risk measures. The proposed approach in the previous chapter can only tackle small genetic variant datasets since discovering SNP combinations is a computational challenge. The available local discriminative pattern mining algorithms can be applied to handle this problem. However, some major problems remain.

First, they are exclusively based on enumeration strategies. This is a very time-consuming approach for datasets with a large number of items (where the "items" are SNPs in biological datasets). Many discriminative patterns cannot be discovered due to the exponential number of combinations among individual items. In addition, patterns of little biological interest may occur. A post-processing step (or domain knowledge step) is often required to select patterns with potential biological interest [16, 103, 22].

Second, most of discriminative measures are not anti-monotonic [44, 45, 47, 13]. It means that there exists no correlation between a pattern and its subsets with

regard to discriminative scores. Thus, discriminative measures cannot be used to prune the search space like in classical frequent itemset mining [47, 61].

Third, mining low frequency patterns is algorithmically challenging. The approaches based on the frequency threshold usually ignore these patterns. However, in practice, there exists many patterns with a low frequency but high discriminative scores. Discovering these patterns is necessary since they give valuable information [24, 23].

Fourth, beside the computational problems, multiple hypothesis testing is an even more serious challenge. Existing algorithms often generate a large number of combinations. Many of them could be discovered even due to random chance. Thus, a huge number of hypothesis tests is needed to test the statistical significance of results [71, 33, 72].

In this chapter, we propose an algorithm, named "Statistically Significant Discriminative Pattern Search" (SSDPS), that discovers discriminative patterns in two-class datasets. More precisely, the SSDPS algorithm aims at searching patterns satisfying both discriminative scores (equivalent to risk scores) and confidence intervals thresholds. These patterns are defined as **statistically significant discriminative patterns**. The SSDPS algorithm is based on a strategy in which risk measures and confidence intervals can be used as anti-monotonic properties. These properties allow the search space to be efficiently pruned. All patterns are directly tested for risk scores and confidence intervals in the mining process. Only patterns satisfying discriminative and statistical significance thresholds are considered as the target output. The algorithm can discover complete set of discriminative patterns with a very low frequency threshold. It can also use heuristic strategies to mine only the largest statistically significant discriminative patterns with regard to a set of risk measures and confidence intervals. The heuristic strategies allow users to choose a trade-off between execution time and result quality.

The SSDPS algorithm has been used to conduct various experiments on both synthetic datasets and real SNP datasets: Age-Related Macular Degeneration, Breast Cancer and Type 2 Diabetes. The experiments show that SSDPS algorithm can effectively discover interesting patterns with a short execution time. Many of them contain SNPs which are already known as associated with diseases. In addition, the SSDPS algorithm detects patterns which include very low frequency SNPs, and which can open new investigations. We also evaluate the performances of SSDPS algorithm. They are comparable with other existing methods such as SFP-Growth [103] or CIMCP [60], while the proportion of generated patterns is less than the amount of patterns output by these methods.

The rest of this chapter is organized as follows: Section 3.2 presents the background of risk measures and statistical significance tests which are used to evaluate

Table 3.1: A 2x2 contingency table of a pattern in case-control data

	Presence	Absence	Total
Case	a	b	$a+b$
Control	c	d	$c+d$

the discriminative patterns and prune the search space. Section 3.3 precisely defines the concept of statistically significant discriminative pattern, and Section 3.4 presents the enumeration strategy used by the SSDPS algorithm. In Section 3.5, the design and implementation of the SSDPS algorithm are described. Section 3.6 is dedicated to experiments and results. Section 3.7 concludes the chapter.

3.2 Risk measures and statistical significance tests

In this section, we present the background of risk measures and statistical significance tests which are used as constraints in the SSDPS algorithm to efficiently discover patterns with a high statistical significance.

3.2.1 Risk measures

Odds ratio (OR), risk ratio (RR) and absolute risk reduction (ARR) are bio-statistics measurements that are widely used for testing association in case-control studies [117] [59] [118]. They are used to quantify how strongly the presence or absence of property A is associated with the presence or absence of property B in a given population. Suppose that cases and controls are conducted to evaluate exposure to a suspected causal factor. The observation data can be tabulated by a 2x2 contingency table as shown in Table 3.1.

Where:

a is the number of presence in case group.

b is the number of absence in case group.

c is the number of presence in control group.

d is number of absence in control group.

The OR, RR, ARR are estimated based on the relation of odds between the two groups of subjects. They are computed by the following equations:

$$OR = \frac{a/b}{c/d} = \frac{a.d}{b.c} \tag{3.1}$$

$$RR = \frac{a/(a+b)}{c/(c+d)} \tag{3.2}$$

$$ARR = \frac{a}{a+b} - \frac{c}{c+d} \tag{3.3}$$

The estimation of OR, RR, or ARR indicates the association between variants and disease. In particular, there is no association if $OR = 1$, $RR = 1$, or $ARR = 0$; risk increases if $OR > 1$, $RR > 1$, or $ARR > 0$; risk decreases if $OR < 1$, $RR < 1$, or $ARR < 0$.

Finding variant combinations with high risk scores is the objective of GWAS. It shows that variant combinations may be associated with a specific disease.

For example, observing variant combination p in 100 individuals effected by Type 2 Diabetes (case group), and 100 healthy individuals (control group), we have the following results:

Situation 1: p occurs in 50 case individuals and 40 control individuals. OR, RR and ARR are equal to: $OR = \frac{50/50}{40/60} = 1.5$, $RR = \frac{50/100}{40/100} = 1.25$, $ARR = \frac{50}{100} - \frac{40}{100} = 0.1$.

$OR > 1$, $RR > 1$ and $ARR > 0$ indicate that p is associated with disease.

Situation 2: p occurs in 60 cases and in 10 controls. OR, RR and ARR are equal to: $OR = \frac{60/40}{10/60} = 9$, $RR = \frac{60/100}{10/100} = 6$, $ARR = \frac{60}{100} - \frac{10}{100} = 0.5$.

In this situation the association between p and disease is strongly recognized.

3.2.2 Statistical significance tests

p_value and confidence intervals are statistical measures. Both of them are often used to assess the statistical significance of results since they provide complementary information [119, 118].

3.2.2.1 *p_value*

The *p_value* is a probability, which is the result of a statistical test. It is used to determine if a null hypothesis of a study is to be accepted or rejected, or used to determine the statistical significance of results. A small *p_value* corresponds to a strong evidence. The results are indicated as "statistically significan" if the *p_value* is below a given threshold. A *p_value* threshold of 0.05 (or 5%) is often chosen to indicate the level of significance [2]. The *p_value* can be estimated by different mathematical methods such as Fisher Exact Probability Test or Pearson's chi-square test.

3.2.2.2 Confidence intervals

Confidence intervals (CI) are the result of a statistical measure. They provide information about a range of values (lower confidence interval (LCI) to upper confidence

interval (UCI)) in which the true value lies with a certain degree of probability. CI is able to assess the statistical significance of a result [118]. A confidence level of 95% is usually selected. It means that the CI covers the true value in 95 out of 100 studies.

A 95% CI for the population value of OR is estimated by the two quantities: lower CI (denoted LCI_{OR}) and upper CI (denoted UCI_{OR}).

$$LCI_{OR} = e^{\left(ln(OR) - 1.96\sqrt{\frac{1}{a}+\frac{1}{b}+\frac{1}{c}+\frac{1}{d}}\right)} \tag{3.4}$$

$$UCI_{OR} = e^{\left(ln(OR) + 1.96\sqrt{\frac{1}{a}+\frac{1}{b}+\frac{1}{c}+\frac{1}{d}}\right)} \tag{3.5}$$

Similarly, a 95% CI for the population value of RR is estimated by the two quantities: lower CI (denoted LCI_{RR}) and upper CI (denoted UCI_{RR}).

$$LCI_{RR} = e^{\left(ln(RR) - 1.96\sqrt{\frac{1}{a}-\frac{1}{a+b}+\frac{1}{c}-\frac{1}{c+d}}\right)} \tag{3.6}$$

$$UCI_{RR} = e^{\left(ln(RR) + 1.96\sqrt{\frac{1}{a}-\frac{1}{a+b}+\frac{1}{c}-\frac{1}{c+d}}\right)} \tag{3.7}$$

In case-control studies, $OR = 1$ or $RR = 1$ indicates "no association" between the exposure and the disease. Thus, if the 95% CI does not contain the value 1.0, the association is statistically significant at 0.05. In contrast, if the 95% CI of OR or RR contains 1.0, the association is not significant at the 0.05 level.

Consider the previous example. Suppose $p\_value = 0.05$ is the significant level (obtained by Fisher Exact Probability Test) and we expect to find variants associated with disease (OR and RR larger than 1). We have the following results:

In situation 1, we have: $p\_value = 0.1$; 95% CI of OR is (0.856 - 2.626); 95% CI of RR is (0.9169 - 1.7041). Hence, p is not statistically significant at the 0.05 level, since both 95% CI of OR and RR contain 1, and p_value does not satisfy the significance threshold (although the OR and RR are larger than 1).

In situation 2, we have: $p\_value = 2.5e^{-14}$; 95% CI of OR is (6.2751 - 29.0434); 95% CI of RR is (3.2621 - 11.0358). Hence, p is statistically significant since all 95% CI do not contain 1 and p_value satisfies the significance threshold.

In short, OR, RR or ARR of a result larger than a predefined limit does not necessarily indicate that this association is statistically significant. Users must consider the CI or p_value to determine significance.

Table 3.2: Transaction table of two-class data

Tids	Items										Class
1	a	b	c			f			i	j	1
2	a	b	c		e		g		i		1
3	a	b	c			f		h		j	1
4		b		d	e		g		i	j	1
5				d		f	g	h	i	j	1
6		b	c		e		g	h		j	0
7	a	b	c			f	g	h			0
8		b	c	d	e			h	i		0
9	a			d	e		g	h		j	0

3.3 Statistically significant discriminative patterns

The goal of the study is to find patterns in GWAS that are at the same time discriminative and statistically significant, as defined in Section 3.2. In this section, we present our definition of such patterns.

The input of discriminative pattern mining algorithms or GWASs is presented as a matrix including n rows and m columns. Each row corresponds to a transaction (or an individual) which belongs to positive or negative class (case or control group), whereas columns are items (or SNPs).

For example, Table 3.2 presents a dataset including 9 transactions (identified by 1..9) which are described by 10 items (denoted by $a..j$). The dataset is partitioned into two classes. The positive class (class label = 1) includes 5 transaction ids from 1 to 5, and the negative class (class label = 0) consists of 4 transaction ids from 6 to 9.

The objective of GWASs or discriminative pattern mining algorithms is to find groups of items satisfying some constraint thresholds such as risk ratio, odds ratio or risk difference.

The formal presentation of this problem is given in the following:

Let I be a set of m items $I = \{i_1, ..., i_m\}$ and S_1, S_2 be two labels .

A *transaction* over I is a pair $t_i = \{(x_i, y_i)\}$, where $x_i \subseteq I$, $y_i \in \{S_1, S_2\}$. Each transaction is identified by an integer, denoted *tid*.

A set of *tids* $T = \{1..n\}$ over I can be termed as a *transaction dataset* D over I.

The two sets of *tids* that belong to S_1 and S_2 are denoted by D_1 and D_2, and we have $|D| = |D_1| + |D_2|$.

A set $p \subseteq I$ is called an *itemset* (or pattern) and a set $q \subseteq \{1..n\}$ is called a *tidset*. For convenience, we write a tidset $\{1, 2, 3\}$ as 123, and an itemset $\{a, b, c\}$

as *abc*. The number of transactions in D_i containing p is denoted by $|D_i(p)|$. The *relational support* of pattern p in class D_i, denoted $sup(p, D_i)$, is defined as:

$$sup(p, D_i) = \frac{|D_i(p)|}{|D_i|} \tag{3.8}$$

The negated support of p in D_i, denoted $\overline{\text{sup}}(p, D_i)$, is defined as:

$$\overline{\text{sup}}(p, D_i) = 1 - sup(p, D_i) \tag{3.9}$$

Pattern p is *frequent* in D_i if its support value in D_i is no less than a given threshold; p is *closed frequent* in D_i if there doesn't exist any frequent pattern which contains p and has the same support as p in D_i; p is *maximal frequent* in D_i if it is not a subset of any other frequent pattern in D_i.

Taking again Table 3.2 as example, let $min\_sup = 0.3$ be the support threshold. Then *abc* is frequent since $sup(abc, D_1) = 3/5 = 0.6 \geq min\_sup$. In addition, *abc* is closed frequent in D_1 since there exist no frequent pattern containing *abc* and having the same support as *abc* in D_1. In contrast, $abcf$ is frequent but not closed frequent in D_1. Because $abcf$ is a subset of $abcfj$ and $sup(abcf, D_1) = sup(abcfj, D_1) = 2/5$. $abcfj$ is a maximal frequent pattern in D_1.

Discriminative score of a pattern p in dataset D, denoted $scr(p, D)$, is defined over the relational supports of p in the two classes such as support difference, growth rate or odds ratio support.

Support difference of pattern p in dataset D, denoted $SD(p, D)$, is defined as:

$$SD(p, D) = sup(p, D_1) - sup(p, D_2) \tag{3.10}$$

Growth rate support of pattern p in dataset D, denoted $GR(p, D)$, is defined as:

$$GR(p, D) = \frac{sup(p, D_1)}{sup(p, D_2)} \tag{3.11}$$

Odds ratio support of pattern p in dataset D, denoted $ORS(p, D)$, is defined as:

$$ORS(p, D) = \frac{sup(p, D_1)/\overline{\text{sup}}(p, D_1)}{sup(p, D_2)/\overline{\text{sup}}(p, D_2)} \tag{3.12}$$

For example, $sup(abc, D_1) = 0.6$, $sup(abc, D_2) = 0.25$, then we have $SD(abc, D) = 0.35$, $GR(abc, D) = 2.4$, $ORS(abc, D) = 4.5$.

Definition 3.1 (Discriminative pattern) Let α be a discriminative threshold, $scr(p, D)$ be the discriminative score of pattern p in D. The pattern p is discriminative if $scr(p, D) \geq \alpha$.

Table 3.3: The equivalence of terms between GWAS and discriminative pattern mining

GWAS	**Discriminative pattern mining**
Case group	Positive class
Control group	Negative class
Individual SNP	Item
SNP combination	Itemset (or pattern)
The presence of SNP combination in the case group	The number of transactions in the positive class containing pattern
The presence of SNP combination in the control group	The number of transactions in the negative class containing pattern

For example, let $\alpha = 0.2$ be the SD threshold. Then abc is a discriminative pattern since its score satisfies the threshold. In contrast, pattern $abcf$ is not discriminative since $SD(abcf, D) = 0.15$.

Searching for the combinations of SNPs associated with diseases is equal to finding discriminative patterns in two-class datasets. The equivalence of terms between GWAS and discriminative pattern mining is shown in Table 3.3. In this setting, the risk measures are discriminative measures [59]. In particular, the ARR of SNPs combination in the case and the control group can be exactly said to be the SD of a pattern in the positive and the negative class. Similarly, we can conclude the equivalence between RR and GR, between OR and ORS.

In addition, we can also demonstrate that 95% CI of GR and 95% CI of ORS are equivalent to 95% CI of RR and 95% CI of OR, respectively.

Let $a = |D_1(p)|$ (the number of transactions in D_1 that contains p), $b = |D_1| - |D_1(p)|$ (the number of transactions in D_1 that does not contain p), $c = |D_2(p)|$ (the number of transactions in D_2 that contains p), $d = |D_2| - |D_2(p)|$ (the number of transactions in D_2 that does not contain p).

A 95% CI of GR is estimated by lower CI (denoted LCI_{GR}) and upper CI (denoted LCI_{GR}) which are given by:

$$LCI_{GR} = e^{\left(ln(GR) - 1.96\sqrt{\frac{1}{a} - \frac{1}{a+b} + \frac{1}{c} - \frac{1}{c+d}}\right)} \tag{3.13}$$

$$UCI_{GR} = e^{\left(ln(GR) + 1.96\sqrt{\frac{1}{a} - \frac{1}{a+b} + \frac{1}{c} - \frac{1}{c+d}}\right)} \tag{3.14}$$

A 95% CI of ORS is estimated by lower CI (denoted LCI_{ORS}) and upper CI (denoted LCI_{ORS}) which are given by:

$$LCI_{ORS} = e^{\left(ln(ORS) - 1.96\sqrt{\frac{1}{a}+\frac{1}{b}+\frac{1}{c}+\frac{1}{d}}\right)} \tag{3.15}$$

$$UCI_{ORS} = e^{\left(ln(ORS) + 1.96\sqrt{\frac{1}{a}+\frac{1}{b}+\frac{1}{c}+\frac{1}{d}}\right)} \tag{3.16}$$

Definition 3.2 (Statistically significant pattern) Let β be a lower confidence interval threshold, $lci(p, D)$ be the lower confidence interval of pattern p in D. The pattern p is statistically significant if $lci(p, D) > \beta$.

Definition 3.3 (Statistically significant discriminative pattern) Given a discriminative threshold α and a lower confidence interval threshold β. Pattern p is statistically significant discriminative in D if $scr(p, D) \geq \alpha$ and $lci(p, D) > \beta$.

Problem statement: Given a two-class dataset D, the problem is to discover complete set of patterns P in D where all p in P satisfy $scr(p, D) \geq \alpha$ and $lci(p, D) > \beta$.

Note that this problem can be extended to discover all patterns which satisfy multiple discriminative score thresholds and confidence intervals. In particular, given a set of discriminative thresholds $\{SD = \alpha_1, GR = \alpha_2, ORS = \alpha_3\}$, and a set of lower confidence interval thresholds $\{LCI_{GR} = \beta_1, LCI_{ORS} = \beta_2\}$. We want to discover all patterns which satisfy $SD \geq \alpha_1$ and $GR \geq \alpha_2$ and $ORS \geq \alpha_3$ and $LCI_{GR} > \beta_1$ and $LCI_{ORS} > \beta_2$.

For example, let $\alpha_1 = 0.2, \alpha_2 = 2, \alpha_3 = 2$ be the thresholds of SD, GR, and ORS, respectively. *abc* is a discriminative pattern since its scores satisfy the thresholds. In this example we don't consider confidence intervals because the sample size is too small.

3.4 Enumeration strategy

The main practical contribution of this chapter is SSDPS, an efficient algorithm for mining statistically significant discriminative patterns. This algorithm will be presented in the next section (Section 3.5). SSDPS owes its efficiency to an original enumeration strategy of the patterns, which allows to exploit some degree of anti-monotonicity on the measures of discriminance and statistical significance.

The majority of enumeration strategies used in pattern mining algorithms make a tree-shaped enumeration (called an *enumeration tree*) over all the possible itemsets. This enumeration tree is based on *itemset augmentation*: each itemset p is represented by a node, and the itemsets $p \cup \{e\}$ (for e in I) are children of p: the

augmentation is the transition from p to $p \cup \{e\}$. If such augmentation was conduced for all $e \in I$, this would lead to enumerating multiple times the same itemset (ex: $ab \cup c = bc \cup a = abc$). Each enumeration strategy imposes constraints on the e that can be used for augmentation at each step, preventing redundant enumeration while preserving completeness. The other important component of pattern mining enumeration strategies is the use of *anti-monotonicity properties*. When enumerating frequent itemsets, one can notice that if an itemset p is unfrequent ($sup(p, D) < min\_sup$), then no super-itemsets $p' \supset p$ can be frequent (necessarily $sup(p', D) < sup(p, D) < min\_sup$). This allows to stop any further enumeration when an unfrequent itemset p is found, allowing a massive reduction in the search space [39]. As far as we know, no such anti-monotonicity could be defined on measures of discriminance or statistical significance.

The enumeration strategy proposed in SSDPS also builds an enumeration tree. However, it is based on the tidsets and not the itemsets. Each node of the enumeration tree is a tidset (with the empty tidset at the root), and the augmentation operation consists in adding a single tid: the children of node of tidset t are nodes of tidset $t \cup i$ for some $i \in \{1..n\}$. An example enumeration tree for the data of Table 3.2 is presented in Figure 3.1, with the tidset written on the top of each node. Note that the tidset is displayed with a separation of the tids from D_1 (case) and from D_2 (control). For example, consider the node represented by $\boxed{12:8}$: this node corresponds to the tidset 128 in which 12 are the positive tids, and 8 is the negative tid. The children of $\boxed{12:8}$ are $\boxed{12:68}$ (augmentation by 6) and $\boxed{12:78}$ (augmentation by 7).

Before delving deeper on the enumeration strategy that was used to construct this tree, we explain how it is possible to recover the itemsets (which are our expected outputs) from the tidsets. This is a well known problem: itemsets and tidsets are in facts dual notions, and they can be linked by two functions that form a *Galois connection* [120]. The main difference in our definition is that the main dataset can be divided into two parts ($D = D_1 \cup D_2$), and we want to be able to apply functions of the Galois connection either in the complete dataset D or in any of its parts D_1 or D_2.

Definition 3.4 (Galois connection) For a dataset $D = D_1 \cup D_2$:

- For any tidset $q \subseteq \{1..n\}$ and any itemset $p \subseteq I$, we define:

$$f(q, D) = \{i \in I \mid \forall k \in q \;\; i \in t_k\}$$

$$g(p, D) = \{k \in \{1..n\} \mid p \subseteq t_k\}$$

- For any tidset $q_1 \subseteq D_1$ and any itemset $p \subseteq I$, we define:

$$f_1(q_1, D_1) = \{i \in I \mid \forall k \in q_1 \;\; i \in t_k\}$$

$$g_1(p, D_1) = \{k \in D_1 \mid p \subseteq t_k\}$$

- For any tidset $q_2 \subseteq D_2$ and any itemset $p \subseteq I$, we define:

$$f_2(q_2, D_2) = \{i \in I \mid \forall k \in q_2 \;\; i \in t_k\}$$

$$g_2(p, D_2) = \{k \in D_2 \mid p \subseteq t_k\}$$

Note that this definition marginally differs from the standard definition presented in [120]: here for convenience we operate on the set of tids $\{1..n\}$, whereas the standard definition operates on the set of transaction $\{t_1, ..., t_n\}$.

In Figure 3.1, under each tidset q, its associated itemset $f(q, D)$ is displayed. For example for node 12:8, the itemset $f(128, D) = bci$ is displayed. One can verify in Table 3.2 that *bci* is the only itemset common to the transactions t_1, t_2 and t_8.

Finding an itemset associated to a tidset is a trivial use of the Galois connection. A more advanced use is to define a *closure operator*, which takes as input any tidset q, and returns the closed pattern that has the smallest tidset containing q.

Definition 3.5 (Closure operator) For a dataset D and any tidset $q \subseteq \{1..n\}$, the closure operator is defined as:

$$c(q, D) = g \circ f(q, D)$$

The output of $c(q, D)$ is the tidset of the closed itemset having the smallest tidset containing q.

We can similarly define $c_1(q_1, D_1) = g_1 \circ f_1(q_1, D_1)$ for $q_1 \subseteq D_1$ and $c_2(q_2, D_2) = g_2 \circ f_2(q_2, D_2)$ for $q_2 \subseteq D_2$.

Note that the standard literature on pattern mining defines the closure operator as taking an itemset as input, whereas here we define it as having a tidset as input. Replacing $g \circ f$ by $f \circ g$ gives the dual closure operator taking itemsets as input.

The basics of the enumeration have been given: the enumeration proceeds by augmenting tidsets (starting from the empty tidset), and for each tidset function f of the Galois connection gives the associated itemset. The specificity of our enumeration strategy is to be designed around statistically significant discriminative

patterns. This appears first in our computation of closure: we divide the computation of closure in the two sub-datasets D_1 and D_2. This intermediary step allows some early pruning. Second, most measures of discriminance require the pattern to have a non-zero support in D_2 (GR and ORS). The same condition apply for measures of statistical significance: in both cases we need to defer measures of interest of patterns until it has some tids in D_2.

Our enumeration strategy thus operates in two steps:

1. From the empty set, it enumerates closed tidsets containing only elements of D_1 (case group).

2. For each of those tidset containing only tids of D_1, augmentations using only tids of D_2 are generated and their closure is computed. Any subsequent augmentation of such nodes will only be allowed to be augmented by tids of D_2.

More formally, let $q \subseteq \{1..n\}$ be a tidset, with $q = q^+ \cup q^-$, where $q^+ \subseteq D_1$ and $q^- \subseteq D_2$. Then the possible augmentations of q are:

- (*Rule 1*) if $q^- = \emptyset$: q can either:
  - (*Rule 1a*) be augmented with $k \in D_1$ such that $k < min(q^+)$
  - (*Rule 1b*) be augmented with $k \in D_2$
- (*Rule 2*) if $q^- \neq \emptyset$: q can only be augmented with tid $k \in D_2$ such that $k < min(q^-)$

This enumeration mechanic is based on imposing an arbitrary ordering on the tidsets, a classical technique when enumerating itemsets. It is guaranteed to avoid enumerating duplicates.

More interestingly, we show that it allows to benefit from an anti-monotony property on the measures of statistical significance and discriminance.

Theorem 3.1 (Anti-monotonicity) Let q_1 and q_2 be two tidsets such as: $q_1^+ = q_2^+$ and $q_1^- \subset q_2^-$ (we have $q_1^+ \neq \emptyset$ and $q_2^- \neq \emptyset$). Let $p_1 = f(q_1, D)$ and $p_2 = f(q_2, D)$. Then:

1. $scr(p_1, D) > scr(p_2, D)$ with scr a discriminance measure in $\{SD, GR, ORS\}$.

2. $lci(p_1, D) > lci(p_2, D)$ with lci a low confidence interval in $\{LCI_{ORS}, LCI_{GR}\}$.

Proof: 1) For the tidset q_1, let $a = |q_1^+|$ be the number of positive tids and $c = |q_1^-|$ be the number of negative tids ($0 \leq a \leq |D_1|$, $0 \leq c \leq |D_2|$). Let

$b = |D_1| - a$, and $d = |D_2| - c$. Then SD, GR, and ORS of p_1 are estimated as follows:

$$SD(p_1, D) = \frac{a}{a+b} - \frac{c}{c+d}$$

$$GR(p_1, D) = \frac{a/(a+b)}{c/(c+d)}$$

$$ORS(p_1, D) = \frac{a.d}{b.c}$$

We have $q_1 \subset q_2$, then $|q_1| - |q_2| = x > 0$, where by definition of q_1 and q_2 those x tids are part of D_2. SD, GR, and ORS of p_2 are thus estimated as follows:

$$SD(p_2, D) = \frac{a}{a+b} - \frac{c+x}{c+d} < SD(p_1, D)$$

$$GR(p_2, D) = \frac{a/(a+b)}{(c+x)/(c+d)} < GR(p_1, D)$$

$$ORS(p_2, D) = \frac{a.(d-x)}{b.(c+x)} < ORS(p_1, D)$$

2) Please refer to the supporting document for the detailed demonstration of this part.

□

This theorem provides pruning by anti-monotonicity in our enumeration strategy: for a node having a tidset with tids both from D_1 and D_2, if the discriminance or statistical significance measures are below a given threshold, then necessarily its augmentations will also be under the threshold. Hence this part of the enumeration tree can be pruned.

Consider the node $\boxed{2{:}8}$ for example. Its associated itemset is *bcei* and $ORS(bcei, D)$ = 3/4. If the threshold is 2, then this node can be pruned and its augmentations need not be computed. This allows to significantly reduce the search space.

3.5 SSDPS: Algorithm design and implementation

In this section, we present the SSDPS algorithm. We first present in details how the enumeration strategy presented in Section 3.4 is exploited in the algorithm. We then show several techniques to improve the efficiency of the algorithm. Last, we modify the algorithm to perform heuristic search, in order to trades exhaustiveness for significantly reduced running times.

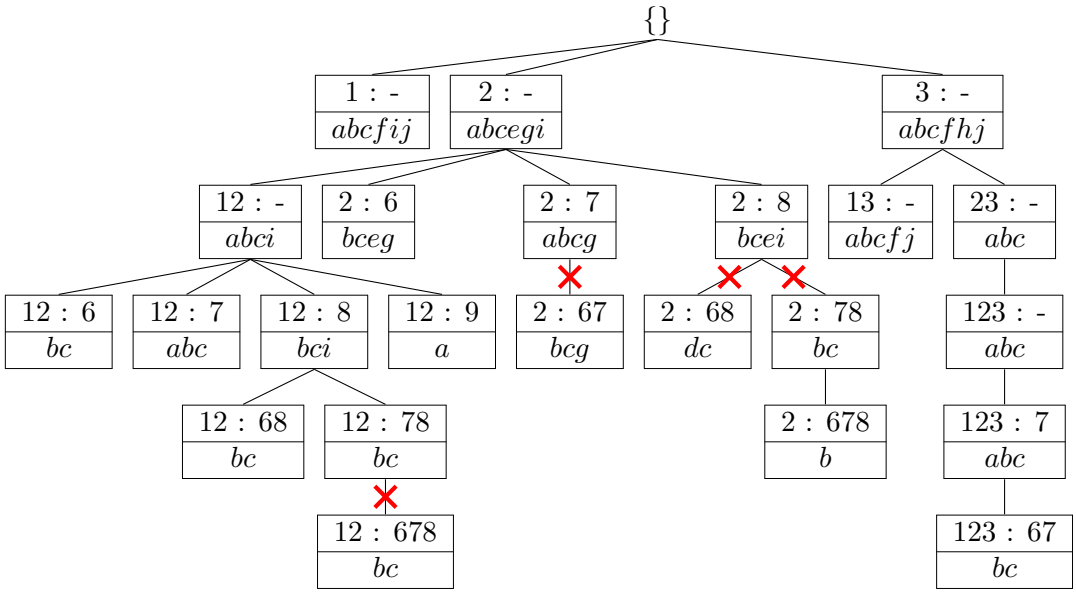


Figure 3.1: Tidset-itemset search tree

3.5.1 Exhaustive search

As mentioned in the previous section, our algorithm is based on an enumeration of the tidsets. It discovers statistically significant discriminative closed patterns.

The main procedure for enumerating tidsets is given in Algorithm 1. This procedure calls the recursive procedure *positive_expand* (Algorithm 2) to find closed frequent itemsets in the positive class. Computing discriminative patterns relies on the recursive procedure *negative_expand* (Algorithm 3).

Delving more into details, *positive_expand* (Algorithm 2) is based on the principles of the LCM algorithm [86], the state of the art for mining closed frequent itemsets. *positive_expand* takes as input the tidset t of a pattern that is closed in D_1 and a tid $e \in D_1$ that can be used to augment t. This augmentation is performed on line 1, and the pattern p associated to the augmented tidset $t^+ = t \cup \{e\}$ is computed in line 2. If $p = \emptyset$, there are no items common to all transactions of t^+ so the enumeration can stop (test of line 3). Else, we can continue the enumeration by applying *Rule 1* of enumeration presented in Section 3.4. Lines 4 to 10 apply the LCM principles of enumerating closed itemsets without redundancies (the interested reader referred to [121] Section 3.2 for a description of these principles). At this step of the enumeration, the closure is computed in D_1 (line 4). The test of line 5 verifies if the closure actually extends the tidset, requiring a further verification in line 10, and the construction of the new extended tidset (line 7).

Lines 9 to 11 implement *Rule 1a* of enumeration, allowing to grow the positive part of the tidset. Lines 12 to 13 implement *Rule 1b* of enumeration, stopping the growth of the positive part and starting to grow the negative part of the tidset.

The same logic is followed in lines 15 to 20, in the case where the tidset is not extended by the closure (test of line 10 is false).

The final expansion of the tidset is handled by *negative_expand* (Algorithm 3), that can only perform augmentations with negative tidsets. It is very similar to *positive_expand*, with several key differences. The first obvious one is that the closure is this time computed in D_2 (line 5). The second one is that only *Rule 2* of enumeration can be applied (lines 17 and 25). The third and most important difference is that because we have tidsets with positive and negative tids, we can compute discriminance as well as statistical significance measures. Hence, Theorem 3.1 can be applied to benefit from pruning by anti-monotonicity. This is done in line 4.

As an example of the execution of the algorithm, consider tidset 12. Its associated itemset is *abci* and its closure in D_1 is 12. Thus *abci* is closed in D_1. Then 12 will be combined with all tids in D_2 to find discriminative patterns. In particular, the following tidsets are created: 126, 127, 128, and 129.

Consider the tidset of 128. We have $f(128, D) = bci$ and $c_2(128, D_2) = 128$. Thus *bci* is closed in D_2. The discriminative scores of *bci* in D are: $ORS(bci, D) = 2$, $GR(bci, D) = 1.6$, $SD(bci, D) = 0.15$.

Suppose the discriminative thresholds are: $ORS = 1.5$, $GR = 1.5$ and $SD = 0.1$. *bci* is a discriminative pattern since it satisfies all given thresholds, and 128 is the tidset containing *bci*.

In contrast, 1278 does not satisfy discriminative thresholds. Thus all branches expanded from this node are pruned.

The SSDPS algorithm can discover patterns even from small tidset (upper nodes of the enumeration tree). It means that the patterns with very low support are taken into consideration.

Algorithm 1 Exhaustive search algorithm

Input: two-class dataset D, discriminative thresholds α, confidence intervals β
Output: the set of statistically significant discriminative patterns

1: transaction id set $t = \emptyset$
2: **for** each transaction id e in *positive_class* **do**
3: $\quad$ *positive_expand*(t, e, D, α, β)

Algorithm 2 Positive class expanding

Procedure $positive\_expand(t, e, D, \alpha, \beta)$

1: $t^+ \leftarrow t \cup \{e\}$
2: $p \leftarrow f(t^+, D)$
3: **if** p is not empty **then**
4: $\quad t\_ext^+ \leftarrow c_1(t^+, D_1)$
5: $\quad$ **if** $t\_ext^+ \neq t^+$ **then**
6: $\quad\quad$ **if** $max(t\_ext^+) < e$ **then**
7: $\quad\quad\quad q \leftarrow t^+ \cup t\_ext^+$
8: $\quad\quad\quad RD \leftarrow reduced\_dataset(q, D)$
9: $\quad\quad\quad$ **for** each e^+ in $D_1 \setminus q$ **do**
10: $\quad\quad\quad\quad$ **if** $e^+ < e$ **then**
11: $\quad\quad\quad\quad\quad positive\_expand(q, e^+, RD, \alpha, \beta)$
12: $\quad\quad\quad$ **for** each e^- in D_2 **do**
13: $\quad\quad\quad\quad negative\_expand(q, e^-, RD, \alpha, \beta)$
14: $\quad$ **else**
15: $\quad\quad RD \leftarrow reduced\_dataset(t^+, D)$
16: $\quad\quad$ **for** each e^+ in D_1 **do**
17: $\quad\quad\quad$ **if** $e^+ < min(t^+)$ **then**
18: $\quad\quad\quad\quad positive\_expand(t^+, e^+, RD, \alpha, \beta)$
19: $\quad\quad$ **for** each e^- in D_2 **do**
20: $\quad\quad\quad negative\_expand(t^+, e^-, RD, \alpha, \beta)$

Algorithm 3 Negative class expanding

Procedure $negative\_expand(t, e, D, \alpha, \beta)$

```
1: t⁻ ← t ∪ {e}
2: p ← f(t⁻, D)
3: if p ≠ ∅ then
4:     if check_significance(p, D, α, β) is true then
5:         t_ext⁻ ← c₂(t⁻, D₂)
6:         if t_ext⁻ ≠ t⁻ then
7:             if max(t_ext⁻) < e then
8:                 q ← t⁻ ∪ t_ext⁻
9:                 q_ext ← c(q, D)
10:                p′ ← f(q, D)
11:                if q_ext = q then
12:                    if check_significance(p′, D, α, β) is true then
13:                        output: p′
14:                RD ← reduced_dataset(q, D)
15:                for each e⁻ ∈ D₂ \ q do
16:                    if e⁻ < e then
17:                        negative_expand(q, e⁻, RD, α, β)
18:        else
19:            t_ext ← c(t⁻, D)
20:            if t_ext = t⁻ then
21:                output: p
22:            RD ← reduced_dataset(t⁻, D)
23:            for each e⁻ ∈ D₂ \ t⁻ do
24:                if e⁻ < e then
25:                    negative_expand(t⁻, e⁻, RD, α, β)
```

3.5.2 Searching the largest patterns

Exhaustive mining generates an exponential number of patterns, and specifically many redundant ones. Hence, filtering out a limited proportion of highly statistically significant patterns is important. To limited the amount of output patterns, we consider searching only largest (in size) statistically significant discriminative patterns (largest patterns in short). They are defined as follow:

Definition 3.6 (Largest statistically significant discriminative pattern) p is a largest statistically significant discriminative pattern if there does not exist any pattern p', so that $p \subset p'$ and the discriminative scores of p' are larger than the discriminative scores of p.

If p is such a largest pattern, all subsets of p will have discriminative scores smaller than discriminative scores of p. Therefore, instead of discovering all patterns which satisfy the constraints, we focus on finding only the largest patterns.

As presented in the Section 3.5.1, discovering discriminative patterns is performed by the *negative_expand* procedure which was presented in Algorithm 3. We propose in Algorithm 4 a new negative expansion procedure, *negative_expand_largest*, which replaces *negative_expand* and which allows to directly compute largest patterns. The intuition of *negative_expand_largest* is that once *positive_expand* has found a set of tids t from D_1 and a corresponding pattern p, the function will try to discover the largest extension of t with tids of D_2 that preserves the pattern p. Two cases can arise: either such extension with tids of D_2 exists and discriminance/significance measures can be computed. Or such extension does not exist: in this case we choose to output the pattern p with only its tids $t \subseteq D_1$: this is a pattern that occurs only in case, such kind of discriminative patterns are called *jumping emerging patterns* [45].

For example, consider the tidset of 13 and its corresponding pattern $abcfj$. 13 has no tid extension in D_2. Thus, $abcfj$ occurs only in D_1: it is a jumping emerging pattern. On the other hand, consider the tidset of 123 and its corresponding pattern abc. Its tid extension in D_2 is 7, no further extension in D_2 preserves pattern abc. Thus, abc is a largest discriminative pattern.

In practice, in Algorithm 4, line 1 first verifies that the tid $t \subseteq D_1$ is large enough, by comparing it to a user given threshold u. This allows to avoid output largest patterns that only cover few lines in D_1 (i.e. few individuals in case). Then the tids of D_2 that contain pattern p corresponding to t are computed and stored in t_ext^- (line 3). If t_ext^- is empty, then the pattern p corresponding to t is output, this is a jumping emerging pattern (line 5). Else we join t and t_ext^- in k, compute the closure of this extended tidset in D, and check in line 9 that the tidset k is closed. If k is closed, we know that its corresponding pattern is p: we can compute

its discriminance and significance in line 10 and if they are above thresholds output pattern p.

Algorithm 4 Negative class expanding for searching the largest pattern

Procedure $negative\_expand\_largest(t, e, D, \alpha, \beta, u)$

```
1:  if size of t ≥ u then
2:      p ← f1(t, D1)
3:      t_ext⁻ ← f2 ∘ g2(p, D2)
4:      if t_ext⁻ = ∅ then
5:          output p
6:      else
7:          k ← t ∪ t_ext⁻
8:          k_ext ← c(k, D)
9:          if k_ext = k then
10:             if check_significance(p, D, α, β) then
11:                 output p
```

With this framework the number of generated patterns is limited. In addition, the execution time is highly reduced. The reason is that the algorithm spends time only for discovering the tidsets which can generate closed patterns in the positive class, while the tasks of identifying the largest patterns are computed quickly in the negative class.

Furthermore, to make a trade off between execution time and the number of generated largest patterns, three heuristic strategies are used:

1. Reverse order of searching: the main loop (line 2 of Algorithm 1) starts with the tids of highest numerical value, and proceeds towards the tids of lowest numerical value. Recall that our enumeration strategy does not allow to enumerate a tidset containing a tid of higher value than the maximal tid of the tidset (arbitrary order to avoid duplicates in the enumeration). Starting with tids of high numerical value thus allows to make a full enumeration immediately, and discover early the largest tidsets.

2. Increase risk score thresholds: For each successful output pattern, the risk score thresholds are increased (by 0.1 for example). This strategy guarantees that the later output patterns have better risk scores than the current pattern. Moreover, with increasing risk threshold strategy, the pruning based on risk scores is even more efficient.

3. Control searching steps: When the risk scores get higher, the algorithm spend more time to find patterns which satisfy the thresholds. In this case, we impose

Table 3.4: Vertical binary data representation

Items	**Tids**
	1 2 3 4 5 6 7 8 9
a	1 1 1 0 0 0 1 0 1
b	1 1 1 1 0 1 1 1 0
c	1 1 1 0 0 1 1 1 0
d	0 0 0 1 1 0 0 1 1
e	0 1 0 1 0 1 0 1 1
f	1 0 1 0 1 0 1 0 0
g	0 1 0 1 1 1 1 0 1
h	0 0 1 0 1 1 1 1 1
i	1 1 0 1 1 0 0 1 0
j	1 0 1 1 1 1 0 0 1
Class	1 1 1 1 1 0 0 0 0

a limit on the number of enumeration steps, in order to control the running time of program. In particular, if the algorithm cannot discover any patterns which have risk scores better than the current pattern after a given number of steps, the algorithm is forced to stop.

3.5.3 Implementation

The SSDPS algorithm uses vertical data format [122, 41] combined with a binary data representation to improved its performances. In this format, each row represents an item and columns correspond to tids. The value 1 at position (i, j) indicates that the item i is presents in the transaction having tid j. In contrast, 0 indicates that item i is absent in the transaction having tid j. Considering again the data of Table 2, the vertical binary data format is illustrated in Table 4. Each item of Table 2 is transformed into a row in Table 4. Consider item a for example, in the original data, it is present in tidset 01268, and then transformed as a vector of bits (the first row) in Table 4.

The benefits of this data representation are: 1) The task of computing support is simpler and faster. We only need tidset to compute the support of an itemset. 2) The vector of bits (bitset) representation allows to efficiently compute support of itemsets, using bitset or AVX2 AND operations. 3) We can easily distinguish the positive and negative tids in a tidset. This helps us to estimate the discriminative scores and confidence intervals effectively.

The performance of the SSDPS algorithm relies on the computation of 2 functions: $f()$ (compute associated itemset of a tidset) and $c()$ (compute closure operator

of a tidset). Both functions need to compute the intersection of two sets. With integer data presentation this operator spends $O(max(n, m))$ iterations, where n and m are the size of the two sets. Thus, the time required for each task of computing associated itemset (or closure operator) is $O(I * max(n, m))$, where I is the number of items in dataset. In this study, we use the dataset reduction technique [40] to decrease the number of rows, i.e. the number of items I (function $reduced\_dataset$). With the use of this technique, the number of items is significantly reduced after each step of searching.

3.6 Experiments and results

In this section, we first present a two-step framework to find high-order SNP combinations in case-control datasets. We then apply this framework to several synthetic and real variant datasets. All experiments have been conducted on a laptop with Intel(R) Core(TM) i7-4600U CPU @ 2.10GHz, 16GB memory and Linux operating system.

3.6.1 Two-step framework

In genetic variant datasets, each SNP has three genotypes which are here considered as the items. Since the amount of genotypes is very large, using all genotypes to find combinations is infeasible. In addition, many individual genotypes are no really meaningful. For example, the genotypes that have very high frequency or that occur more in control group than in case group are not very interesting. These genotypes are considered as noise since they can be combined with many discriminative pattern without decreasing their score. Thus, discarding these genotypes is important.

To effectively search multiple SNPs combinations, two-step approaches are investigated [23, 106, 107, 108]. Specifically, [23] proposed MSCD algorithm to discover SNPs combinations. In the first step, MSCD selects candidate SNPs according to energy distribution difference of all SNPs. Then, in the second step, it uses a pruning-tree search to find SNPs combinations. Similarly, [106] proposed an algorithm which first runs k-means clustering algorithm on the set of all SNPs and then selects candidates from each cluster. These candidates are examined to find the SNPs combinations. With the same strategy, epiMiner algorithms [107] uses Co-Information Index(CII) to rank contributions of individual SNPs to the phenotype in the first step. To search SNPs interactions within the retained SNPs, in the second step, epiMiner sequentially builds combinations and test their p_values. On the other hand, the approach of EDCF [108] is different. It is based on clustering of relatively frequent items. First, three groups of genotypes are created: frequent genotypes in cases, frequent genotypes in controls and the remaining genotypes.

Then, items in the three groups are constructed sequentially to find high-order SNPs interactions. The significance of the final combinations are evaluated by Pearson's χ^2 test.

Similar with these approaches, for detecting the interesting SNP combinations which occur frequently in the case group but less frequently in the control group, we propose to use a two-step approach. However, the first step of our method is different: we use *p_value* and support of genotype in the control group (denoted *control_support*) to select candidate genotypes. In particular, if a is the *p_value* threshold and b the control support threshold, we select genotypes which have $p\_value \leq a$ and $control\_support \leq b$. The reason is that the *p_value* guaranties that the selected candidates are significant, while the control support is used to eliminate very common genotypes. These genotypes are then used to find the largest statistically significant discriminative patterns using the SSDPS algorithm.

3.6.2 Experiments on synthetic datasets

3.6.2.1 Evaluation of pruning efficiency

To evaluate the pruning efficiency of the SSDPS algorithm, we create a dataset including 260 items and 100 transactions. In this dataset, 50 transactions belong to the positive class and 50 transactions belong to the negative class. The values of items in the positive and negative classes are randomly set to 0 or 1.

We then use two approaches to discover statistically significant discriminative patterns: 1) perform SSDPS exhaustive search with pruning strategy and 2) perform SSDPS exhaustive search without pruning strategy. Both approaches use the same parameters: $OR = 2$, $RR = 2$, $ARR = 0.2$, and $LCI_{OR} = 1$.

As the result, the search space is effectively reduced when using risk measures and confidence interval. In particular, without pruning strategy SSDPS checks 24,793,469 nodes to find 14,530 statistically significant discriminative patterns. With pruning strategy, SSDPS examines 3,406,007 nodes to discover the same amount of patterns. Note that this amount of pruned nodes is only counted in the negative expand function where the pruning strategy is applied.

3.6.2.2 Evaluation of the two-step framework

For evaluating the effectiveness of the SSDPD algorithm in the two-step framework, we create six synthetic datasets. For all datasets, the number of individuals is set to 100: half belong to the positive class and half belong to the negative class. The number of items are set to 1000, 2000, 4000, 6000, 8000, and 10000 respectively. The item values in the positive and negative classes are randomly set to 0 or 1. To simulate practical situations, we set a density to 0.33 (the density has been

Table 3.5: Summary of three variant datasets

Data	**Case**	**Control**	**SNPs**
AMD	96	50	103611
BC	1045	1463	15436
T2D (chromosome 16)	1991	1500	15309

approximated with real SNP datasets). In each dataset, five statistically significant discriminative patterns of size 2, 4, 6, 8, and 10 have been inserted.

Then we test the two approaches:

1. Exhaustive approach: all SNPs are considered;

2. Two-step approach, as described previously, with a *p_value* threshold set to 0.1 and *control_support* threshold set to 100% (the support parameter is thus not used in this experiment).

In all cases, the five patterns have been found. As shown in Figure 3.2a, the execution times of the two-step approach is approximately one order of magnitude faster than the exhaustive search. The total number of patterns generated by the two-step approach is more than one order of magnitude smaller than the number of patterns generated by the exhaustive search (Figure 3.2b). These first experiments show that the two-step approach is very efficient compared to the exhaustive approach: relevant patterns can be found in a shorter time.

3.6.3 Experiments on real datasets

In this subsection, we present the experiments for identifying high-order SNP combinations associated with diseases on three real genetic variant datasets.

3.6.3.1 Dataset summary

The three datasets are the following: Age-Related Macular Degeneration (AMD) [123], Breast Cancer (BC) [124] and Type 2 Diabetes (T2D) [116]. With T2D, we only use chromosome 16 which contains 3 significant SNPs associated to this disease. The summary of the three datasets is shown in Table 3.5.

Based on previous studies [116, 124, 123], each disease is linked to a few SNPs, as illustrated in Table 3.6.

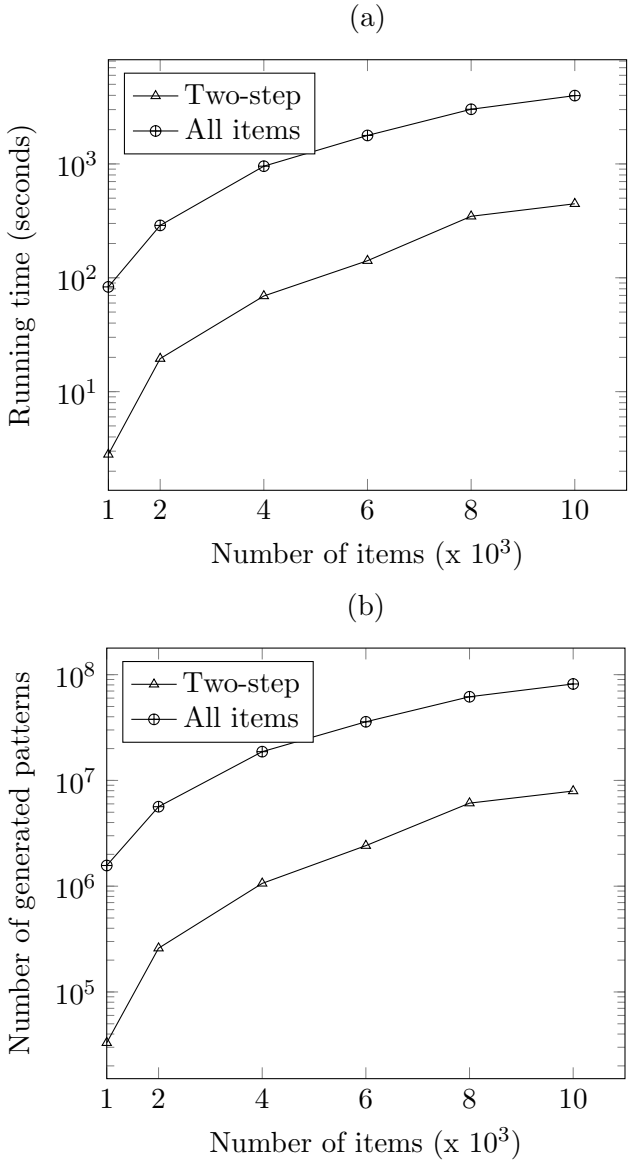


Figure 3.2: Results of two approaches on simulated datasets. (a) running times, (b) number of generated patterns.

Table 3.6: Individual SNPs associated with diseases

Disease	**SNP**	**Chromosome**	**Gene**
AMD	rs1329428	1	CFH
	rs380390	1	CFH
BC	rs2107732	7	CCM2
	rs4986790	9	TLR4
	rs2285374	11	VPS11
	rs7313899	12	OR6C4
	rs2879097	17	MEL18
	rs2822558	21	ABCC13
	rs2230018	23	UTX
T2D	rs7193144	16	FTO
	rs8050136	16	FTO
	rs9939609	16	FTO

Table 3.7: Pattern generated on AMD dataset with different control support

Support	rs1329428_2	rs380390_0	Both	Patterns	Time(s)
30%	21	5	5	29	16
50%	59	9	9	299	145
70%	45	2	2	307	287

rs1329428_2, rs380390_0: the number of patterns containing these SNPs. Both: the number of patterns including both SNPs. Patterns: the total number of patterns generated. Time: the execution time in second.

3.6.3.2 Experiment on AMD dataset

To search SNPs combinations associated with AMD the two-step framework is used with two sets of parameters:

- Set 1: a fixed *p_value* threshold at 0.001 and three *control_support* thresholds: 30%, 50% and 70%.
- Set 2: a fixed *control_support* at 30% and three *p_value* thresholds: 0.005, 0.01, and 0.05.

As the results, patterns including SNP rs1329428 and rs380390 are reported in all cases. Table 3.7 summarizes the results of the SSDPS algorithm with parameters tuned according to Set 1 (variation of *control_support*).

To analyze specific pattern set and compare our result with EDCF and MSCD we present the list of patterns output with $p\_value = 0.001$ and $control\_support = 30\%$ in Table 3.8. In this specific case, SSDPS generates 29 patterns from size 2 to 4. From these 29 patterns, 21 patterns contain SNP rs1329428_2, 5 patterns contain SNP rs380390_0 and 5 patterns have both of them. Note also the very short execution time (16 seconds).

In comparison with EDCF and MSCD, SSDPS is more efficient. According to [23], EDCF spent 2,800 seconds to discover the top 20 significant SNP combinations which include 1 pattern containing disease SNPs. On the other hand, MSCD took 77 seconds to identify 27 significant patterns of size ranging from 2 to 4. In which, 11 patterns contain rs1329428_2 and 3 patterns contain rs380390_0. Most of these patterns are of size of 2 SNPs. In addition, there is no pattern in this set that contains both disease SNPs. Note that, the execution time of MSCD is fast because the number of SNPs after filtering is very small. More precisely, MSCD selects 32 sets of SNPs, each of them having only 28 significant SNPs. These candidate SNPs are then used for discovering combinations.

With larger *control_support* thresholds (50% and 70%), the number of output patterns increase, as well as the number of patterns having these two SNPs.

Table 3.9 summarizes the results of the SSDPS algorithm with parameters tuned according to Set 2 (variation of the *p_value*). Again, in all cases, patterns including the two interesting SNPs are output. Furthermore, the total number of output patterns is limited, whatever the *p_value*. However, the execution times are more important. This is mainly due to the number of selected SNPs during the filtering step. In that case, the number of selected SNPs with *p_value* of 0.005, 0.01, and 0.05 are 315, 778 and 4470, respectively.

3.6.3.3 Experiment on BC dataset

Note that in the following experiments, we can no longer compare with EDCF and MSCD, as they have not been applied to this data. For the Breast Cancer experiment, parameters are set as follows: $p\_value = 0.005$ and $control\_support = 20\%$, 25% and 30%. The *p_value* threshold is larger than in the AMD tests since the BC number of genotypes is much smaller. With these filter conditions, 5 out of 7 SNPs associated with BC are selected in step 1. 2 interesting SNPs are not selected for the second step.

Then, the SSDPS algorithm is run and output patterns from which 2 SNPs related to BC are present. More specifically, with $p\_value = 0.005$ and $control\_support = 20\%$, 50 patterns are generated, in which there are 3 patterns that contain rs2230018_AC, and 2 patterns that contain rs2107732_TC. The top 10 out of 50 patterns having the highest risk scores are shown in Table 3.10. These patterns have a very low fre-

Table 3.8: Patterns generated on the AMD dataset with $p\_value = 0.001$ and $control\_support = 30\%$

SNP combinations	|Pa|	|Po|
rs10504339_1 rs10483226_0	31	2
rs1329428_2 rs10504121_1	35	1
rs6598991_0 **rs1329428_2**	33	0
rs6598991_0 **rs1329428_2** rs273185_0	32	0
rs404199_2 rs6598991_0 **rs1329428_2** rs273185_0	31	0
rs718309_1 **rs1329428_2 rs380390_0**	32	1
rs288247_2 rs3844556_1	35	1
rs7185187_2 rs3844556_1	32	2
rs10488343_1 **rs1329428_2 rs380390_0**	31	2
rs1329428_2 rs380390_0 rs287020_2	33	1
rs287020_2 rs3844556_1	32	2
rs10520583_2 **rs1329428_2**	33	0
rs1329428_2 rs10254116_0	37	2
rs7185187_2 rs1363688_0 rs1394608_0	35	2
rs962848_2 rs1363688_0	31	1
rs10488343_1 rs4894840_1	33	2
rs1363688_0 **rs1329428_2**	45	3
rs1363688_0 **rs1329428_2** rs1394608_0	41	1
rs1363688_0 **rs1329428_2** rs1394608_0 **rs380390_0**	30	1
rs1363688_0 rs1394608_0 rs287020_2	35	2
rs1363688_0 **rs1329428_2** rs287020_2	34	0
rs1363688_0 **rs1329428_2** rs1394608_0 rs287020_2	31	0
rs7185187_2 **rs1329428_2**	42	2
rs7185187_2 **rs1329428_2 rs380390_0**	31	1
rs7185187_2 rs1363688_0 **rs1329428_2**	31	1
rs10507949_2 rs1363688_0 **rs1329428_2** rs1394608_0	33	0
rs1146382_2 **rs1329428_2**	34	1
rs82159_2 **rs1329428_2**	30	1
rs1329428_2 rs6730141_1	30	1

|Pa|, |Po|: number of individuals in case, control

Table 3.9: Patterns generated on AMD dataset for different p_values

P_value	rs1329428_2	rs380390_0	Both	Patterns	Time(s)
0.005	22	4	4	35	120
0.01	25	5	5	46	465
0.05	25	3	3	51	1750

rs1329428_2, rs380390_0: the number of patterns containing these SNPs. Both: the number of patterns including both SNPs. Patterns: the total number of patterns generated. Time: the execution time in second.

Table 3.10: Top 10-highest risk scores patterns of Breast Cancer

SNP combinations	|Pa|	|Po|
rs12842916_TC rs209373_AG rs3788890_TG rs5955139_TC	22	0
rs2884554_AA rs5951332_AG	21	0
rs1968987_AT rs2856705_AG rs5955139_TC	25	1
rs12842916_TC rs6580942_CC rs7066252_GC	24	1
rs4827909_TC rs6580942_CC rs7066252_GC	24	1
rs1048369_TC rs1129980_AC rs179008_TA rs1968987_AT rs7884806_AG	24	1
rs2242801_GG rs2498323_AG rs5951332_AG	23	1
rs12842916_TC rs2707164_AG rs4907817_AA	22	1
rs4907817_AA rs5955353_AG	22	1
rs1132201_AG rs1266719_CG rs1385699_TC **rs2107732_TC**	21	1

|Pa|, |Po|: number of individuals in case, control

quency. For instance, the highest occurrence of the pattern containing rs2107732_TC is 21 out of 1045 case individuals and 1 out of 1463 control individuals. Other SNPs combinations have also a low frequency. In this pattern set, a particular pattern occurs in 22 case individuals but is absent in the control group. It has 4 SNPs which are located in difference genes. An interesting point is that all SNPs belong to chromosome X and each of them has a very low *p_value*. This kind of information are pertinent clues for clinician to investigate new hypotheses.

3.6.3.4 Experiment on T2D dataset

According to *p_value* of all individual genotypes, three SNPs associated with T2D have *p_value* less than 0.02. Thus, in order to consider these SNPs in mining com-

Table 3.11: Top 10-highest risk scores patterns of T2D

SNP combinations	|Pa|	|Po|
rs10775354_AC rs12051393_GT	41	1
rs4985114@CC rs1684568_GT	28	1
rs16966656@AC rs1684568_GT	27	1
rs1684568_GT rs10500444@AC	26	1
rs2370096_AG rs1078621_AA rs6499591_CG	39	0
rs8045058_GG rs2370096_AG rs6499591_CG	34	0
rs1684568_GT rs9939012_CT	32	0
rs153084_AG rs2370096_AG rs6499591_CG	31	0
rs1684568_GT rs12051393_GT	31	0
rs231619_CC rs10775354_AC	28	0

|Pa|, |Po|: number of individuals in case, control

binations we choose $p\_value = 0.02$ and $control\_support = 20\%$ to filter candidates. With these parameters, the SSDPS algorithm discovers all three SNPs associated with T2D. However, the frequency of patterns containing these SNPs is very low. Particularly, there are 2 patterns including all of three SNPs associated with T2D. The occurrences of these patterns in case group are 23 and 22 out of 1991 case samples, respectively. While both of them exist in only 1 out of 1500 control samples.

Similarly to the previous experiments on AMD and BC datasets, we also use $p\_value = 0.005$ and $control\_support = 20\%$ to filter candidate genotypes. With these parameters, all three SNPs associated with T2D are excluded from the set of candidates. Consequently, the set of generated patterns cannot contain them. However, the output patterns include many interesting ones. Many have high difference of frequency in the two classes. These patterns include many significant individual SNPs. Consider the top of 10-highest risk scores patterns of T2D which are shown in Table 3.11. These patterns are built from 12 significant individual genotypes. In the literature, they are not known as SNPs associated with T2D. However, some of them have remarkable properties. For example, rs231619_CC exists in 62 case individuals but is absent in control (its frequency is approximately 1.8%), and this SNP is located in an unknown gene region. These SNPs have very low frequency. In other word, they are rare relatively to T2D. The list of 12 significant SNPs of the top 10-highest scores patterns is illustrated in Table 3.12.

Table 3.12: Individual SNPs in the 10-highest risk scores patterns of T2D

SNP_genotype	Case	Control	Position	Gene
rs8045058_GG	424	253	3631656	DNASE1
rs231619_CC	62	0	4124765	unknown
rs10775354_AC	288	163	6747691	RBFOX1
rs4985114_CC	464	284	8687186	ABAT
rs16966656_AC	167	61	9899720	GRINA2
rs153084_AG	475	287	13234017	SHISA9
rs2370096_AG	373	101	22466298	LOC653786
rs1684568_GT	105	11	34307201	unknown
rs12051393_GT	88	8	57653933	unknown
rs10500444_AC	381	233	60142165	unknown
rs1078621_AA	471	245	67336497	CDH1
rs6499591_CG	160	63	71464505	ZFHX3
rs9939012_CT	410	249	73031343	unknown

3.7 Conclusion

In this chapter we propose an algorithm, called SSDPS, that efficiently finds statistically significant discriminative patterns from a case-control SNP dataset. The strategy directly uses relative risk measures and confidence intervals as anti-monotonic properties to efficiently prune the less important patterns during the mining process. In addition, a two-step framework is applied to speed-up the process without significant loss in quality.

Experiments on real dataset show that the SSDPS algorithm efficiently detects high-order SNP combinations. Most of known SNPs related to diseases belong to the patterns. Other interesting patterns are also generated and might be of interest for further investigation. Furthermore, contrary to other methods, the number of generated pattern is small, allowing direct interpretation by clinicians.

However, choosing appropriate thresholds to select individual genotypes (step 1) is still difficult, and requires a good expertise from the users. One perspective of this work is to investigate methods to suggest good thresholds to the user based on characteristics of the dataset.

Another perspective is to analyze further the discovered patterns. In this regard, we are working on visualization strategies allowing to present our patterns to biologists in order to quickly focus on the most promising patterns for a biological investigation.

A last direction for future work, as hinted in the introduction, is to frame the

significant pattern discovery problem as a multiple hypothesis problem, in order to further remove uninteresting patterns. The goal will be to have a computationally efficient solution for this problem, possibly using parallel computation.

3.8 Technical details for proof of Theorem 1

In this section, we present the detailed proof of theorem 1.

Let recall the presence and absence of pattern p in D. It is presented in a 2x2 contingency table as follow:

Table 3.13: A 2x2 contingency table of a pattern in case-control data

	Presence	Absence	Total
Case	a	b	D_1
Control	c	d	D_2

Let $q_i - g(q_i, D)$ and $q_j - g(q_j, D)$ be two TI-pairs in the same equivalent class. We have $q_i \subset q_j$ and $p_i = g(q_i, D)$, $p_j = g(q_j, D)$. Let $|q_j| - |q_i| = 1$ be a minimal difference between q_j and q_i we have:

The lower confidence intervals of ORS of p_i ans p_j are given:

$$LCI\_ORS(p_i, D) = \exp\left(ln(\frac{ad}{bc}) - 1.96\sqrt{\frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{1}{d}}\right) \tag{3.17}$$

$$LCI\_ORS(p_j, D) = \exp\left(ln(\frac{a(d-1)}{b(c+1)}) - 1.96\sqrt{\frac{1}{a} + \frac{1}{b} + \frac{1}{c+1} + \frac{1}{d-1}}\right) \tag{3.18}$$

The lower confidence intervals of GR of p_i and p_j are given:

$$LCI\_GR(p_i, D) = \exp\left(ln(\frac{a(c+d)}{c(a+b)}) - 1.96\sqrt{\frac{1}{a} - \frac{1}{a+b} + \frac{1}{c} - \frac{1}{c+d}}\right) \tag{3.19}$$

$$LCI\_GR(p_j, D) = \exp\left(ln(\frac{a(c+d)}{(c+1)(a+b)}) - 1.96\sqrt{\frac{1}{a} - \frac{1}{a+b} + \frac{1}{c+1} - \frac{1}{c+d}}\right) \tag{3.20}$$

For all integers $a, b, c > 0$ and all integers $d > 1$ we want to demonstrate that: (3.17) > (3.18) and (3.19) > (3.20).

3.8.1 Proof of $LCI\_ORS(p_i, D) > LCI\_ORS(p_j, D)$

The lower confidence interval of ORS of p_i and p_j are given:

$$LCI\_ORS(p_i, D) = \exp\left(ln(\frac{ad}{bc}) - 1.96\sqrt{\frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{1}{d}}\right)$$

$$LCI\_ORS(p_j, D) = \exp\left(ln(\frac{a(d-1)}{b(c+1)}) - 1.96\sqrt{\frac{1}{a} + \frac{1}{b} + \frac{1}{c+1} + \frac{1}{d-1}}\right)$$

First of all we can rewrite some terms and give their bounds.

$$\alpha = \frac{1}{a} + \frac{1}{b} \quad \text{so} \quad 0 < \alpha \leq 2$$

$$0 < \frac{1}{d} \leq \frac{1}{2}, \quad 0 < \frac{1}{d-1} \leq 1$$

$$0 < \frac{1}{c} \leq 1, \quad 0 < \frac{1}{c+1} \leq \frac{1}{2}$$

Now, we calculate the difference:

$$LCI\_ORS(p_i, D) - LCI\_ORS(p_j, D)$$

$$= ln\frac{d(c+1)}{c(d-1)} + 1.96\left(\sqrt{\alpha + \frac{1}{c+1} + \frac{1}{d-1}} - \sqrt{\alpha + \frac{1}{c} + \frac{1}{d}}\right)$$

The first term is clearly positive, the last one is the hardest to treat. With a little trick we can give another expression for this difference:

$$\sqrt{\alpha + \frac{1}{c+1} + \frac{1}{d-1}} - \sqrt{\alpha + \frac{1}{c} + \frac{1}{d}}$$

$$= \frac{\frac{1}{c+1} - \frac{1}{c} + \frac{1}{d-1} - \frac{1}{d}}{\sqrt{\alpha + \frac{1}{c+1} + \frac{1}{d-1}} + \sqrt{\alpha + \frac{1}{c} + \frac{1}{d}}}$$

$$= \frac{\frac{-1}{c(c+1)} + \frac{1}{d(d-1)}}{\sqrt{\alpha + \frac{1}{c+1} + \frac{1}{d-1}} + \sqrt{\alpha + \frac{1}{c} + \frac{1}{d}}}$$

The denominator is always positive. We can notice that if $d \leq c+1$ then the numerator is also positive so $LCI\_ORS(p_i, D) > LCI\_ORS(p_j, D)$. We must treat the other case.

Let us suppose $d \geq c+2$. Let us rewrite the difference:

$$LCI\_ORS(p_i, D) - LCI\_ORS(p_j, D) = \ln \frac{d(c+1)}{c(d-1)} - 1.96 \frac{\frac{1}{c(c+1)} - \frac{1}{d(d-1)}}{\sqrt{\alpha + \frac{1}{c+1} + \frac{1}{d-1}} + \sqrt{\alpha + \frac{1}{c} + \frac{1}{d}}}$$

In this case we know that the fraction is strictly positive, so we have to maximize it to lower bound the difference. We can remove some terms:

$$LCI\_ORS(p_i, D) - LCI\_ORS(p_j, D) \geq \ln \frac{d(c+1)}{c(d-1)} - 2 \frac{\frac{1}{c(c+1)}}{\sqrt{\frac{1}{c+1} + \frac{1}{d-1}} + \sqrt{\frac{1}{c} + \frac{1}{d}}}$$

It gives a general expression for the lower bound. The problem is it depends on two variables, so the idea is to removed d. We get quickly:

$$\ln \frac{d(c+1)}{c(d-1)} = \ln \left(1 + \frac{1}{d-1}\right) + \ln \left(1 + \frac{1}{c}\right) \geq \ln \left(1 + \frac{1}{c}\right)$$

Moreover

$$\sqrt{\frac{1}{c+1} + \frac{1}{d-1}} + \sqrt{\frac{1}{c} + \frac{1}{d}} \geq \sqrt{\frac{1}{c+1}} + \sqrt{\frac{1}{c}}$$

Then we get,

$$LCI\_ORS(p_i, D) - LCI\_ORS(p_j, D) \geq \ln \left(1 + \frac{1}{c}\right) - 2 \frac{\frac{1}{c(c+1)}}{\sqrt{\frac{1}{c+1}} + \sqrt{\frac{1}{c}}} \tag{3.21}$$

This lower bound depends only on c but studying directly this function is not simple. That's why, we can first simplify it.

$$LCI\_ORS(p_i, D) - LCI\_ORS(p_j, D) \geq \ln \left(1 + \frac{1}{c}\right) - 2 \frac{\frac{1}{c(c+1)}}{2\sqrt{\frac{1}{c+1}}}$$

$$\geq \ln \left(1 + \frac{1}{c}\right) - \frac{1}{c\sqrt{c+1}}$$

$$\geq \ln\left(1+\frac{1}{c}\right) - \frac{1}{c\sqrt{c}}$$

Let us introduce the function f defined by:

$$\forall x > 0, f(x) = \ln\left(1+\frac{1}{x}\right) - \frac{1}{x\sqrt{x}}$$

We can derive this function

$$\begin{aligned} f'(x) &= \frac{-\frac{1}{x^2}}{1+\frac{1}{x}} + \frac{3}{2}\frac{1}{x^2\sqrt{x}} \\ &= \frac{-1}{x(x+1)} + \frac{3}{2}\frac{1}{x^2\sqrt{x}} \\ &= \frac{-2x\sqrt{x}+3(x+1)}{2x^2\sqrt{x}(x+1)} \end{aligned}$$

This denominator is always positive. Let us look at $-2x\sqrt{x}+3(x+1) \leq 0$:

$$-2x\sqrt{x}+3(x+1) \leq 0 \longleftrightarrow 3 \leq x(2\sqrt{x}-3)$$

The function $x \mapsto x(2\sqrt{x}-3)$ is clearly a growing function. As when $x = 4$ the inequality is true, it is true for all $x \geq 4$. It shows that f' is negative on $[4, +\infty]$. So f is decreasing on the same interval. However $lim_{x\to+\infty} f(x) = 0$. Hence, we know that $LCI\_ORS(p_i, D) \geq LCI\_ORS(p_j, D)$ for all $c \geq 4$.

The three cases $c = 1, 2$ and 3 have finally to be treated, but the function f cannot be used for that. For the last steps we will use the initial bound (1):

$$LCI\_ORS(p_i, D) - LCI\_ORS(p_j, D) \geq \ln\left(1+\frac{1}{c}\right) - 2\frac{\frac{1}{c(c+1)}}{\sqrt{\frac{1}{c+1}}+\sqrt{\frac{1}{c}}}$$

If $c = 1$

$$LCI\_ORS(p_i, D) - LCI\_ORS(p_j, D) = \ln 2 - \frac{1}{1+\sqrt{\frac{1}{2}}} \geq 0$$

If $c = 2$

$$LCI\_ORS(p_i, D) - LCI\_ORS(p_j, D) = \ln\frac{3}{2} - \frac{\frac{1}{3}}{\sqrt{\frac{1}{2}}+\sqrt{\frac{1}{3}}} \geq 0$$

If $c = 3$

$$LCI\_ORS(p_i, D) - LCI\_ORS(p_j, D) = \ln \frac{4}{3} - \frac{\frac{1}{6}}{\frac{1}{2} + \sqrt{\frac{1}{3}}} \geq 0$$

Eventually, for all $c \geq 1$ we have $LCI\_ORS(p_i, D) > LCI\_ORS(p_j, D)$. Gathering the cases $d \leq c+1$ and $d \geq c+2$, we have $LCI\_ORS(p_i, D) > LCI\_ORS(p_j, D)$ for all $a, b, c, d \in \mathbb{N}^*$ with $d \geq 2$.

3.8.2 Proof of $LCI\_GR(p_i, D) > LCI\_GR(p_j, D)$

Lower confidence intervals of GR of p_i and p_j are given:

$$LCI\_GR(p_i, D) = \exp\left(ln(\frac{a(c+d)}{c(a+b)}) - 1.96\sqrt{\frac{1}{a} - \frac{1}{a+b} + \frac{1}{c} - \frac{1}{c+d}}\right)$$

$$LCI\_GR(p_j, D) = \exp\left(ln(\frac{a(c+d)}{(c+1)(a+b)}) - 1.96\sqrt{\frac{1}{a} - \frac{1}{a+b} + \frac{1}{c+1} - \frac{1}{c+d}}\right)$$

We want to approve $LCI\_GR(p_i, D) > LCI\_GR(p_j, D)$. Similar with proof of lower confidence interval of ORS, we want to demonstrate that $LCI\_GR(p_i, D) - LCI\_GR(p_j, D) > 0$, we rewrite this inequality as follow:

$$g_4 = ln(\frac{c+1}{c}) - 1.96\frac{\frac{1}{c(c+1)}}{\sqrt{\frac{1}{a} - \frac{1}{a+b} + \frac{1}{c} - \frac{1}{c+d}} + \sqrt{\frac{1}{a} - \frac{1}{a+b} + \frac{1}{c+1} - \frac{1}{c+d}}} > 0$$

We set $\alpha = \frac{1}{a} - \frac{1}{a+b} > 0$ and we have:

$$g_4 > g_2 = \ln\left(1 + \frac{1}{c}\right) - 1.96\frac{\frac{1}{c(c+1)}}{\sqrt{\frac{1}{c} - \frac{1}{c+d}} + \sqrt{\frac{1}{c+1} - \frac{1}{c+d}}}$$

We try to delete d in this lower bound:

$$\frac{1}{c} - \frac{1}{c+d} \geq \frac{1}{c} - \frac{1}{c+1} = \frac{1}{c(c+1)}$$

$$\frac{1}{c+1} - \frac{1}{c+d} \geq \frac{1}{c+1} - \frac{1}{c+1} = 0$$

The second inequality involves problems with the lower-bound g_2 (this lower bound will not be positive). So, in the next part we assume $d \geq 2$. Thus:

$$\frac{1}{c} - \frac{1}{c+d} \geq \frac{1}{c} - \frac{1}{c+2} = \frac{2}{c(c+2)}$$

$$\frac{1}{c+1} - \frac{1}{c+d} \geq \frac{1}{c+1} - \frac{1}{c+2} = \frac{1}{(c+1)(c+2)}$$

We get:

$$g_2 \geq g_1 = \ln\left(1+\frac{1}{c}\right) - 1.96\frac{\frac{1}{c(c+1)}}{\sqrt{\frac{2}{c(c+2)}} + \sqrt{\frac{1}{(c+1)(c+2)}}}$$

We can simplify a little:

$$g_1 = \ln\left(1+\frac{1}{c}\right) - 1.96\sqrt{\frac{c+2}{c(c+1)}}.\frac{1}{\sqrt{c}+\sqrt{2(c+2)}}$$

We have to show that this function is positive for all $c \geq 1$. However, this is not the case for $c = 1$ and $c = 2$ but we can show that it is true for all $c \geq 3$. Hence, the issue is that g_1 is not directly easy to analyze, so we have to provide a easier lower bound but it implies some singular cases to treat.

$$g_1 \geq \ln\left(1+\frac{1}{c}\right) - 1.96\frac{1}{\sqrt{c}}.\sqrt{\frac{c+2}{c+1}}.\frac{1}{\sqrt{c}+\sqrt{2c}}$$

$$\geq \ln\left(1+\frac{1}{c}\right) - \frac{\beta}{c}.\sqrt{1+\frac{1}{c+1}}$$

$$\geq \ln\left(1+\frac{1}{c}\right) - \frac{\beta}{c}.\sqrt{1+\frac{1}{c}}$$

Where $\beta = \frac{1.96}{1+\sqrt{2}} \simeq 0.812$. Let us use $f(c)$ as new lower bound. We can calculate:

$$f'(c) = -\frac{1}{c(c+1)}\left(1 - \beta\sqrt{1+\frac{1}{c}}\left(1+\frac{3}{2c}\right)\right) = -\frac{1}{c(c+1)}.g(c)$$

We want to show that the lower bound function f is positive. Actually, we are going to show that this function is decreasing beyond some point. First we can notice that g is increasing:

$$U : x \mapsto 1 - \beta\sqrt{1+x}\left(1 + \frac{3}{2}x\right) \text{ and } V : x \mapsto \frac{1}{x} \text{ are clearly decreasing on } \mathbb{R}^+$$

So $g = U \circ V$ is increasing on $\mathbb{R}^+$. Moreover $g(9) > 0$, then for all $c \geq 9$, $g(c) \geq 0$. It implies that for all $c \geq 9$, $f'(c) \leq 0$. Nonetheless, we notice that $f(9) \geq 0$ and $f(c) \mapsto 0$ when $c \mapsto \infty$. As f is decreasing for $c \geq 9$, it means that f is positive for $c \geq 9$. We sum up (for $d \geq 2, c \geq 9$):

$$LCI\_GR(p_i, D) - LCI\_GR(p_j, D) \geq g_4 > g_1 \geq f(c) \geq 0$$

The cases $(c = 1, d > 1)$ and $(c = 2, d > 1)$ need to be treated. We will use the g_2 function, we recall:

$$g_2 = \ln\left(1 + \frac{1}{c}\right) - 1.96\frac{\frac{1}{c(c+1)}}{\sqrt{\frac{1}{c} - \frac{1}{c+d}} + \sqrt{\frac{1}{c+1} - \frac{1}{c+d}}}$$

Let us set $c = 2$, then:

$$g_2 = \ln\left(\frac{3}{2}\right) - \frac{1.96}{6}\frac{1}{\sqrt{\frac{1}{2} - \frac{1}{2+d}} + \sqrt{\frac{1}{3} - \frac{1}{2+d}}}$$

$$\text{The function } P : x \mapsto \ln\left(\frac{3}{2}\right) - \frac{1.96}{6}.x \text{ and } Q : x \mapsto \frac{1}{\sqrt{\frac{1}{2} + x} + \sqrt{\frac{1}{3} + x}}$$

are decreasing and $R : x \mapsto -\frac{1}{d+x}$ is increasing, so that $d \mapsto g_2 = P \circ Q \circ R$ is increasing. Unfortunately g_2 with $c = 2, d = 2$ is negative but with $c = 2, d = 3$ is positive. It means that the inequality is true in the case $(c = 2, d = 3)$ because g_2 is increasing, but not for the case $(c = 2, d = 2)$. In the same way, we can show that the inequality is true for the case $(c = 1, d \geq 4)$ but not for the remaining cases: $(c = 1, d = 2)$ and $(c = 1, d = 3)$.

The inequality $LCI\_GR(p_i, D) - LCI\_GR(p_j, D) > 0$ is true except the cases $(c = 1, d = 2)$, $(c = 1, d = 3)$ and $(c = 2, d = 2)$. However, we have $c + d = |D_2|$, is a large integer. Thus these remaining cases cannot happen in practice.

Chapter 4

SNP visualization

This chapter presents a graphical tool which is used to visualize the combinations of genetic variants in a whole genome. It is an efficient and easy-to-use genetic-analysis tool that supports biologists in their search for relations between genetic variant combinations and interesting phenotypes.

4.1 Introduction

Discriminative patterns are used to present GWAS results to an expert who will take decisions based on the analysis results. Existing methods usually generate a large number of patterns which include many redundant ones. In addition, patterns are presented in long textual lists. This can be very complicated for experts to analyze the knowledge represented by this list of patterns. Particularly, in GWAS analysis, biologists may want to present a limited number of patterns in the form of real SNP combinations with other related biological information. Available software such as PLINK [125] and SNPsys [126] can only search and visualize single or pair of SNPs interactions. The discriminative patterns present interactions between many more SNPs, which can be interesting for biologists. Thus it is necessary to have an interactive graphical tool to present them.

This chapter presents a graphical tool, named *SNPVisual*, to visualize the discriminative patterns. They are represented as easy to understand genetic variant combinations with their biological context. This tool can be used to visualize the combinations of genetic variants in various real variant datasets (we tested it on human and plant datasets). SNP combinations are illustrated in chromosomes according to their positions and various related information such as genes and quantitative trait locus (QTL) regions. With this tool one can easily observe which groups of SNPs are interacting in a whole genome.

Table 4.1: SNPs dataset example

Individual	**SNP**					**Label**
	SNP_1	SNP_2	SNP_3	SNP_4	SNP_5	
1	AT	GC	AT	GC	AG	Case
2	AT	GC	AA	CC	AG	
3	AT	CC	AT	GG	AG	
4	AA	GG	AA	GC	AA	Control
5	AT	GG	AT	GC	AA	
6	TT	GC	AA	CC	GG	

The rest of this chapter is organized as follows: Next section briefly introduces the architecture of the software. The different methods which are used to tackle each step of the software are then detailed. At the end, a summary of the results and future research directions are presented.

4.2 Overall architecture

To dectect and visualize interesting SNP combinations, the software conducts several steps. These steps are divided into two main parts: pattern detection and pattern visualization. Fig. 4.1 illustrates the overall architecture of the software. Visualization of interesting SNP combinations on whole genome is a real challenge since it is not so easy to present many patterns. Thus this task is needed to be decompressed into several steps.

4.2.1 Pattern detection

Pattern detection aims to discover high-order SNP combinations which satisfy given constraints such as risk measures and statistical significance thresholds. This process consists of several steps which have been presented in the Chapter 3. However, to follow easily, we briefly recall these tasks.

The input of the software is a case/control dataset which is represented by a matrix. In this matrix rows are individuals and columns are SNPs. Each SNP has 2 alleles which form three genotypes. Table 4.1 presents an example of SNP data with 6 individuals which belong to two groups. Each individual contains 5 SNPs.

To be used in discriminative pattern mining algorithm, the SNP data is transformed into a binary matrix. In the binary matrix, columns correspond to SNP genotypes and rows correspond to individuals which have labels of case or control. The data transformation task is done by step 1 (Formatting) of this software.

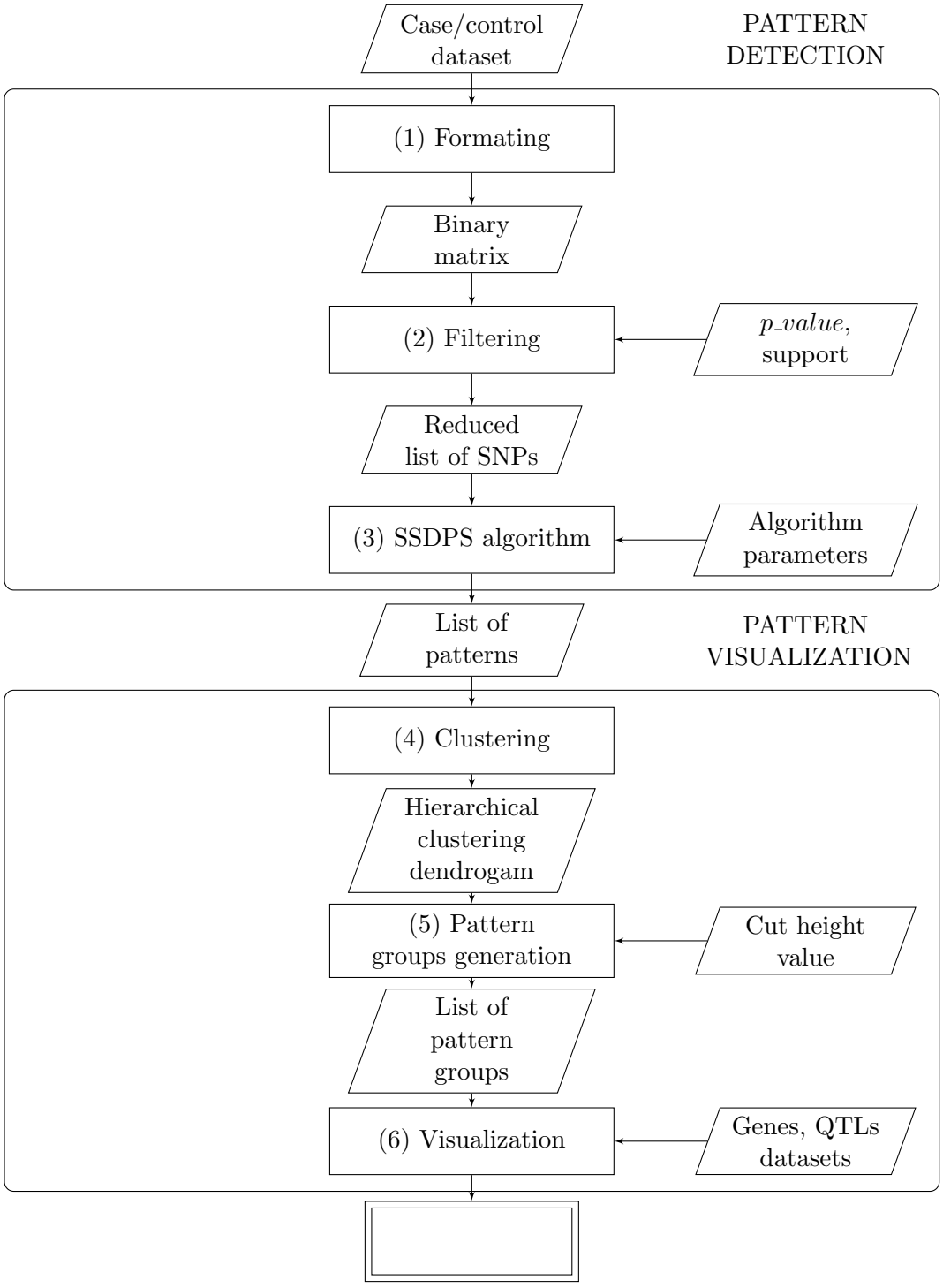


Figure 4.1: The software architecture

Using all SNP genotypes to find combinations is infeasible since the number of SNP genotypes is very large. Thus filtering (step 2) is needed to select the most interesting individual SNP genotypes based on *p_value* and support in the control group. These candidate SNP genotypes are used to find combinations by the SSDPS algorithm (step 3). The SSDPS algorithm exploits multiple risk measures and confidence intervals to discover statistically significant discriminative patterns. In addition, various parameters can be used to trade off execution time and the number of generated patterns. After this step, a set of statistically significant discriminative patterns is generated. The number of generated patterns is often large to manually analyze. In addition, they are represented in the form of long textual texts. This may be complicated to understand the knowledge that is related to the generated patterns.

4.2.2 Pattern visualization

In order to present generated patterns as genetic variant combinations in real genomic datasets, several steps are required.

Clustering (step 4) aims to regroup similar discriminative patterns. This task helps to reduce the number of analysis patterns. To find similar patterns, hierarchical clustering algorithm is proposed to use. This algorithm automatically calculates the similarity of patterns and partitions them into appropriate groups.

Pattern group generation (step 5) aims to represent the "pattern groups" which are found by the previous step. Each pattern group represents a set of similar discriminative patterns. To compute the representative of pattern groups, different methods such as union, intersection and majority are used.

Visualization (step 6) provides various interactive functions to visualize the SNP combinations on a specific genome with other related biological data. SNPs are demonstrated on chromosomes in graphical style according to their real position. In addition, several functionalities such as overview, zoom and detail are provided to help users to analyze the SNP combinations.

The specific methods used to tackle these steps are detailed in the following sections.

4.3 Clustering

The SSDPS algorithm generates a set of discriminative patterns which include some redundancies. In order to limit the number of analysis patterns we propose to group the similar patterns into sensible groups.

Each pattern consists of a set of items. A *pattern group* is a set of similar patterns.

Table 4.2: Popular distance measures

Measure	**Equation**
Euclidean distance	$\|\|a-b\|\|_2 = \sqrt{\sum_{i=1}^{2}(a_i-b_i)^2}$
Manhattan distance	$\|\|a-b\|\|_1 = \sum_{i=1}^{2}\|a_i-b_i\|^2$
Maximum distance	$\|\|a-b\|\|_\infty = max\|a_i-b_i\|$

For example, given a set of 5 discriminative patterns: $p_1 = \{a, b, c, d\}$, $p_2 = \{a, b, c, f, g\}$, $p_3 = \{a, b, d, h\}$, $p_4 = \{d, g, i, j, k\}$, $p_5 = \{i, j, k, h\}$, example pattern groups can be: $g_1 = \{p_1, p_2, p_3\}$, $g_2 = \{p_4, p_5\}$, $g_3 = \{p_1, p_4\}$, $g_4 = \{p_4, p_5\}$.

Intuitively, two patterns are similar if they share a large number of items. For example, p_4 and p_5 share 3 items thus they are more similar than p_1 and p_4 which share only 1 item.

In order to find pattern groups which contain a set of similar patterns, we propose to use clustering algorithms which automatically calculate the similarity of patterns and partition them into appropriate groups.

Clustering algorithms aim to organize data into sensible groups according to the similarity of data. A formal definition of the clustering problem can be stated as follows: Given n objects, find k groups based on the similarity of these objects (with $k < n$). The similarity of the objects is often calculated by distance measures such as Euclidean and Manhattan distance. The equations of some popular distance metrics for two-dimensional space are illustrated in Table 4.2.

There is a wide variety of algorithms for clustering data. Among them, K-means [127] is the most popular and the simplest one. K-means finds all clusters simultaneously by partitioning the data. To perform clustering by K-means, two important parameters are required: number of clusters K and distance measure. Choosing an appropriate K is the most difficult task. There is no perfect mathematical criterion existing for this task.

On the other hand, hierarchical clustering algorithms [128] build a binary tree of the data that successively merges similar groups of points. The binary tree can be built in 2 ways: each point of data is a cluster at the beginning, and the most similar pair of clusters are merged to form a hierarchical cluster. Or all data points together are considered as one cluster at the beginning and each cluster is recursively divided into smaller clusters. Fig 4.2 shows an example of a hierarchical clustering with 7 objects. Hierarchical clustering algorithms are widely used in practice since they only require a measure of similarity between groups of data points. In addition, the hierarchical tree provides a useful summary of the data.

In this study, we use hierarchical clustering algorithms to classify similar discriminative patterns into sensible groups.

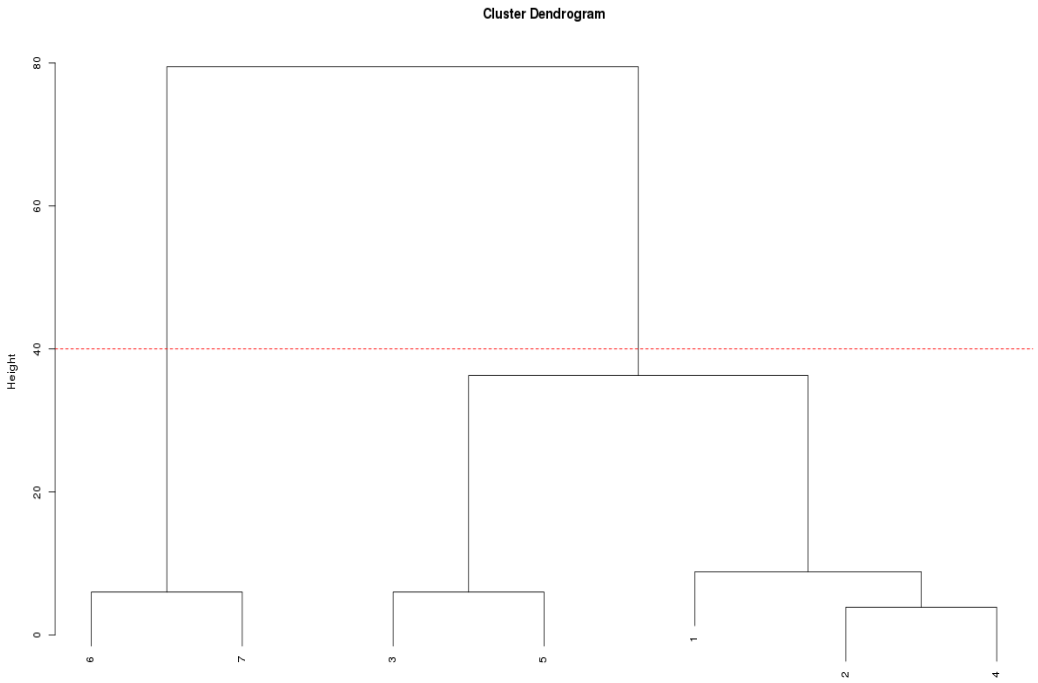


Figure 4.2: Example of hierarchical clustering

4.4 Pattern groups generation

Hierarchical clustering algorithm organizes the set of discriminative patterns in a binary tree. Each leaf is a discriminative pattern and each branch is equivalent to a group of similar discriminative patterns. The clustering algorithm allows to cut the tree into different sub-clusters based on a given threshold value. Each generated sub-cluster is a pattern group.

As discussed in the previous section, each pattern group is a set of its discriminative pattern members. It means that the pattern group contains a larger number of items than its members. It is out of scope of this thesis to find good representatives of pattern groups. However, for sake of visualization we propose to use different methods to represent a pattern group by a set of items. In particular, the representative of a pattern group can be created by union, intersection or majority of all individual items which belong to all discriminative pattern members of the pattern group.

Let g be a pattern group containing s discriminative patterns.

The union of pattern group g is defined by:

$$uni(g) = p_1 \cup p_2 \cup ... \cup p_s$$

For example, $uni(g_1) = \{a, b, c, d, f, g, h\}$, $uni(g_2) = \{d, g, i, j, k, h\}$

The intersection of pattern group g is defined by:

$$inter(g) = p_1 \cap p_2 \cap ... \cap p_s$$

For example, $inter(g_1) = \{a, b\}$, $inter(g_2) = \{i, j, k\}$

The majority of pattern group includes items which have frequency larger than a given threshold β. The majority of pattern group g is defined by:

$$major(g) = \{i \in (p1 \cup p2 \cup ... \cup p_s) \mid fre(i) \geq \beta\}$$

where $fre(i)$ is the percentage of number of patterns that contain item i over the total patterns of the pattern group.

For example, suppose $\beta = 70\%$, the majority g_1 and majority g_2 are: $major(g_1) = \{a, b, c, d\}$, $major(g_2) = \{i, j, k\}$.

4.5 Visualization

This section presents different principles and functionarities which are used to visualize SNP combinations on whole genome.

4.5.1 Visualization principle

To visualize the SNP combinations in whole genome, two principles are used to design our graphical tool.

The first principle is based on work on information visualization [129] which includes different tasks such as overview, zoom and details on demand. Overview task aims to see overall patterns and trends. On the other hand, zoom task aims to see a smaller subset of the data. Usually there are some portion of data which are interesting for the users. Thus to enable users to control the zoom focus and the zoom factor, an interactive zooming tool is needed. A good zooming tool helps users to preserve their sense of position and context. Details on demand aims to see values of patterns when interactively selected. This task allows to select an interesting item or group of items and get the highest level of information.

The second principle is to base our visualization on representation understood by biologists. In biological context, SNPs are often illustrated in specific chromosomes according to their positions. The available tools allow users to see an overview of

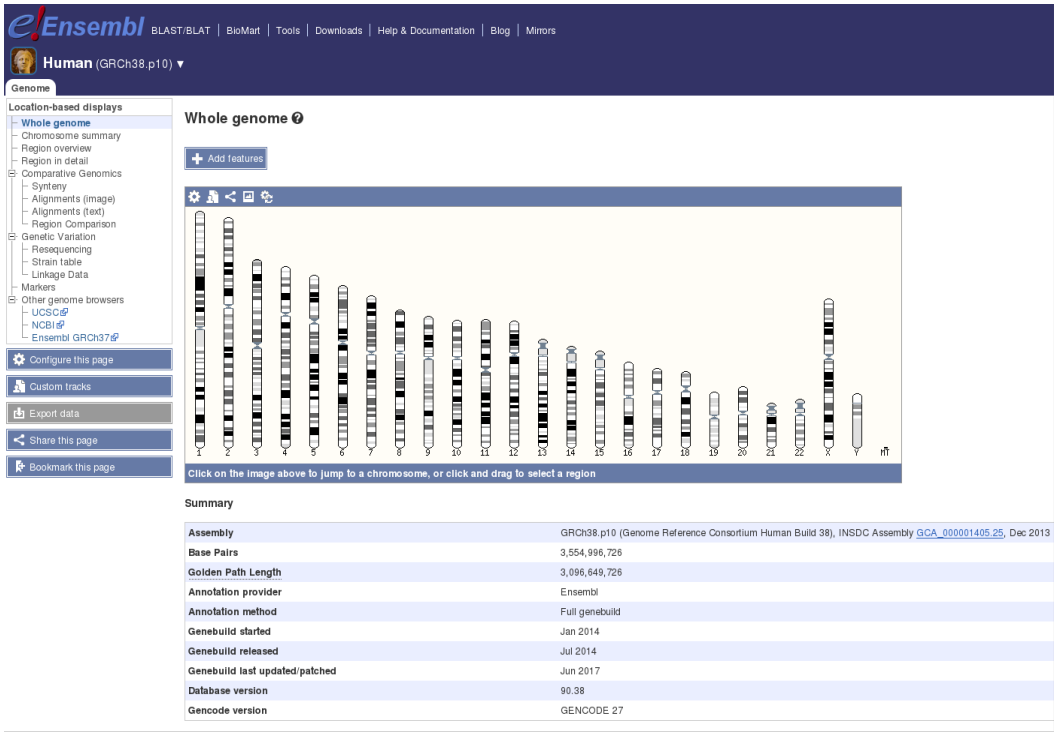


Figure 4.3: Example of a whole genome overview

whole genome or focus on an interesting chromosome region with other related biological information. For example, European Bioinformatics Institute (EMBL-EBI) provides an online tool (http://www.ensembl.org) to visualize the whole genome. This online tool allows users to view all chromosomes of a genome or focus on a specific interesting chromosome region. In addition, various related biological information are also illustrated in the selected region. For instance, Fig 4.3 shows an overview of all chromosomes of human genome while Fig 4.4 presents a short region on chromosome 2.

4.5.2 Visualization functionalities

To implement the graphical tool we use the Shiny package which is a web application framework for R. Results of pattern clustering and SNP combinations are presented to the users through an interactive graphical user interface (GUI). This GUI offers a series of effective visualizations to explore the generated discriminative patterns. The GUI can run as a desktop application or as a web application inside a web

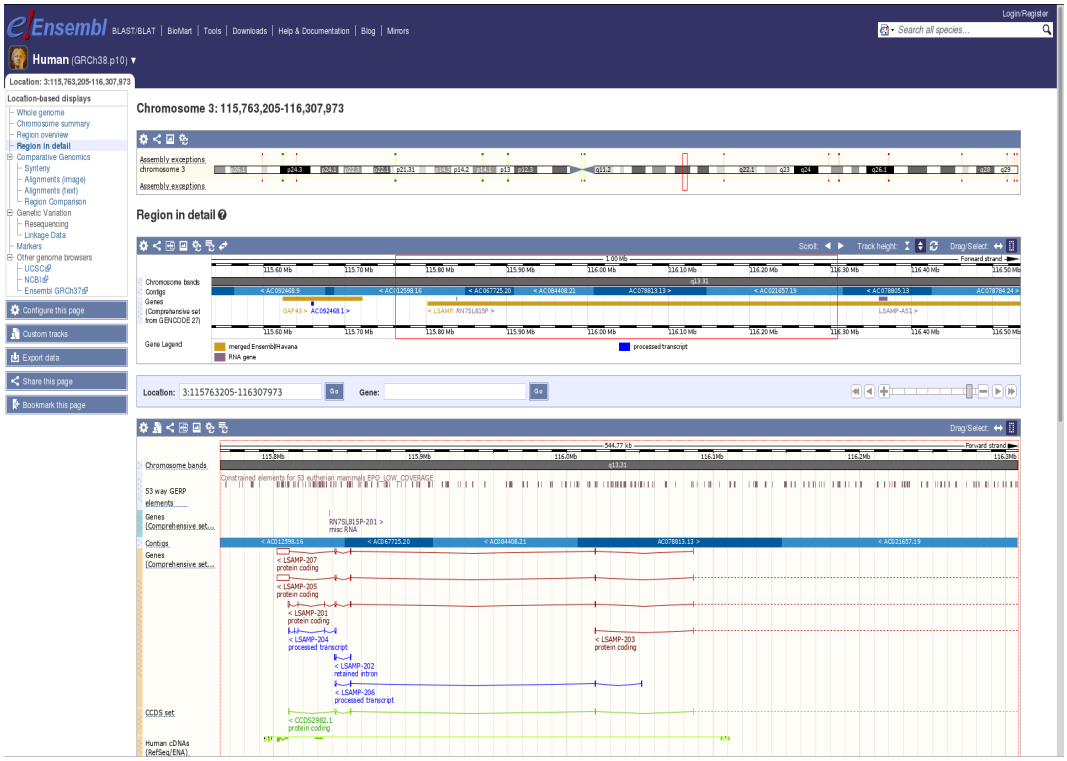


Figure 4.4: Example of zoomed region on a specific chromosome

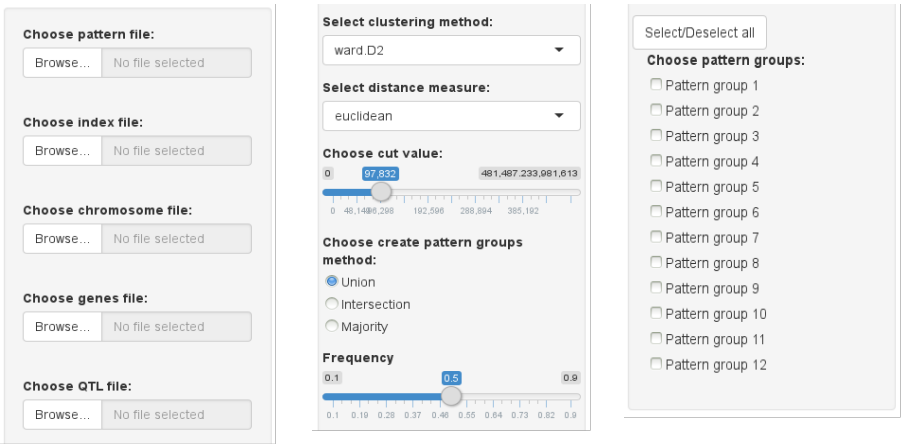
browser.

The main interface of the graphical tool is divided into two parts. The left panel provides space to set up parameters while the right panel is a graphical representation of the pattern groups.

4.5.2.1 Parameters set up

This tool is designed to visualize various genetic variant datasets. Thus it allows to load different data related to analysis patterns. The left panel of the tool provides functions to load data and set up parameters related to clustering and visualization. Fig 4.5 illustrates the left panel of this tool. This panel consists three groups of controls.

Input data parameters (Fig 4.5(a)) includes different file upload controls to load data such as patterns, chromosomes, genes, QTLs. Depending on the analysis data, users have to load appropriate data. For instance, with plaint dataset genes and QTLs data are provided while human dataset only contains genes data.



(a) Input data (b) Clustering and generating pattern groups (c) Choosing pattern groups

Figure 4.5: Left panel of the graphical tool

Clustering and generating pattern groups parameters (Fig 4.5(b)) supplies methods to cluster patterns and create the representation of pattern groups. In particular, various clustering methods and distance measures are provided. A slide bar is given to select threshold value to cut the hierarchical tree to generate sub-clusters. Note that each sub-cluster is a pattern group. Three methods are also provided to create the representation of pattern groups: union, intersection and majority. With majority option, users have to choose the frequency threshold to generate pattern groups.

After all required data is loaded and patterns are regrouped, a set of pattern groups is created according to the given thresholds. These generated pattern groups are shown in the *Choose pattern groups* area (Fig 4.5(c)) where users can select some interesting ones to present on the chromosomes.

4.5.2.2 Graphical presentation

The graphical presentation part consists of several tabs. Each tab corresponds to a visualization function. These functions allow to visualize the discriminative patterns such as clustering similar discriminative patterns, representing pattern groups on chromosomes and show detail of selected pattern groups.

Clustering: This function provides a useful tool to regroup patterns into pattern groups with hierarchical clustering algorithms. Various hierarchical clustering methods and distance measures are provided to cluster the patterns. Fig. 4.6 shows an

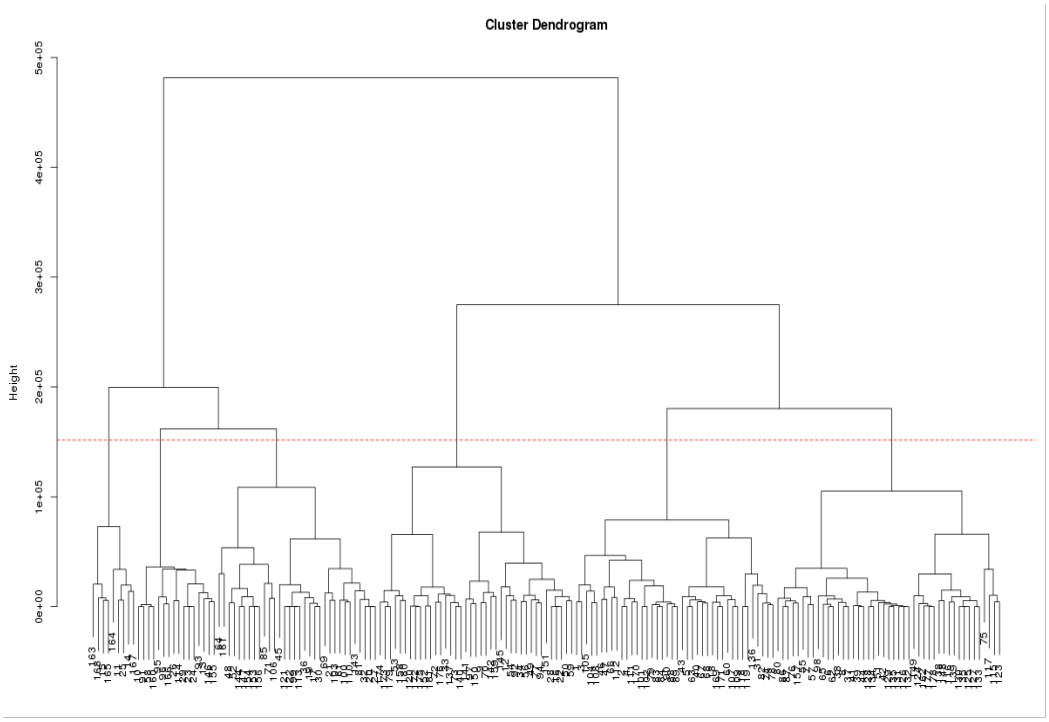


Figure 4.6: Hierarchical clustering dendrogram

example of the hierarchical clustering dendrogram. It visually illustrates the groups of patterns which have similar sets of SNPs.

To generate pattern groups, one can choose an appropriate threshold value to cut the hierarchical tree. Each generated sub-cluster is considered as a pattern group (a set of similar discriminative patterns). For example, with the threshold value of the Fig 4.6, 6 sub-clusters (equivalent to 6 pattern groups) are created. Note that one can easily adjust the cut threshold value to get an appropriate number of pattern groups. In addition, to created representatives of the pattern groups three methods are provided: union, intersection and majority.

Visualization: This function provides an interactive tool to draw SNP combinations on the whole genome. With the pattern groups which are created in the previous step, one can select which pattern groups to show on the chromosomes. For example, Fig 4.7 shows a visualization of SNP combinations on whole chromosomes. To easily observe the interaction of SNPs, each pattern group is assigned a unique color. The SNPs are drawn on the chromosomes based on their real positions with

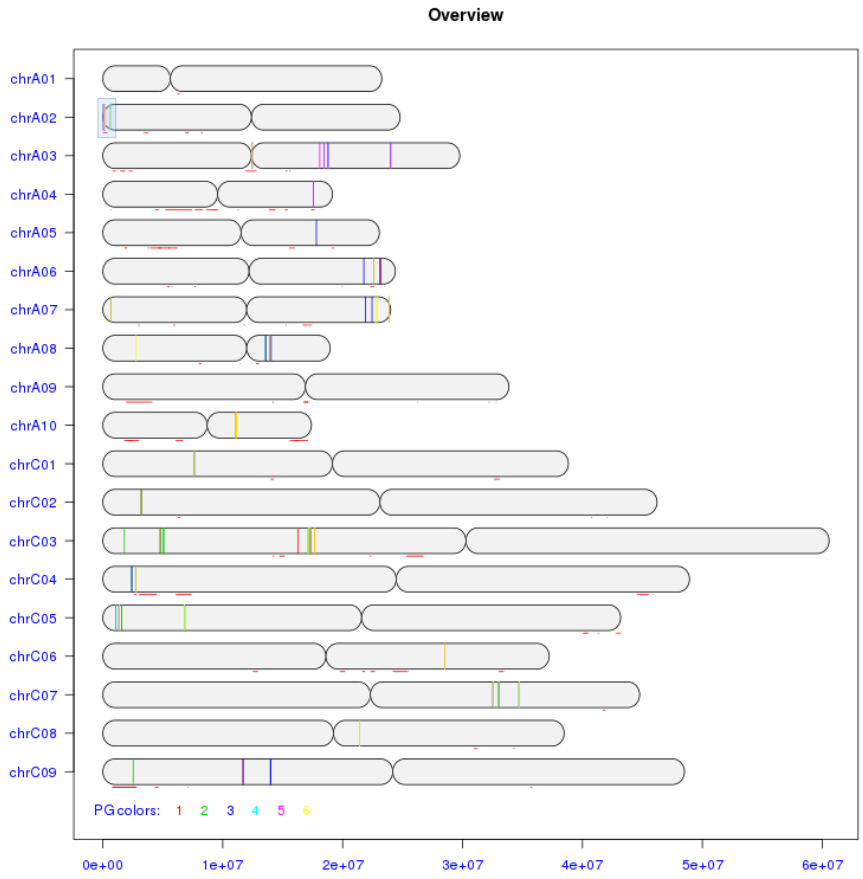


Figure 4.7: SNP combinations overview

the colors of pattern groups that contain them.

The zoom function allows users to focus on a specific region of chromosome. This function provides a variety of information that is related to the selected region such as SNPs, genes and QTL regions. For instance, Fig 4.8 illustrates a zoomed region of chromosome A02. This region shows four SNPs with other related information. One QTL region with its name and covered region (red line) is illustrated in the lower area of chromosome while genes are shown in the upper area of the chromosome.

Detail: This function provides detailed information related to the selected pattern groups. Each pattern group contains a list of SNPs which have many related information such as genotype, position and chromosome. For detail analysis purpose, these features are fully displayed in this function. For example, Fig 4.9 shows

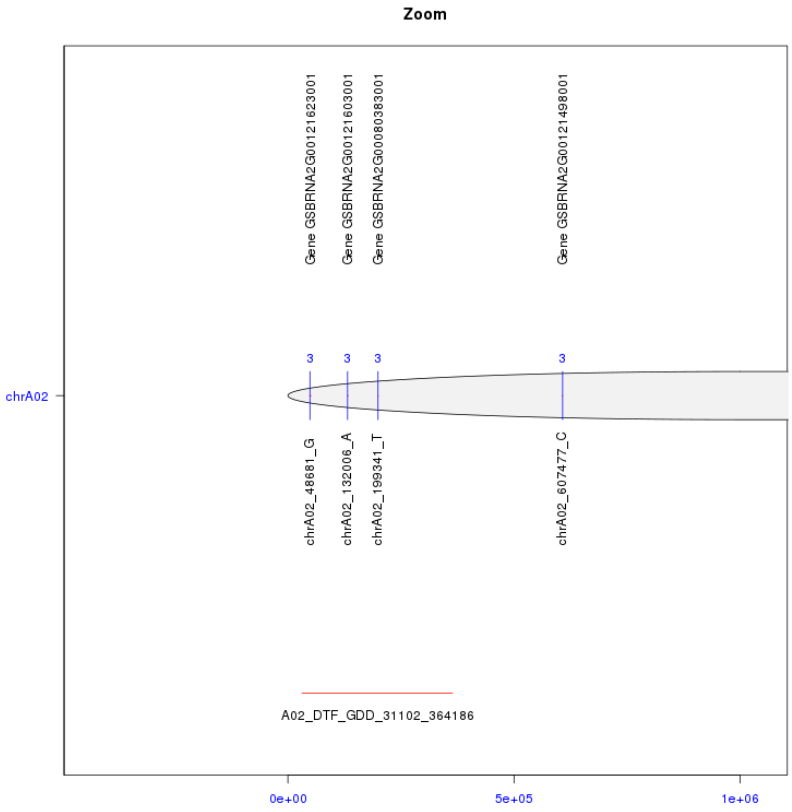


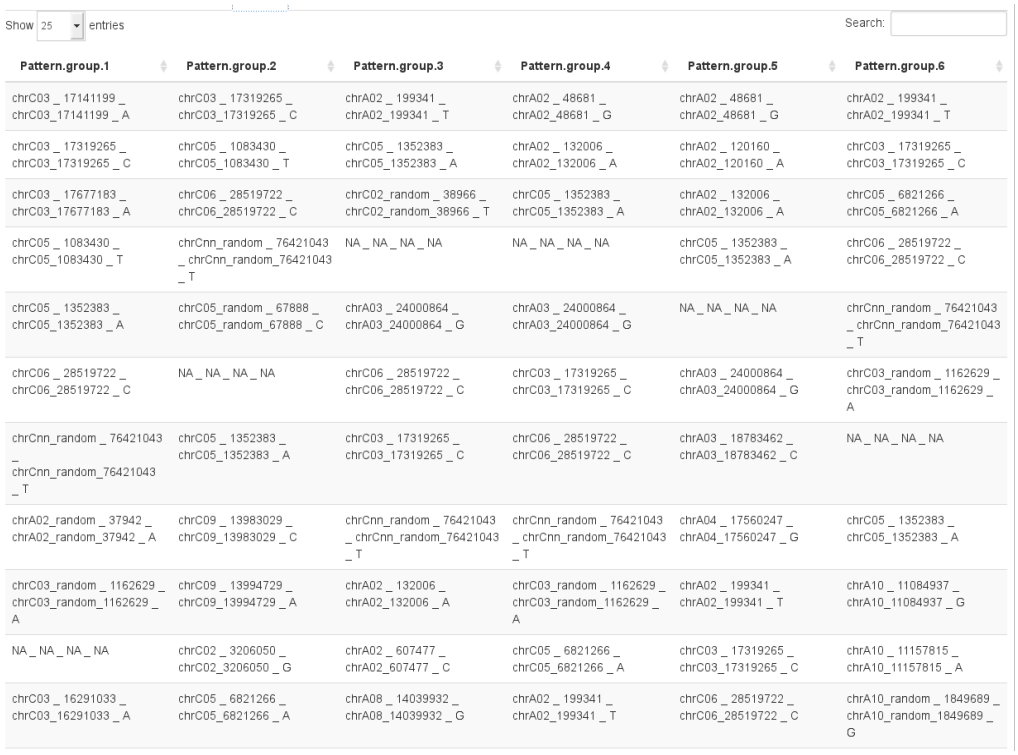
Figure 4.8: Zoom in a short region on chromosome A02

details of 6 pattern groups. Each pattern group consists of a list of SNPs with detail related information such as genotype, position, chromosome. This function also provides utilities to sort or filter pattern groups.

4.6 Related works

Visualizing SNP combinations in the whole genome has widespread attention in bioinformatics. There exist many genetic-analysis tools to discover and visualize the significant SNPs.

Among them, PLINK [125] is the most popular one. This is a tool set for genome wide association studies based on statistical methods. For case/control GWAS, it



Show 25 entries Search:

Pattern.group.1	Pattern.group.2	Pattern.group.3	Pattern.group.4	Pattern.group.5	Pattern.group.6
chrC03 _ 17141199 _ chrC03_17141199 _ A	chrC03 _ 17319265 _ chrC03_17319265 _ C	chrA02 _ 199341 _ chrA02_199341 _ T	chrA02 _ 48681 _ chrA02_48681 _ G	chrA02 _ 48681 _ chrA02_48681 _ G	chrA02 _ 199341 _ chrA02_199341 _ T
chrC03 _ 17319265 _ chrC03_17319265 _ C	chrC05 _ 1083430 _ chrC05_1083430 _ T	chrC05 _ 1352383 _ chrC05_1352383 _ A	chrA02 _ 132006 _ chrA02_132006 _ A	chrA02 _ 120160 _ chrA02_120160 _ A	chrC03 _ 17319265 _ chrC03_17319265 _ C
chrC03 _ 17677183 _ chrC03_17677183 _ A	chrC06 _ 28519722 _ chrC06_28519722 _ C	chrC02_random _ 38966 _ chrC02_random_38966 _ T	chrC05 _ 1352383 _ chrC05_1352383 _ A	chrA02 _ 132006 _ chrA02_132006 _ A	chrC05 _ 6821266 _ chrC05_6821266 _ A
chrC05 _ 1083430 _ chrC05_1083430 _ T	chrCnn_random _ 76421043 _ chrCnn_random_76421043 _ T	NA _ NA _ NA _ NA	NA _ NA _ NA _ NA	chrC05 _ 1352383 _ chrC05_1352383 _ A	chrC06 _ 28519722 _ chrC06_28519722 _ C
chrC05 _ 1352383 _ chrC05_1352383 _ A	chrC05_random _ 67888 _ chrC05_random_67888 _ C	chrA03 _ 24000864 _ chrA03_24000864 _ G	chrA03 _ 24000864 _ chrA03_24000864 _ G	NA _ NA _ NA _ NA	chrCnn_random _ 76421043 _ chrCnn_random_76421043 _ T
chrC06 _ 28519722 _ chrC06_28519722 _ C	NA _ NA _ NA _ NA	chrC06 _ 28519722 _ chrC06_28519722 _ C	chrC03 _ 17319265 _ chrC03_17319265 _ C	chrA03 _ 24000864 _ chrA03_24000864 _ G	chrC03_random _ 1162629 _ chrC03_random_1162629 _ A
chrCnn_random _ 76421043 _ chrCnn_random_76421043 _ T	chrC05 _ 1352383 _ chrC05_1352383 _ A	chrC03 _ 17319265 _ chrC03_17319265 _ C	chrC06 _ 28519722 _ chrC06_28519722 _ C	chrA03 _ 18783462 _ chrA03_18783462 _ C	NA _ NA _ NA _ NA
chrA02_random _ 37942 _ chrA02_random_37942 _ A	chrC09 _ 13983029 _ chrC09_13983029 _ C	chrCnn_random _ 76421043 _ chrCnn_random_76421043 _ T	chrCnn_random _ 76421043 _ chrCnn_random_76421043 _ T	chrA04 _ 17560247 _ chrA04_17560247 _ G	chrC05 _ 1352383 _ chrC05_1352383 _ A
chrC03_random _ 1162629 _ chrC03_random_1162629 _ A	chrC09 _ 13994729 _ chrC09_13994729 _ A	chrA02 _ 132006 _ chrA02_132006 _ A	chrC03_random _ 1162629 _ chrC03_random_1162629 _ A	chrA02 _ 199341 _ chrA02_199341 _ T	chrA10 _ 11084937 _ chrA10_11084937 _ G
NA _ NA _ NA _ NA	chrC02 _ 3206050 _ chrC02_3206050 _ G	chrA02 _ 607477 _ chrA02_607477 _ C	chrC05 _ 6821266 _ chrC05_6821266 _ A	chrC03 _ 17319265 _ chrC03_17319265 _ C	chrA10 _ 11157815 _ chrA10_11157815 _ A
chrC03 _ 16291033 _ chrC03_16291033 _ A	chrC05 _ 6821266 _ chrC05_6821266 _ A	chrA08 _ 14039932 _ chrA08_14039932 _ G	chrA02 _ 199341 _ chrA02_199341 _ T	chrC06 _ 28519722 _ chrC06_28519722 _ C	chrA10_random _ 1849689 _ chrA10_random_1849689 _ G

Figure 4.9: Detail of pattern groups

offers various tests association such as Cochran-Armitage trend test, Fisher's exact test, genotypic tests (general, dominant, and recessive models), and Cochran-Mantel-Haenszel tests to measure the association strength between SNPs and disease. PLINK allows to test individual SNPs or pair of SNPs that are associated with disease. It tests all SNPs and presents the results with Manhattan plots. This plot shows $-log10\ p\_value$ for each SNP against chromosomal location. For visual effect, chromosomes are shown in different colors. Based on this plot, user can observe which region of chromosomes contains significant SNPs.

Similarly, SNPsys software [126] is a graphical tool that allows to discover and visualize pairs of SNPs from large genetic variant datasets on complex diseases. This software can run on desktop machine or web browser as a stand-alone application. An example of SNP-SNP interactions generated by SNPsys is illustrated in Fig 4.10. With only textual output, it is difficult to observe SNPs interactions on the whole genome by SNPsys.

Beside these tools, some organizations provide public websites to search and view

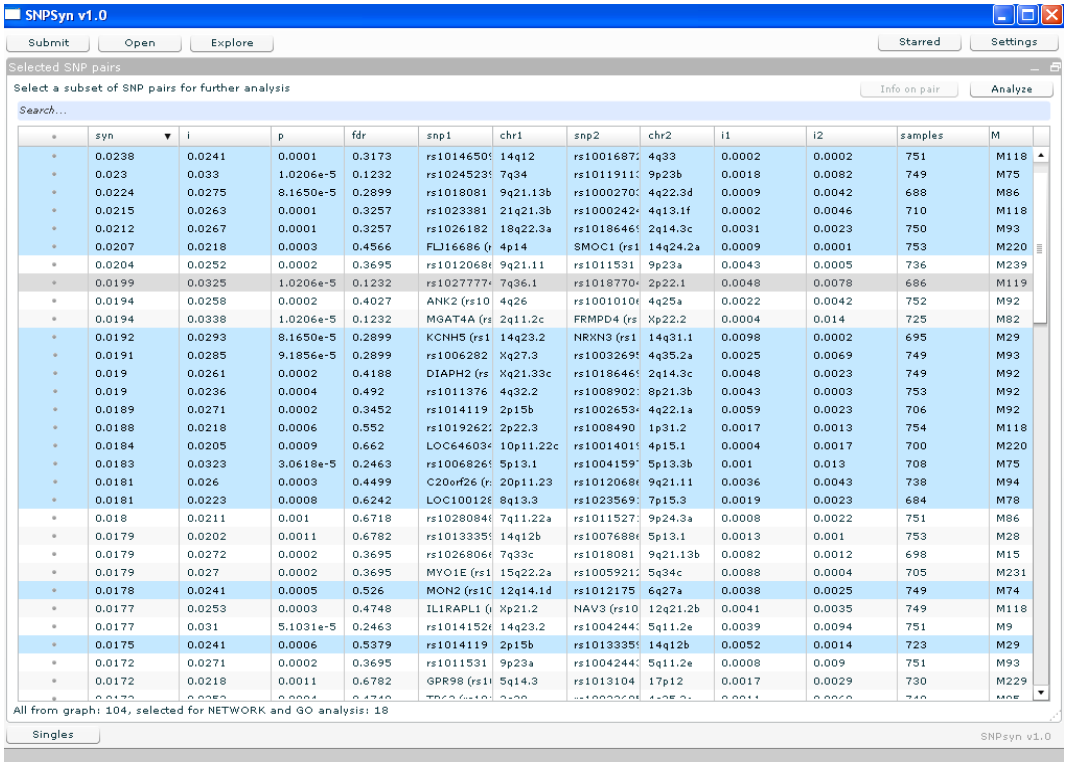


Figure 4.10: Pairs of SNPs interaction discovered by SNPsys

individual SNPs on real chromosomes. For example, National Center for Biotechnology Information (NCBI) supports 1000 genomes browse web page to search and visualize various information related to SNPs. Similarly, other websites such as https://www.snpedia.com/, https://www.ensembl.org/index.html also provide online tool to find SNPs in whole genome.

In short, these are useful graphic tools to search and visualize SNPs in the whole genome. However, these tools are used for individual SNPs or pair of SNP interaction. In comparison with available tools, our software has two different features: First, the interactive GUI provides multiple functions to cluster and visualize SNP combinations on whole chromosomes. The clustering algorithm is efficiently used to regroup similar patterns. This task is useful since the number of analysis pattern groups is much more smaller than the proportion of beginning patterns. In addition, the combinations of multiple SNPs (much more larger than 2) are taken into analysis. With GUI, users can easily observe these SNPs combinations with various additional biological information such as genes, QTLs. Second, our tool allows to

visualize different genetic variant datasets such as human, plant, animal instead of single dataset like PLINK and SNPsys.

4.7 Conclusion

In this chapter, various effective methods are presented to implement a graphic software for visualization of SNP combinations in the whole genome. The software applies discriminative pattern mining algorithms to discover high-order SNP combinations in large genetic datasets and visualize them with a GUI. The interactive GUI is an efficient and easy-to-use genetic-analysis tool that is required to support biologists in their search for relations between genotype and phenotype. Currently, the software is made of two separated modules and data preparation step for each SNP dataset. One perspective for further work is to integrate these steps in a single GUI. Another perspective is to improve the performance of the software to efficiently work with larger number of patterns. A last direction for future work is to investigate efficient method to represent the pattern groups.

Chapter 5

Conclusions and Perspectives

5.1 Contributions summary

Discovering SNPs association with diseases is an important task of bioinformatics. Once new genetic variant associations are identified, they can be used to develop better strategies to detect, treat and prevent the disease. However, discovering multiple SNP combinations is still a challenge since the number of SNP combinations is huge. The major problems of this issue include association strength evaluation, SNP combinations discovery, multiple hypothesis testing and interesting SNP combinations visualization.

To address these challenges this thesis has advanced the state-of-the-art of discriminative pattern mining techniques to discover SNP combinations associated with interesting phenotype. Different solutions have been proposed in this thesis to efficiently support all steps of GWAS analysis.

First, an efficient evaluation method has been proposed to assess the association strength between SNP combinations and diseases. The proposed method is based on the skypattern technique which allows multiple measures to be used to evaluate the importance of a pattern without giving specific thresholds. Experiment on various real SNP datasets demonstrate that this evaluation method is efficient for identifying genetic variant combinations associated with diseases.

Second, an efficient algorithm has been proposed to address the problems of computation and multiple hypothesis testing. The algorithm applies risk measures such as risk difference, risk ratio and odds ratio combined with confidence intervals to directly discover statistically significant discriminative pattern in two-class datasets. Experiment on various large genetic variant datasets demonstrate that the investigated algorithm efficiently discovers high-order SNP combinations in a short execution time. Many of them contain SNPs which are already known as associated

with diseases.

Third, to visualize the interesting SNP combinations, an interactive graphical tool has been implemented. This tool is used to regroup similar SNP combinations into sensible groups and present them on the whole genome. This is an efficient and easy-to-use genetic-analysis tool that supports biologists in their search for the relations between genetic variant combinations and interesting phenotype.

In addition, although this thesis focuses on GWAS, other bioinformatics tasks can also benefit from the proposed techniques.

5.2 Perspectives

In general, the techniques proposed in this thesis efficiently discover high-order SNP combinations associated with an interesting phenotype. However, to more efficiently tackle GWAS, several directions should be explored in future work.

Skypattern is a promising approach for association strength evaluation. However, applying this technique to identify interesting genetic variant combinations association with diseases remains challenging. To find skypatterns, a two-step approach is used: first discovering all SNP combinations which satisfy a given minimum support threshold. Then using Skycube software to find skypatterns over the generated patterns. This approach is time-consuming and is only suitable for small genetic variant datasets. Thus to efficiently use the skypattern technique to measure the association strength between SNP combinations and a disease, one perspective is to directly discover skypatterns in one stage.

The SSDPS algorithm uses a search strategy which allows a combination of measures to be used to prune the search space. However the pruning process is mainly done in the negative procedure. Whereas, the positive procedure has to compute all closed patterns in the positive class based on the LCM principles. This task is still time-consuming since it cannot early prune the search space. Thus another perspective is to investigate novel strategies to early prune the search space parts which will not create discriminative patterns.

Global discriminative pattern mining algorithms have been proposed to effectively discover non-redundant discriminative patterns [49, 80]. They are very promising for discovering discriminative patterns in various biological datasets. However, there exist very few studies that using global discriminative pattern mining techniques to handle bioinformatics problems. This would be a direction for future work for applying these data mining techniques to bioinformatics.

Discovering less patterns but which are highly statistically significant and discriminative is a very important task. The existing approaches have combined multiple hypothesis testing and pattern mining in one stage to discover statistically sig-

nificant patterns. These are potential approaches to deal with bioinformatics tasks. However, the available approaches are still time-consuming. To handle large biological datasets such as genetic variant, further research on this direction is needed.

QTL analysis aims to detect chromosome regions which are correlated with interesting traits. The existing approaches use statistical methods which can detect individual QTLs on a short region of chromosome. However, in a biological context, these regions may be correlated to affect the trait. Identifying these region interactions is thus very interesting. Using discriminative pattern mining techniques to tackle this issue is a promising future work since it can discover high-order SNP combinations across multiple chromosomes. These SNP combinations can be used to analyze the correlation of QTLs.

Last but not least, discovering genetic variant combinations and presenting them as an easily to understand way is necessary for biologists. However, the available algorithms often generate a large number of patterns which contain many redundant ones. Represent these patterns as a small set of non-redundant patterns is a challenge. Thus, to provide better data visualization of patterns, collaborate with researchers in data visualization and biologists is an appropriate approach. To reduce number of patterns to show, different approaches can be considered such as post processing with clustering algorithm, using global pattern set mining techniques to find immediately few patterns. In addition, investigate the optimization measure for the set of patterns is also an important work.

Glossary

AMD	Age-Related Macular Degeneration
ARR	Absolute Risk Reduction
AVX2	Advanced Vector Extensions 2
BC	Breast Cancer
BD	Bipolar Disorder
BFS	Breadth-First Search
CAD	Coronary Artery Disease
CD	Crohn's Disease
CI	Confidence Interval
CII	Co-Information Index
CMH	Cochran-Mantel-Haenszel
DFS	Depth-First Search
EMBL-EBI	European Bioinformatics Institute
FACS	Fast Automatic Conditional Search
FWER	Family Wise Error Rate
GUI	Graphical User Interface
GWAS	Genome-Wide Association Study
HT	Hypertension

LAMP	Limitless Arity Multiple-testing Procedure
LCI	Lower Confidence Interval
NCBI	National Center for Biotechnology Information
OR	Odds Ratio
QTL	Quantitative Trait Locus
RA	Rheumatoid Arthritis
RR	Risk Ratio
SIMD	Single instruction multiple data
SNP	Single Nucleotide Polymorphism
SSDPS	Statistically Significant Discriminative Pattern Search
T1D	Type 1 Diabetes
T2D	Type 2 Diabetes
TFs	Transcription Factors
UCI	Upper Confidence Interval
UKBS	UK Blood Services

Bibliography

[1] J. N. Hirschhorn and M. J. Daly, "Genome-wide association studies for common diseases and complex traits," Nat Rev Genet, vol. 6, pp. 95–108, Feb. 2005.

[2] B. WS and M. JH, "Chapter 11: Genome-wide association studies," PLOS, 2012.

[3] P. M. Visscher, M. A. Brown, M. I. McCarthy, and J. Yang, "Five years of gwas discovery," American Journal of Human Genetics, vol. 90, pp. 7–24, Jan. 2012.

[4] H. Cordell, "Detecting gene-gene interactions that underlie human diseases," Nature Reviews Genetics, vol. 10, no. 6, pp. 392–404, 2009.

[5] H. Schwender and K. Ickstadt, "Identification of snp interactions using logic regression," Biostatistics, vol. 9, no. 1, p. 187, 2008.

[6] B. L. Fridley, "Bayesian variable and model selection methods for genetic association studies," Genetic Epidemiology, vol. 33, no. 1, pp. 27–37, 2009.

[7] J. Listgarten, S. Damaraju, B. Poulin, L. Cook, J. Dufour, A. Driga, J. Mackey, D. Wishart, R. Greiner, and B. Zanke, "Predictive models for breast cancer susceptibility from multiple single nucleotide polymorphisms," Clinical Cancer Research, vol. 10, no. 8, pp. 2725–2737, 2004.

[8] A. Serretti and E. Smeraldi, "Neural network analysis in pharmacogenetics of mood disorders," BMC Medical Genetics, vol. 5, pp. 27–27, Dec. 2004.

[9] Q. Xie, L. D. Ratnasinghe, H. Hong, R. Perkins, Z.-Z. Tang, N. Hu, P. R. Taylor, and W. Tong, "Decision forest analysis of 61 single nucleotide polymorphisms in a case-control study of esophageal cancer; a novel method," BMC Bioinformatics, vol. 6, no. 2, pp. S4–, 2005.

[10] S. J. Winham, C. L. Colby, R. R. Freimuth, X. Wang, M. de Andrade, M. Huebner, and J. M. Biernacka, "Snp interaction detection with random forests in high-dimensional genetic data," BMC Bioinformatics, vol. 13, no. 1, p. 164, 2012.

[11] C. C. M. Chen, H. Schwender, J. Keith, R. Nunkesser, K. Mengersen, and P. Macrossan, "Methods for identifying snp interactions: A review on variations of logic regression, random forest and bayesian logistic regression," IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 8, pp. 1580–1591, Nov. 2011.

[12] R. Upstill-Goddard, D. Eccles, J. Fliege, and A. Collins, "Machine learning approaches for the discovery of genegene interactions in disease data," Briefings in Bioinformatics, 2012.

[13] X. Liu, J. Wu, F. Gu, J. Wang, and Z. He, "Discriminative pattern mining and its applications in bioinformatics," Briefings in Bioinformatics, Nov. 2014.

[14] S. Naulaerts, P. Meysman, W. Bittremieux, T. N. Vu, W. Vanden Berghe, B. Goethals, and K. Laukens, "A primer to frequent itemset mining for bioinformatics," Briefings in Bioinformatics, vol. 16, pp. 216–231, Sept. 2013.

[15] S. Helal, "Subgroup discovery algorithms: A survey and empirical evaluation," Journal of Computer Science and Technology, vol. 31, no. 3, pp. 561–576, 2016.

[16] F. Gang, H. M, W. W, Y. H, S. M, and C. TR, "High-order snp combinations associated with complex diseases: Efficient discovery, statistical power and functional interactions.," PLoS ONE, 2012.

[17] L. T. H. Yu, F.-l. Chung, S. C. F. Chan, and S. M. C. Yuen, "Using emerging pattern based projected clustering and gene expression data for cancer detection," in Proceedings of the Second Conference on Asia-Pacific Bioinformatics - Volume 29, APBC '04, (Darlinghurst, Australia, Australia), pp. 75–84, Australian Computer Society, Inc., 2004.

[18] X. Liu, J. Wu, H. Gong, S. Deng, and Z. He, "Mining conditional phosphorylation motifs," IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 11, pp. 915–927, Sept. 2014.

[19] F. Herrera, C. J. Carmona, P. González, and M. J. del Jesus, "An overview on subgroup discovery: foundations and applications," Knowledge and Information Systems, vol. 29, pp. 495–525, Dec 2011.

[20] N. Yosef, Z. Yakhini, A. Tsalenko, V. Kristensen, A.-L. Borresen-Dale, E. Ruppin, and R. Sharan, "A supervised approach for identifying discriminating genotype patterns and its application to breast cancer data," Bioinformatics, vol. 23, pp. e91–e98, Jan. 2007.

[21] L.-Y. Chuang, H.-W. Chang, M.-C. Lin, and C.-H. Yang, "Improved branch and bound algorithm for detecting snp-snp interactions in breast cancer," Journal of Clinical Bioinformatics, vol. 3, no. 1, pp. 1–10, 2013.

[22] Z. Q, L. Q, and O. J, "Apriorigwas, a new pattern mining strategy for detecting genetic variants associated with disease through interaction effects," PLoS Comput Biol 10(6), 2014.

[23] X. Ding, J. Wang, A. Zelikovsky, X. Guo, M. Xie, and Y. Pan, "Searching high-order snp combinations for complex diseases based on energy distribution difference," IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 12, pp. 695–704, May 2015.

[24] G. Fang, G. Pandey, W. Wang, M. Gupta, M. Steinbach, and V. Kumar, "Mining low-support discriminative patterns from dense and high-dimensional data," IEEE Transactions on Knowledge and Data Engineering, vol. 24, pp. 279–294, Feb 2012.

[25] J. Li and L. Wong, "Emerging patterns and gene expression data," Genome Informatics, vol. 12, pp. 3–13, 2001.

[26] J. Li and L. Wong, "Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns," Bioinformatics, vol. 18, no. 5, pp. 725–734, 2002.

[27] A.-L. Boulesteix, G. Tutz, and K. Strimmer, "A cart-based approach to discover emerging patterns in microarray data," Bioinformatics, vol. 19, no. 18, pp. 2465–2472, 2003.

[28] A. Ritz, G. Shakhnarovich, A. R. Salomon, and B. J. Raphael, "Discovery of phosphorylation motif mixtures in phosphoproteomics data," Bioinformatics, vol. 25, no. 1, p. 14, 2009.

[29] Z. He, C. Yang, G. Guo, N. Li, and W. Yu, "Motif-all: discovering all phosphorylation motifs," in BMC Bioinformatics, vol. 12, pp. 1–8, BioMed Central, 2011.

[30] T. Wang, A. N. Kettenbach, S. A. Gerber, and C. Bailey-Kellogg, "Mmfph: a maximal motif finder for phosphoproteomics datasets," Bioinformatics, vol. 28, no. 12, p. 1562, 2012.

[31] A. Terada, K. Tsuda, and J. Sese, "Fast westfall-young permutation procedure for combinatorial regulation discovery," in 2013 IEEE International Conference on Bioinformatics and Biomedicine, pp. 153–158, Dec 2013.

[32] A. Terada, M. Okada-Hatakeyama, K. Tsuda, and J. Sese, "Statistical significance of combinatorial regulations," Proceedings of the National Academy of Sciences, vol. 110, no. 32, pp. 12996–13001, 2013.

[33] F. Llinares-López, M. Sugiyama, L. Papaxanthos, and K. Borgwardt, "Fast and memory-efficient significant pattern mining via permutation testing," in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, (New York, NY, USA), pp. 725–734, ACM, 2015.

[34] P. Kralj, N. Lavra, D. Gamberger, and A. Krstai, "Contrast set mining through subgroup discovery applied to brain ischaemina data," in Advances in Knowledge Discovery and Data Mining. Heidelberg: Springer Berlin, pp. 579 – 586, 2007.

[35] M. Mueller, R. Rosales, H. Steck, S. Krishnan, B. Rao, and S. Kramer, "Subgroup discovery for test selection: A novel approach and its application to breast cancer diagnosis," in Advances in Intelligent Data Analysis VIII: 8th International Symposium on Intelligent Data Analysis, IDA 2009, Lyon, France, August 31 - September 2, 2009. Proceedings, pp. 119–130, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[36] R. Sherhod, P. N. Judson, T. Hanser, J. D. Vessey, S. J. Webb, and V. J. Gillet, "Emerging pattern mining to aid toxicological knowledge discovery," J. Chem. Inf. Model., vol. 54, pp. 1864–1879, July 2014.

[37] M. Fabrgue, A. Braud, S. Bringay, C. Grac, F. Le Ber, D. Levet, and M. Teisseire, "Discriminant temporal patterns for linking physico-chemistry and biology in hydro-ecosystem assessment," Ecological Informatics, vol. 24, no. Supplement C, pp. 210–221, 2014.

[38] M. Garca-Borroto, J. Martnez-Trinidad, and J. Carrasco-Ochoa, "A survey of emerging patterns for supervised classification," Springer Netherlands, vol. 42, no. 4, pp. 705–721, 2014.

[39] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," SIGMOD Rec., vol. 22, pp. 207–216, June 1993.

[40] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," SIGMOD Rec., vol. 29, pp. 1–12, May 2000.

[41] M. J. Zaki and C.-J. Hsiao, "Charm: An efficient algorithm for closed itemset mining," in Proceedings of the 2002 SIAM International Conference on Data Mining, pp. 457–473, 2002.

[42] C. Borgelt, "Frequent item set mining," WIREs Data Mining Knowl Discov 2012, vol. 2, pp. 437–456, 2012.

[43] M. J. del Jesus, P. Gonzalez, F. Herrera, and M. Mesonero, "Evolutionary fuzzy rule induction process for subgroup discovery: A case study in marketing," IEEE Transactions on Fuzzy Systems, vol. 15, pp. 578–592, Aug 2007.

[44] H. Cheng, X. Yan, J. Han, and P. S. Yu, "Direct discriminative pattern mining for effective classification," ICDE '08, (Washington, DC, USA), pp. 169–178, IEEE Computer Society, 2008.

[45] G. Dong and J. Li, "Efficient mining of emerging patterns: Discovering trends and differences," in Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '99, (New York, NY, USA), pp. 43–52, ACM, 1999.

[46] X. Zhang, J. Li, and G. Dong, "Discovering jumping emerging patterns and experiments on real datasets," in Proceedings of 9th International Database Conference on Heterogeneous and Internet Databases (IDC99), Hong Kong, July 15-17, 1999., 1999.

[47] S. Bay and M. Pazzani, "Detecting group differences: Mining contrast sets," Kluwer Academic Publishers, vol. 5, no. 3, pp. 213–246–, 2001.

[48] T. Abudawood and P. Flach, "Evaluation measures for multi-class subgroup discovery," in Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2009, Bled, Slovenia, September 7-11, 2009, Proceedings, Part I, pp. 35–50, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[49] T. Guns, S. Nijssen, and L. D. Raedt, "k-pattern set mining under constraints," IEEE Transactions on Knowledge and Data Engineering, vol. 25, pp. 402–418, Feb 2013.

[50] G. A. Barnard, "Introduction to pearson (1900) on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," in Breakthroughs in Statistics: Methodology and Distribution, pp. 1–10, New York, NY: Springer New York, 1992.

[51] R. A. Fisher, "On the interpretation of x2 from contingency tables, and the calculation of p," Journal of the Royal Statistical Society, no. 85(1), pp. 87 – 94, 1922.

[52] C. E. Bonferroni, "Teoria statistica delle classi e calcolo delle probabilit'a," Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze, vol. 8, pp. 3 – 62, 1936.

[53] R. E. Tarone, "A modified bonferroni method for discrete data," Biometrics, no. 46(2), pp. 515 – 522, 1990.

[54] P. H. Westfall and S. S. Young, "Resampling-based multiple testing: Examples and methods for p-value adjustment," New York: Wiley, 1993.

[55] G. Fang, W. Wang, B. Oatley, B. Van Ness, M. Steinbach, and V. Kumar, "Characterizing discriminative patterns," ArXiv e-prints, Feb. 2011.

[56] L. Geng and H. J. Hamilton, "Interestingness measures for data mining: A survey," ACM Comput. Surv., vol. 38, Sept. 2006.

[57] N. Lavrač, B. Kavšek, P. Flach, and L. Todorovski, "Subgroup discovery with cn2-sd," J. Mach. Learn. Res., vol. 5, pp. 153–188, Dec. 2004.

[58] D. Gamberger and N. Lavrac, "Expert-guided subgroup discovery: Methodology and application," J. Artif. Int. Res., vol. 17, pp. 501–527, Dec. 2002.

[59] J. Li and Q. Yang, "Strong compound-risk factors: Efficient discovery through emerging patterns and contrast sets," IEEE Transactions on Information Technology in Biomedicine, vol. 11, no. 5, pp. 544–552, 2007.

[60] T. Guns, S. Nijssen, and L. De Raedt, "Itemset mining: A constraint programming perspective," Artificial Intelligence, vol. 175, no. 12, pp. 1951–1983, 2011.

[61] S. Morishita and J. Sese, "Transversing itemset lattices with statistical metric pruning," in Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS '00, (New York, NY, USA), pp. 226–236, ACM, 2000.

[62] M. Atzmueller and F. Puppe, "Sd-map a fast algorithm for exhaustive subgroup discovery," PKDD 2006, pp. 6–17, 2006.

[63] B. Kavek and N. Lavra, "Apriori-sd: Adapting association rule learning to subgroup discovery," Applied Artificial Intelligence, vol. 20, no. 7, pp. 543–583, 2006.

[64] T. D. Cook, "Advanced statistics: Up with odds ratios! a case for odds ratios when outcomes are common," Academic Emergency Medicine, vol. 9, no. 12, pp. 1430–1434, 2002.

[65] H. Cheng, X. Yan, J. Han, and C. W. Hsu, "Discriminative frequent pattern analysis for effective classification," in 2007 IEEE 23rd International Conference on Data Engineering, pp. 716–725, April 2007.

[66] Y. Morimoto, T. Fukuda, H. Matsuzawa, T. Tokuyama, and K. Yoda, "Algorithms for mining association rules for binary segmentations of huge categorical databases," in: Proceedings of 24rd International Conference on Very Large Data Bases, Morgan Kaufmann, pp. 380 – 391, 1998.

[67] Q. Liu and G. Dong, "A contrast pattern based clustering quality index for categorical data," in 2009 Ninth IEEE International Conference on Data Mining, pp. 860–865, Dec 2009.

[68] F. Geerts, B. Goethals, and T. Mielikinen, "Tiling databases," in Discovery Science: 7th International Conference, DS 2004, Padova, Italy, October 2-5, 2004. Proceedings, pp. 278–289, Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[69] T. Guns, S. Nijssen, and L. De Raedt, "Evaluating pattern set mining strategies in a constraint programming framework," in Advances in Knowledge Discovery and Data Mining: 15th Pacific-Asia Conference, PAKDD 2011, Shenzhen, China, May 24-27, 2011, Proceedings, Part II (J. Z. Huang, L. Cao, and J. Srivastava, eds.), pp. 382–394, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[70] M. Sugiyama, F. L. Lpez, N. Kasenburg, and K. M. Borgwardt, "Significant subgraph mining with multiple testing correction," in Proceedings of the 2015 SIAM International Conference on Data Mining, pp. 37–45, 2015.

[71] G. I. Webb, "Discovering significant patterns," Machine Learning, vol. 68, no. 1, pp. 1–33, 2007.

[72] L. Papaxanthos, F. Llinares-Lopez, D. Bodenham, and K. Borgwardt, "Finding significant combinations of features in the presence of categorical covariates," in Advances in Neural Information Processing Systems 29 (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), pp. 2279–2287, Curran Associates, Inc., 2016.

[73] R. E. Tarone, "A modified bonferroni method for discrete data," in Biometrics, vol. 46, pp. 515–522, [Wiley, International Biometric Society], 1990.

[74] S. Minato, T. Uno, K. Tsuda, A. Terada, and J. Sese, "Fast statistical assessment for combinatorial hypotheses based on frequent itemset mining," Hokkaido University, 2014.

[75] S. Dudoit, J. P. Shaffer, and J. C. Boldrick, "Multiple hypothesis testing in microarray experiments," in Statistical Science, vol. 18, pp. 71–103, Institute of Mathematical Statistics, 2003.

[76] N. Meinshausen, M. H. Maathuis, and P. Bhlmann, "Asymptotic optimality of the westfall–young permutation procedure for multiple testing under dependence," The Annals of Statistics, vol. 39, no. 6, pp. 3369 – 3391, 2011.

[77] S. Nijssen, T. Guns, and L. De Raedt, "Correlated itemset mining in roc space: A constraint programming approach," in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, (New York, NY, USA), pp. 647–656, ACM, 2009.

[78] M. van Leeuwen and A. Knobbe, "Diverse subgroup set discovery," Data Mining and Knowledge Discovery, vol. 25, no. 2, pp. 208–242, 2012.

[79] T. Guns, S. Nijssen, A. Zimmermann, and L. D. Raedt, "Declarative heuristic search for pattern set mining," in 2011 IEEE 11th International Conference on Data Mining Workshops, pp. 1104–1111, Dec 2011.

[80] Z. He, F. Gu, C. Zhao, X. Liu, J. Wu, and J. Wang, "Conditional discriminative pattern mining: Concepts and algorithms," Information Sciences, vol. 375, pp. 1 – 15, 2017.

[81] T. Lucas, T. C. Silva, R. Vimieiro, and T. B. Ludermir, "A new evolutionary algorithm for mining top-k discriminative patterns in high dimensional data," Applied Soft Computing, vol. 59, pp. 487 – 499, 2017.

[82] T. Pontes, R. Vimieiro, and T. B. Ludermir, "Ssdp: A simple evolutionary approach for top-k discriminative patterns in high dimensional databases," in

2016 5th Brazilian Conference on Intelligent Systems (BRACIS), pp. 361–366, Oct 2016.

[83] P. Terlecki and K. Walczak, "Efficient discovery of top-k minimal jumping emerging patterns," in Rough Sets and Current Trends in Computing: 6th International Conference, RSCTC 2008 Akron, OH, USA, October 23-25, 2008 Proceedings (C.-C. Chan, J. W. Grzymala-Busse, and W. P. Ziarko, eds.), pp. 438–447, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[84] C. J. Carmona, P. Gonzalez, M. J. d. Jesus, and F. Herrera, "Nmeef-sd: Non-dominated multiobjective evolutionary algorithm for extracting fuzzy rules in subgroup discovery," IEEE Transactions on Fuzzy Systems, vol. 18, pp. 958–970, Oct 2010.

[85] V. Pachón, J. Mata, J. L. Domínguez, and M. J. Maña, "Multi-objective evolutionary approach for subgroup discovery," in Hybrid Artificial Intelligent Systems: 6th International Conference, HAIS 2011, Wroclaw, Poland, May 23-25, 2011, Proceedings, Part II (E. Corchado, M. Kurzyński, and M. Woźniak, eds.), pp. 271–278, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[86] T. Uno, M. Kiyomi, and H. Arimura, "Lcm ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets," in Workshop Frequent Item Set Mining Implementations, 2004.

[87] L. F, R. M, and A. M, "Fast discovery of relevant subgroup patterns," in In Florida Artificial Intelligence Research Society Conference., 2010.

[88] W. Fan, K. Zhang, H. Cheng, J. Gao, X. Yan, J. Han, P. Yu, and O. Verscheure, "Direct mining of discriminative and essential frequent patterns via model-based search tree," in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, (New York, NY, USA), pp. 230–238, ACM, 2008.

[89] W. Li, J. Han, and J. Pei, "Cmar: accurate and efficient classification based on multiple class-association rules," in Proceedings 2001 IEEE International Conference on Data Mining, pp. 369–376, 2001.

[90] X. Yin and J. Han, "Cpar: Classification based on predictive association rules," in Proceedings, pp. 331–335–, Society for Industrial and Applied Mathematics, May 2003.

[91] C. J. Carmona, P. González, M. J. del Jesus, and F. Herrera, "Overview on evolutionary subgroup discovery: Analysis of the suitability and potential of

the search performed by evolutionary algorithms," Wiley Int. Rev. Data Min. and Knowl. Disc., vol. 4, pp. 87–103, Mar. 2014.

[92] H. Fan and K. Ramamohanarao, "An efficient single-scan algorithm for mining essential jumping emerging patterns for classification," PAKDD 2002, Springer, Heidelberg, p. 456462, 2002.

[93] M. van Leeuwen and A. Knobbe, "Non-redundant subgroup discovery in large and complex data," in Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part III, pp. 459–474, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[94] H. GroBkreutz, D. Paurat, and S. Rüping, "An enhanced relevance criterion for more concise supervised pattern discovery," in Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, (New York, NY, USA), pp. 1442–1450, ACM, 2012.

[95] Y. Kameya and T. Sato, "Rp-growth: Top-k mining of relevant patterns with minimum support raising," in Proceedings, pp. 816–827–, Society for Industrial and Applied Mathematics, Apr. 2012.

[96] G. C. Garriga, P. Kralj, and N. Lavrač, "Closed sets for labeled data," Journal of Machine Learning Research, no. 9, pp. 559 – 580, 2008.

[97] M. Boley and H. Grosskreutz, "Non-redundant subgroup discovery using a closure system," in Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2009, Bled, Slovenia, September 7-11, 2009, Proceedings, Part I (W. Buntine, M. Grobelnik, D. Mladenić, and J. Shawe-Taylor, eds.), pp. 179–194, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[98] B. Negrevergne, A. Termier, M.-C. Rousset, and J.-F. Mhaut, "Paraminer: a generic pattern mining algorithm for multi-core architectures," Data Mining and Knowledge Discovery, vol. 28, no. 3, pp. 593–633, 2014.

[99] J. Li, G. Liu, and L. Wong, "Mining statistically important equivalence classes and delta-discriminative emerging patterns," in Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '07, (New York, NY, USA), pp. 430–439, ACM, 2007.

[100] N. Tatti, "Maximum entropy based significance of itemsets," Knowledge and Information Systems, vol. 17, pp. 57–77, Oct 2008.

[101] M. Mampaey, N. Tatti, and J. Vreeken, "Tell me what i need to know: Succinctly summarizing data with itemsets," in Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11, (New York, NY, USA), pp. 573–581, ACM, 2011.

[102] N. Mantel and W. Haenszel, "Statistical aspects of the analysis of data from retrospective studies of disease," Journal of the National Cancer Institute, 1959.

[103] L. Ma, T. L. Assimes, N. B. Asadi, C. Iribarren, T. Quertermous, and W. H. Wong, "An almost exhaustive search-based sequential permutation method for detecting epistasis in disease association studies," Genetic Epidemiology, vol. 34, no. 5, pp. 434–443, 2010.

[104] M. D. Ritchie, L. W. Hahn, N. Roodi, L. R. Bailey, W. D. Dupont, F. F. Parl, and J. H. Moore, "Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer," The American Journal of Human Genetics, vol. 69, pp. 138–147, July 2001.

[105] C.-H. Yang, Y.-D. Lin, C.-S. Yang, and L.-Y. Chuang, "An efficiency analysis of high-order combinations of gene-gene interactions using multifactor-dimensionality reduction," BMC Genomics, vol. 16, no. 1, p. 489, 2015.

[106] S. Leem, H. hwan Jeong, J. Lee, K. Wee, and K.-A. Sohn, "Fast detection of high-order epistatic interactions in genome-wide association studies using information theoretic measure," Computational Biology and Chemistry, vol. 50, pp. 19 – 28, 2014. Advances in Bioinformatics: Twelfth Asia Pacific Bioinformatics Conference (APBC2014).

[107] J. Shang, J. Zhang, Y. Sun, and Y. Zhang, "Epiminer: A three-stage co-information based method for detecting and visualizing epistatic interactions," Digital Signal Processing, vol. 24, pp. 1–13, Jan. 2014.

[108] M. Xie, J. Li, and T. Jiang, "Detecting genome-wide epistases based on the clustering of relatively frequent items," Bioinformatics, vol. 28, no. 1, p. 5, 2011.

[109] P. Salle, S. Bringay, M. Teisseire, F. Chakkour, M. Roche, R. Abdel Rassoul, J.-M. Verdier, and G. Devau, "GeneMining: Identification, Visualization, and Interpretation of Brain Ageing Signatures," in Medical Informatics in a United and Healthy Europe, p. 5, Aug. 2009.

[110] A. Sallaberry, N. Pecheur, S. Bringay, M. Roche, and M. Teisseire, "Sequential patterns mining and gene sequence visualization to discover novelty from microarray data," Journal of Biomedical Informatics, vol. 44, no. 5, pp. 760–774, 2011.

[111] Y. Lai, B. Wu, L. Chen, and H. Zhao, "A statistical method for identifying differential gene-gene co-expression patterns," Bioinformatics, vol. 20, pp. 3146–3155, Nov. 2004.

[112] D. Schwartz and S. P. Gygi, "An iterative statistical approach to the identification of protein phosphorylation motifs from large-scale data sets," Nat Biotech, vol. 23, pp. 1391–1398, Nov. 2005.

[113] Y.-C. Chen, K. Aguan, C.-W. Yang, Y.-T. Wang, N. R. Pal, and I.-F. Chung, "Discovery of protein phosphorylation motifs through exploratory data analysis," PLOS ONE, vol. 6, pp. e20025–, May 2011.

[114] A. Soulet, C. Raissi, M. Plantevit, and B. Cremilleux, "Mining dominant patterns in the sky," in ICDM, pp. 655–664, 2011.

[115] W. Ugarte, P. Boizumault, S. Loudni, and B. Cremilleux, "Computing skypattern cubes using relaxation," in ICTAI, pp. 859–866, Nov 2014.

[116] T. W. T. C. C. Consortium, "Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls," Nature, vol. 447, pp. 661–678, June 2007.

[117] G. M. Clarke, C. A. Anderson, F. H. Pettersson, L. R. Cardon, A. P. Morris, and K. T. Zondervan, "Basic statistical analysis in genetic case-control studies," Nature protocols, vol. 6, pp. 121–133, Feb. 2011.

[118] J. A. Morris and M. J. Gardner, "Statistics in medicine: Calculating confidence intervals for relative risks (odds ratios) and standardised ratios and rates," British Medical Journal (Clinical research ed.), vol. 296, pp. 1313–1316, May 1988.

[119] J.-B. du Prel, G. Hommel, B. Rhrig, and M. Blettner, "Confidence interval or p-value?: Part 4 of a series on evaluation of scientific publications," Deutsches rzteblatt International, vol. 106, pp. 335–339, Aug. 2008.

[120] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, "Discovering frequent closed itemsets for association rules," in Proceedings of the 7th International Conference on Database Theory, ICDT '99, (London, UK, UK), pp. 398–416, Springer-Verlag, 1999.

[121] V. Leroy, M. Kirchgessner, A. Termier, and S. Amer-Yahia, "Toppi: An efficient algorithm for item-centric mining," Inf. Syst., vol. 64, pp. 104–118, 2017.

[122] F. Pan, G. Cong, A. K. H. Tung, J. Yang, and M. J. Zaki, "Carpenter: Finding closed patterns in long biological datasets," in Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03, (New York, NY, USA), pp. 637–642, ACM, 2003.

[123] R. J. Klein, C. Zeiss, E. Y. Chew, J.-Y. Tsai, R. S. Sackler, C. Haynes, A. K. Henning, J. P. SanGiovanni, S. M. Mane, S. T. Mayne, M. B. Bracken, F. L. Ferris, J. Ott, C. Barnstable, and J. Hoh, "Complement factor h polymorphism in age-related macular degeneration," Science, vol. 308, no. 5720, pp. 385–389, 2005.

[124] WTCCC and TASC, "Association scan of 14,500 nssnps in four common diseases identifies variants involved in autoimmunity," Nature genetics, vol. 39, pp. 1329–1337, Oct. 2007.

[125] S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. Ferreira, D. Bender, J. Maller, P. Sklar, P. deBakker, M. Daly, and P. Sham, "Plink: A tool set for whole-genome association and population-based linkage analyses," American Journal of Human Genetics, vol. 81, pp. 559–575, May 2007.

[126] T. Curk, G. Rot, and B. Zupan, "Snpsyn: detection and exploration of snp-snp interactions," Nucleic Acids Research, vol. 39, pp. W444–W449, July 2011.

[127] A. K. Jain, "Data clustering: 50 years beyond k-means," Pattern Recognition Letters, vol. 31, no. 8, pp. 651 – 666, 2010. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR)19th International Conference in Pattern Recognition (ICPR).

[128] S. C. Johnson, "Hierarchical clustering schemes," Psychometrika, 1967.

[129] B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," in Proceedings 1996 IEEE Symposium on Visual Languages, pp. 336–343, Sep 1996.

List of Tables

List of Figures

List of Algorithms

Abstract

Genome-wide association studies (GWAS) is designed to discover single nucleotide polymorphism (SNP) combinations associated with diseases. Once new genetic associations are identified, they can be used to develop better strategies to detect, treat and prevent the diseases. Recently, GWAS has been tackled with discriminative pattern mining algorithms. However, discovering of SNP combinations in large genetic variant datasets remains challenging. To address these challenges this thesis advances the state-of-the-art of discriminative pattern mining techniques to discover SNP combinations associated with interesting phenotype. Different solutions have been proposed in this thesis. They focus on major problems of GWAS such as association strength evaluation, SNP combinations discovery and interesting SNP combinations visualization. The proposed solutions in this thesis are also promising for other tasks of bioinformatics such as differential gene expression discovery, phosphorylation motifs detection and regulatory motif combination mining.

Keywords: Genome-wide association studies, single nucleotide polymorphism, discriminative pattern mining, association strength measure, visualization.

Résumé

Les études d'association sur un génome complet (GWAS) sont conçues pour découvrir les combinaisons de points de polymorphisme (SNP) associes à des maladies. La découverte de ces associations permet d'élaborer de meilleures stratégies pour détecter, traiter ou prévenir les maladies. Récemment, l'utilisation de techniques d'extraction de patterns discriminatif a été investiguée dans le cadre de problématiques GWAS. Toutefois, la découverte de combinaisons de SNP dans de grands jeux de données GWAS est encore difficile à cause de la complexité des algorithmes utilisés. La thèse se propose donc d'améliorer l'état de l'art des approches d'extraction de motifs discriminants, dans le cadre d'extraction de combinaisons de SNP corrélées à un phénotype d'intérêt. Plusieurs solutions ont été proposées, s'attaquant aux problèmes majeurs en GWAS : évaluation de la force d'association, découverte efficace de combinaisons de SNP et visualisation de ces combinaisons. Les approches proposées sont également prometteuses pour d'autres tâches de bioinformatique comme la découverte d'expressions génique, la détection de motifs de phosphorylation et la détection de motifs de régulation.

Mot clé: etudes d'association sur genome complet, points de polymorphisme, extraction de motifs discriminants, mesure de force d'association, visualisation