Journal of Biomedical Informatics 56 (2015) 273–283

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

DMET-Miner: Efficient discovery of association rules from pharmacogenomic data

Giuseppe Agapito [a], Pietro H. Guzzi [a], Mario Cannataro [a,b,*]

[a] Dep. of Medical and Surgical Sciences, University Magna Graecia of Catanzaro, Italy
[b] ICAR-CNR, Italy

ARTICLE INFO

Article history:
Received 29 December 2014
Revised 9 May 2015
Accepted 3 June 2015
Available online 16 June 2015

Keywords:
Personalized medicine
Single nucleotide polymorphism
Frequent itemset mining
Association rules

ABSTRACT

Microarray platforms enable the investigation of allelic variants that may be correlated to phenotypes. Among those, the Affymetrix DMET (Drug Metabolism Enzymes and Transporters) platform enables the simultaneous investigation of all the genes that are related to drug absorption, distribution, metabolism and excretion (ADME). Although recent studies demonstrated the effectiveness of the use of DMET data for studying drug response or toxicity in clinical studies, there is a lack of tools for the automatic analysis of DMET data. In a previous work we developed DMET-Analyzer, a methodology and a supporting platform able to automatize the statistical study of allelic variants, that has been validated in several clinical studies. Although DMET-Analyzer is able to correlate a single variant for each probe (related to a portion of a gene) through the use of the Fisher test, it is unable to discover multiple associations among allelic variants, due to its underlying statistic analysis strategy that focuses on a single variant for each time. To overcome those limitations, here we propose a new analysis methodology for DMET data based on Association Rules mining, and an efficient implementation of this methodology, named DMET-Miner. DMET-Miner extends the DMET-Analyzer tool with data mining capabilities and correlates the presence of a set of allelic variants with the conditions of patient's samples by exploiting association rules. To face the high number of frequent itemsets generated when considering large clinical studies based on DMET data, DMET-Miner uses an efficient data structure and implements an optimized search strategy that reduces the search space and the execution time. Preliminary experiments on synthetic DMET datasets, show how DMET-Miner outperforms off-the-shelf data mining suites such as the FP-Growth algorithms available in Weka and RapidMiner. To demonstrate the biological relevance of the extracted association rules and the effectiveness of the proposed approach from a medical point of view, some preliminary studies on a real clinical dataset are currently under medical investigation.

1. Introduction

Personalized medicine refers to the possibility to tailor therapies on the basis of the genome of patients, under the assumption that different genomic variants may impact on the response to drugs Wang and Liotta [19], Stranger et al. [17], Lombardi et al. [11], and Rumiato et al. [14]. A main driver for the development of personalized medicine is the availability of novel high-throughput platforms such as microarray, at a relative low cost, Wang and Liotta [19] and Stranger et al. [17], enabling the large scale screening of genomes of patients for possible known or unknown genetic variants (i.e. Single Nucleotide Polimorphisms or allelic variants) and then the further selection of drugs on the basis of the patient's genotype, in order to maximize their efficacy or to reduce their toxicity.

A set of SNPs and allelic variants, known to be related to Adverse Drug Reactions (ADR), have been determined in the past Li et al. [10]. ADRs occur most frequently when a drug has a narrow therapeutic index. The therapeutic index is a measure of the amount of drug that may cause lethal effect. When a drug has a narrow therapeutic index it means that there exists little difference between the lethal and the therapeutic dose. Consequently the investigation of these polymorphism may avoid the incorrect dosage of drugs and then the insurgence of reactions since their presence/absence favorites ADRs.

The DMET (Drug Metabolism Enzymes and Transporters) platform developed by Affymetrix is used to detect in human samples the allelic variants on 225 genes that are related with drug

* Corresponding author at: Dep. of Medical and Surgical Sciences, University Magna Graecia of Catanzaro, Italy.
E-mail addresses: agapito@unicz.it (G. Agapito), hguzzi@unicz.it (P.H. Guzzi), cannataro@unicz.it (M. Cannataro).

http://dx.doi.org/10.1016/j.jbi.2015.06.005
1532-0464/

absorption, distribution, metabolism and excretion (ADME) Li et al. [10]; variations are detected considering 1936 probes and with respect to a reference population.[1] Many recent works demonstrated the role of genetic variations in ADME genes in association with the heterogeneity in drug treatment effects Lombardi et al. [11], Rumiato et al. [14], Di Martino et al. [5,6], Shiotani et al. [15], Hu et al. [9], Mizzi et al. [12], and Wakil et al. [18].

In a classical case-control study, the DMET platform generates a $np \times ns$ matrix of alleles, where np is the number of probes ($np = 1936$ for current DMET chips) and ns is the number of samples (patients). Each value of such a matrix is represented by a string that may theoretically take all the possible symbols "s_1/s_2", where $s_1, s_2 \in S = \{A, C, G, T, -\}$, or the "*NoCall*" symbol. S contains all the nucleotides symbols and the special symbol "–", that is used to denote insertion/deletion of nucleotides, while "*NoCall*" indicates that the DMET platform has not been able to detect the nucleotide Wang and Liotta [19] and Li et al. [10].

The development of novel algorithms able to manage and mine SNP data in case-control studies and to provide information for clinical decision is currently a novel research area. In fact, DMET data must be first preprocessed and then analyzed in order to find correlation between the genotype and the condition of samples (e.g. type of drug treatment or response to a drug). To the best of our knowledge, the main available tools to process DMET data are those provided by Affymetrix, and DMET-Analyzer Guzzi et al. [7].

The `apt-dmet-genotype` software of the Affymetrix Power Tools suite, or the DMET Console platform Sissung et al. [16], generally allow only the sequential preprocessing of binary data and simple data analysis operations, but do not allow to test the association of the presence of SNPs with the response to drugs.

On the other hand, DMET-Analyzer Guzzi et al. [7] is a recent software platform for the automatic statistical analysis of DMET data that employs the well-known Fisher test and several statistical corrections such as Bonferroni or False Discovery Rate. DMET-Analyzer also supports the visualization of the polymorphisms detected on the entire dataset as a heat-map to give an immediate visual feedback to the user. Finally, it annotates significant SNPs with information provided by Affymetrix libraries and with links to the dbSNP database (for short genetic variations - http://www.ncbi.nlm.nih.gov/SNP/) and to the PharmaGKB (Pharmacogenomics Knowledge Base - http://www.pharmagkb.org), Hernandez-Boussard et al. [8], giving various information (e.g. pathways) related to pharmacogenomics.

Although DMET-Analyzer has demonstrated its validity in several clinical studies Lombardi et al. [11], Rumiato et al. [14], and Di Martino et al. [5,6], it presents a main limitation: the association among the presence of SNPs and the classes of samples (patients), is determined through the use of the well known Fisher's test, thus it is able to discover only the association among a single allelic variants and the clinical conditions.

Nevertheless, many diseases such as cancer are known to be multi-factorial, i.e. related to variants in more than a single gene. Unfortunately, the mining strategy of DMET-Analyzer is not able to cope with multiple variants. Because it determines only single variants, it is not able to group all of them in a single, easy to understand, and biologically relevant, information.

To overcome those limitations, we developed DMET-Miner, a novel methodology for the concurrent analysis of genomic variants in more than a gene. DMET-Miner is based on the association rules mining methodology Agrawal et al. [1], a well known methodology in the data mining field. Usually association rules are discovered in transaction databases by finding and mining frequent item sets in an efficient manner, thus we formulated the problem of finding a set of candidate allelic variants correlated to the patient's classes, as the finding of Frequent Sets of allelic variants and then as the extraction of association rules from those frequent sets.

In particular DMET-Miner presents two order of innovations with respect to DMET-Analyzer: (i) In addition to DMET-Analyzer functions, DMET-Miner supports the extraction of association rules from DMET datasets, while DMET-Analyzer only supports the exhaustive execution of Fisher tests and related statistical corrections; (ii) DMET-Miner uses optimized data structures that gives good performance results in rule extraction also with huge DMET datasets. The use of a modular software architecture will also allow to easily add new machine learning algorithms with minimal effort.

In summary, DMET-Miner is a software platform able to easily read data produced by the Affymetrix DMET platform and then to extract relevant knowledge by computing frequent item sets in an efficient manner, as well as extracting association rules that link allelic variants in more than one probe with the conditions of patients (e.g. subjects responding or not responding to drugs in oncology). The whole DMET-Miner methodology is based on two main steps: (i) Transformation of a DMET dataset into a transaction database and (ii) learning of significant rules from the transaction database by mining frequent item sets.

The main contributions of the presented work are:

- The identification of candidate genotype variants related to pharmacogenomics of drugs (e.g. drug efficacy or Adverse Drug Reaction – ADR) by using frequent item sets and association rules mining.
- An efficient implementation of the association rule mining strategy that uses optimized data structures adapted to SNPs data, as opposed to general purpose data mining platforms.
- The full integration of this novel mining approach into DMET-Analyzer Guzzi et al. [7], an already available user friendly software, that is able to manage DMET data produced by DMET Console and that spares the user of coping with multiple tools/data format.
- A first step to integrate a data mining strategy into an overall clinical process related to pharmacogenomics. The validation of the effectiveness of our approach in a clinical scenario is under investigation by the medical group of our University.

To evaluate the effectiveness of our approach we considered various synthetic DMET datasets. We mined those datasets and compared the results using both our novel DMET-Miner rule learning methodology as well as the association rules mining algorithms provided in several open source data mining platforms, such as Weka and RapidMiner. Preliminary experiments on those synthetic DMET datasets show how DMET-Miner outperforms off-the-shelf data mining suites considering the response time and memory occupancy, while maintaining coherent rules.

The rest of the paper is structured as follows. Section 2 presents, respectively, the overall proposed methodology for mining frequent itemsets in a DMET SNP dataset (Section 2.1), the synthetic DMET datasets used in our experiments (Section 2.2), the DMET-Miner algorithm and its main data structures (Section 2.3), a short example about the use of the DMET-Miner software tool (Section 2.4), and the related work (Section 2.5). Section 3 presents experimental results obtained by mining a synthetic DMET dataset, and underlines the key form of learned association rules. Section 4 discusses possible approaches to interpret in a clinical context the association rules produce by DMET-Miner. Finally, Section 5 concludes the paper and outlines future perspectives.

[1] Genotyping or genotypization determines differences in the genetic profile (genotype) of an individual by examining the individual's DNA sequence and comparing it to another individual's sequence or to a reference sequence.

2. Materials and methods

2.1. Learning association rules from DMET SNP data

Association rule extraction is a common method in data mining used for discovering associations and relations among features in databases. Historically, the work from Agrawal et al. introduced such methodology for discovering associations in transaction databases in order to support marketing decision Agrawal et al. [1]. The work from Cheng et al. surveys main algorithms for mining frequent itemsets, Cheng et al. [4].

Here we focus on the development of a SNP-based association rule mining platform able to mine genomic data with application to pharmacogenomics. In fact, groups of SNPs may influence the efficacy or toxicity (adverse drug reaction) of drugs in different classes of patients Alonso et al. [2] and Cannataro et al. [3]. Our approach is based on the following steps:

- Initial Preprocessing of DMET dataset consisting in computing the frequency of each allele for each probe (this step is useful to compute Fisher's Test in an efficient way).
- Removing of possible uninformative probes by iteratively applying the Fisher's test.
- Generation of Transaction database, obtained by transposing the filtered input dataset.
- Learning Rules through an optimized FP-Growth algorithm.

2.2. Synthetic DMET SNP dataset

The DMET datasets used in our experiments have been obtained by using a *Random DMET Generator* (RDG) that we developed for this project. At the core of RDG, there is a random function $f_a(\cdot)$ able to generate a pairwise of alleles i.e. $\{a_1/a_2\}$, where $\{a_i, i = 1,2\}$ belongs to the alleles set symbols A. Formally: $\{a_i\} \in A = \{A, C, G, T, -\} \cup \{$"*NoCall*"$\}$. A contains all the nucleotides symbols and the special symbols: "–", that is used to denote insertion/deletion of nucleotides, and "*NoCall*", that is reported by the DMET platform when it has not been able to detect the nucleotide. To build a $m \times n$ SNP DMET table, RDG iteratively fills each element of the table (i,j), where i is the probe identifier (for current DMET platform, $m = 1936$) and j is the sample (e.g. patient) identifier. The user has to provide in input the number of samples n (a positive number) and the number of probes m (the default value is 1936) and the label (e.g. healthy vs diseased) of each sample (by default the first half of samples is labeled as class A and the remaining as class B). An example of synthetic SNP DMET dataset is reported in Table 1.

In order to extract relevant rules, we follow logical steps as depicted in Fig. 1. These steps are described in details below.

1. Loading and conversion of the input DMET dataset into a transaction database. Input data table produced by DMET Console (see e.g. Table 1) is initially loaded and transposed obtaining a $n \times m$ matrix of alleles named T, where n is the number of samples (patients) and m is the number of probes ($m = 1936$ for current DMET chips). In this way, each row of the table T represents a transaction, where all SNPs detected in a patients, on the various probes, are the items of the transaction. Table 2 shows the transformed matrix T for the input dataset of Table 1.

Table 2 is then used to extract itemsets. Thus, in order to explain the overall process, we here recall some main concepts that we use in the following.

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of items, where an item is identified by a specific SNP into a cell (i,j) of Table T.

Let T the set of transactions, formally a transaction over I is a couple $T = (tid, I)$, where tid is the transaction identifier and I is an item or itemset. The number of items present in a transaction is defined as *transaction width*. A transaction T_j contains an itemset J, if J is a subset of T_j, this is $J \subset T$.

Table 1
A simple DMET SNP microarray data set. S and P respectively refer to sample and probe identifiers.

Probes	Samples				
	S_1	S_2	S_3	⋯	S_N
P_1	G/A	A/G	A/G	⋯	T/T
P_2	G/A	A/G	A/G	⋯	T/C
⋮	⋮	⋮	⋮	⋯	⋮
P_M	G/A	A/G	A/G	⋯	T/C

Let $D = \{t_1, t_2, \ldots, t_m\}$ be a set of transactions, called *DMET-Dataset* D hereafter. Each transaction in D is identified by a unique *ID* of the corresponding sample or patient.

2. Computing frequent itemsets and extracting association rules. We now may start the mining phase by applying the following steps:

1. find all combinations of items in a set of transactions that occur with a specified minimum frequency threshold (called **Frequent Items Set**).
2. calculate rules that highlight the probability that one or more items are contained into frequent items sets.

The strength of frequent item sets extraction is based on the ability to discover interesting relationships hidden in large data sets. This feature is based on a very important property of the itemset also known as *Support count*. Support count refers to the number of transactions that contain a particular item or itemset. Formally, the Support-Count $S_c(\cdot)$ of an item $X, S_c(X)$ can be stated as follows: $S_c(X) = |\{\forall t_i \subset X \wedge t_i \in T\}|$, where $|\cdot|$ denotes the cardinality of the set. In other words, $S_c(X)$ is the fraction of transaction of T containing the item/item-set X.

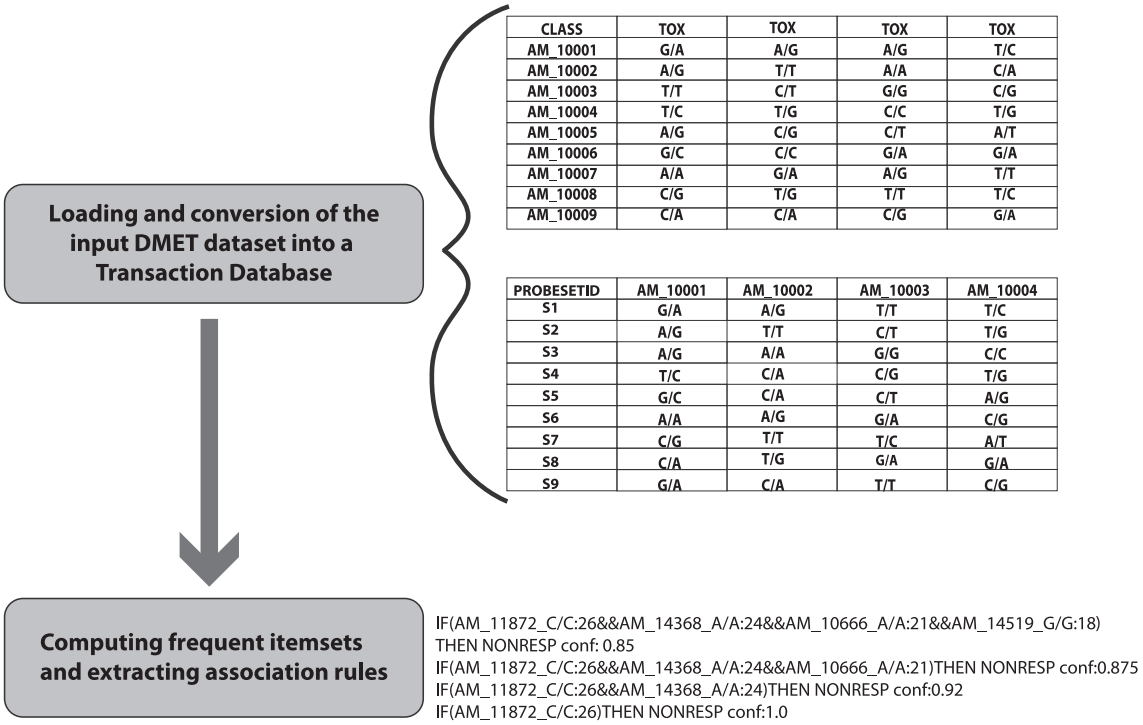
Association models extract rules that express the relationships among items into frequent itemsets. For example, a rule belonging to the frequent itemsets composed by the following elements $\{A/A, G/C, C/T\}$ might be stated as: **IF** $(A/A \wedge G/C)$ **THEN** C/T that can be read as: if A/A and G/C are included in a transaction, then C/T likely should also be included; this rule, for example, may be related with a complex disease such as cancer or diabetes.

2.3. The DMET-Miner algorithm

This section describes the core algorithm of DMET-Miner and its optimizations used to reduce the space search, i.e. used to reduce the number of frequent itemsets examined to extract association rules.

Given a dataset D, our problem is to discover all frequent patterns with respect to a user support threshold $min_{support}$ (Minimum Support).

After the transformation of the input table into a transaction database, the algorithm tries to reduce the search space through an apposite preprocessing methodology in order to reduce the number of possible candidate combinations. The filtering methodology is based on the use of the well known *Fisher's Test* and the filtering technique removes all the rows from the original DMET dataset for which it is not possible to accept the *null hypothesis* ($pvalue \geqslant F_{Thr}$). After the filtering step, the resulting table is transformed into a transaction database (let see Fig. 2). Such transformation is mandatory since the extraction of frequent itemsets is more efficient with this data format (see Fig. 2 (step a.1)). In order to discriminate among alleles of different probes that have the same name (i.e. A/A of the probe X and A/A of the probe Y) we



CLASS	TOX	TOX	TOX	TOX
AM_10001	G/A	A/G	A/G	T/C
AM_10002	A/G	T/T	A/A	C/A
AM_10003	T/T	C/T	G/G	C/G
AM_10004	T/C	T/G	C/C	T/G
AM_10005	A/G	C/G	C/T	A/T
AM_10006	G/C	C/C	G/A	G/A
AM_10007	A/A	G/A	A/G	T/T
AM_10008	C/G	T/G	T/T	T/C
AM_10009	C/A	C/A	C/G	G/A

PROBESETID	AM_10001	AM_10002	AM_10003	AM_10004
S1	G/A	A/G	T/T	T/C
S2	A/G	T/T	C/T	T/G
S3	A/G	A/A	G/G	C/C
S4	T/C	C/A	C/G	T/G
S5	G/C	C/A	C/T	A/G
S6	A/A	A/G	G/A	C/G
S7	C/G	T/T	T/C	A/T
S8	C/A	T/G	G/A	G/A
S9	G/A	C/A	T/T	C/G

Fig. 1. DMET-Miner Workflow. The right side of figure shows the main data formats from the input table to extracted rules.

Table 2
Transposed DMET microarray dataset. *S* and *P* respectively refer to sample and probe identifiers.

Samples	Probes			
	P_1	P_2	$\cdots$	P_M
S_1	G/A	G/A	$\cdots$	G/A
S_2	A/G	A/G	$\cdots$	A/G
S_3	A/G	A/G	$\cdots$	A/G
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
S_N	T/T	T/C	$\cdots$	T/C

designed a unique probe identification system by adding to each allele the name of the probe (i.e. X.A/A and Y.A/A) as shown in Fig. 2 (step a.2).

The resulting table is transformed into a **hashmap** data-structure called *Transaction DB* (*TDB*), where the transaction id (*TID*) is the entry of the map and the matching items set (value) is encoded into hash-set by means of a hash-function, making possible to compress the items and to guarantee constant time for standard operations such as: inserting, deleting and searching items in the hash-set (see Fig. 2 (step a.3)).

Despite the preprocessing, we noted that in real datasets the number of items that compose the *TDB* is huge enough. Thus, a further compression step is necessary in order to better manage the enumeration and generation of frequent itemsets. For this reason, we decided to represent the dataset by means of a tree structure as shown in Fig. 3, for which the traditional phase of frequent itemsets generation and enumeration is not necessary.

The Enumeration of the frequent itemsets is done using a *Depth-First-Search*, (DFS, in short), sorting in descending order the items in each transaction. It is usually best to process the items in the order of decreasing frequency. The reason for this behavior is that the average size of the conditional *TDB* tends to be smaller if the items are processed in this order. Moreover, the order of the items influences only the search time, not the result of the algorithms.

Another aspect to take into account in order to improve the execution time is how the *TDB* is represented. Algorithms that enumerate itemsets generally differ in how transaction databases are represented: *horizontally (Horizontal Representation)*, *vertically (Vertical Representation)*, or in a *hybrid fashion (Hybrid Representations)*, which combines a horizontal and a vertical representation.

In our solution we implemented a *FP-Tree (Frequent Pattern Tree)* inspired data-structure joined to a hybrid *TDB* representation. The main idea is to represent a *TDB* by means of a prefix tree, combining transactions with the same prefix. And in the same time, the *FP-Tree* keeps track of the same item contained in different transactions by linking the prefix tree nodes referring to the same item into a *frequent items list* (see Fig. 2 (step a.4)).

In order to have fast access to the FP-Tree node, the *frequent items list* is used as fast access point to each frequent item ($S_c(I) \geqslant min_{sup}$) discovered into the *TDB* and the related support, and modeled as a tree (see Fig. 2 (step a.4) and 3).

Our algorithm needs to scan twice the *TDB*. The first pass is necessary to discover the frequency of each item into the *TDB*, in order to fill the *frequent items list* with all items (frequent items) for which $S_c(I) \geqslant min_{sup}$ and sorting the items according to their descending frequency. The second *TDB* scan is related with the deletion of all items for which their support is $S_c(I) < min_{sup}$ and the remaining items are ordered according to descending frequency in the transactions. Sorting the items in the order of descending frequency allows to obtain a compact tree, limiting the number of different possible prefixes. This is done using the element contained into the *frequent items list* as *backbone* of the FP-Tree and mapping each transaction in the *TDB* on it.

The mapping is based on two main operations: *support-update* and *node-creation*. If during the mapping, the current element in the FP-Tree matches with the current element in the transaction, the function *support update*, that updates the support of the current node, is activated. Whereas if the current node in the FP-Tree does not match with the current node in the transaction, the function

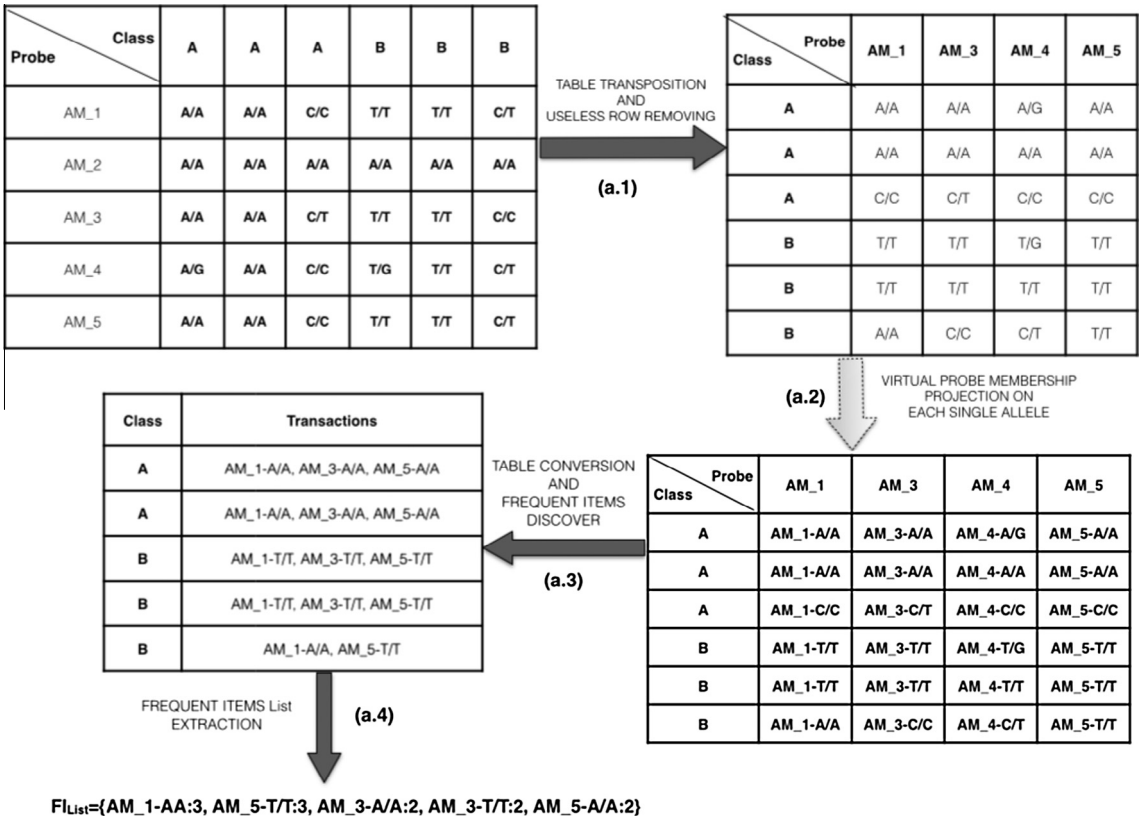


Fig. 2. Main phases of analysis into DMET-Miner.

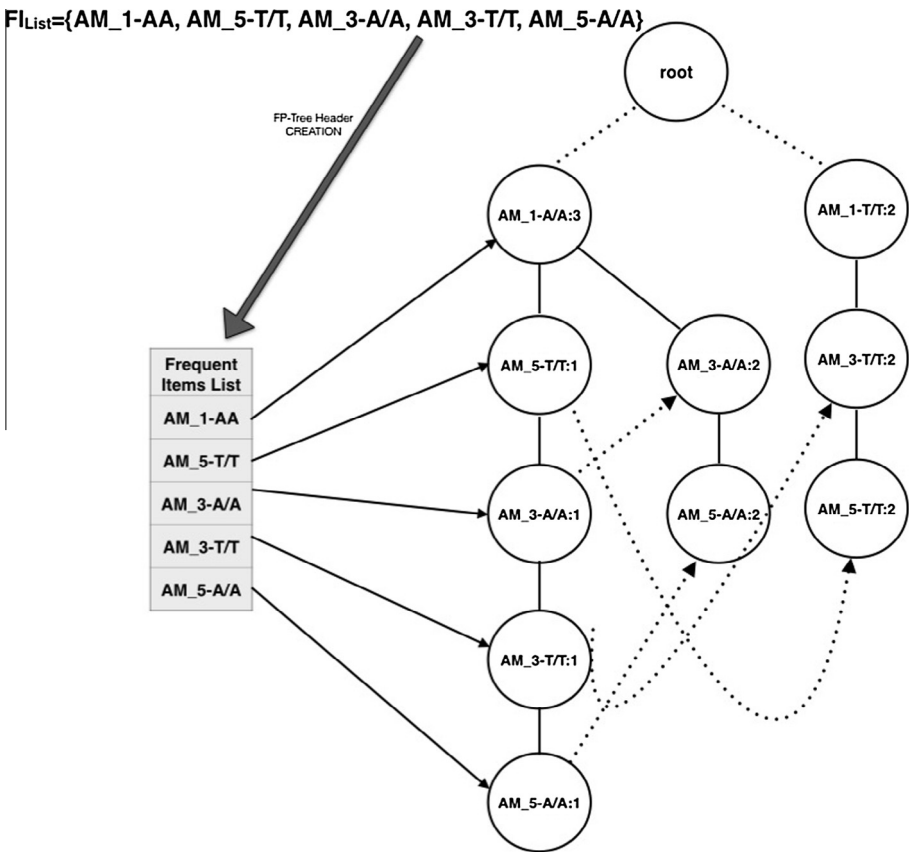


Fig. 3. Use of the frequent item list to access, in an efficient way, the FP-Tree.

node creation, starting from the current item, creates a new node, adding it in the FP-Tree as children of the current FP-Tree node and the remaining elements in the current transaction are added as children of the last created FP-Tree node (as shown in Fig. 4).

Finally, mining of frequent patterns is based on a recursive methodology of Tree visiting, in particular we defined an *inverted DFS (Deep First Search)* scan method to explore the FP-Tree.

Algorithm 1 shows the pseudo code of the core DMET-Miner algorithm.

Algorithm 1. DMET-Miner Core Algorithm.

Require: A DMET input dataset D
Data Structure initialization: $T, TDB, FPTree$
for all $rows \in T$ **do**
  if $(rowsDistribution \leqslant Thr)$ **then**
    $discard(row)$
  else
    $pvalue \leftarrow computeFisherTest(row)$
    if $(pvalue > \sigma_{Thr})$ **then**
      $discard(row)$
    end if
  end if
end for
for all $rows \in T$ **do**
  $TDB \leftarrow row$
  $frequentItemsList.supportUpdate \leftarrow row$
end for
for all $items \in TDB$ **do**
  if $itemfreq \leqslant minsupp$ **then**
    $TDB.remove(item)$
  end if
end for
for all $t \in TDB$ **do**
  $descendingSorting(items \in t)$
end for
$FPTree \leftarrow frequentItemsList$
for all $t \in TDB$ **do**
  map each item of t on FPTREE
  if *perfectmatch* **then**
    supportUpdate
  else
    nodeCreation
  end if
end for

for all $(item \in frequentItemsList)$ **do**
  $cpb \leftarrow (item, FPTree)$
  repeat
    if $cpbnode_{freq} < minsupp$ **then**
      $remove(cpbnode)$
    end if
  until $cpb = \varnothing$
  $saveRule()$
end for

2.4. Using DMET-Miner to discover association rules

Fig. 5 shows the workflow of execution of a typical analysis conducted using DMET-Miner.

Initially the user loads data into DMET-Miner (see Fig. 5a). Then, the user has to attribute the right class to each sample (see Fig. 5b) and can start the analysis method (in this case Mining Association Rules). The DMET input table is then filtered and transposed (see Fig. 5c). Finally, DMET-Miner extracts association rules from the data and shows the results in a new window (see Fig. 5d). Further information about interesting SNPs (i.e. those contained in the extracted association rules) are visualized to the user by providing SNP annotations provided by the Affymetrix libraries embedded in DMET-Miner, and by providing links to dbSNP and PharmaGKB databases.

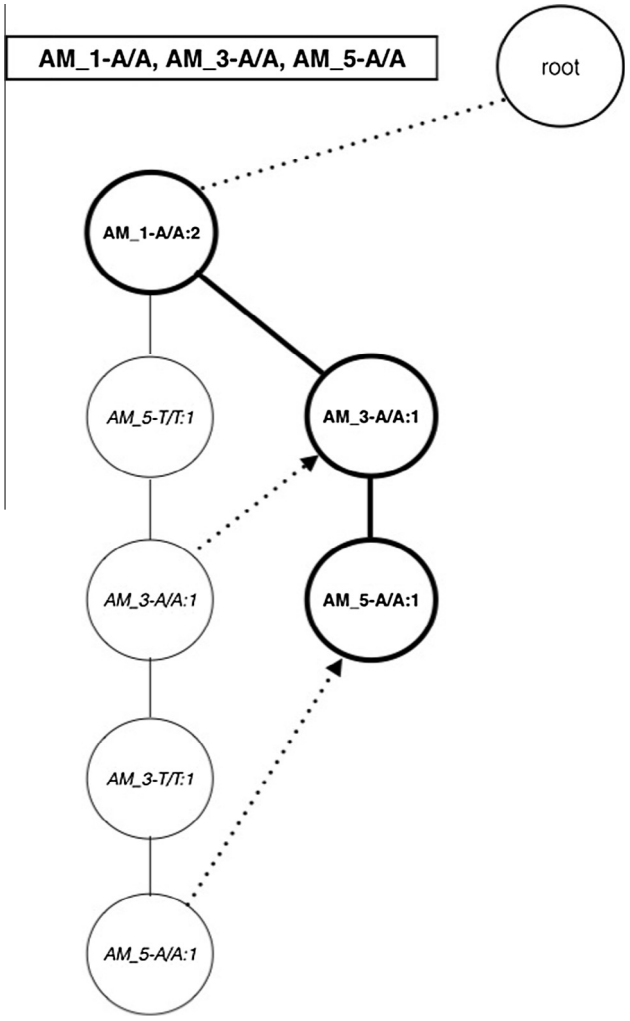


Fig. 4. Support-update and node-creation in DMET-Miner: when the item *AM_1-A/A* is added to the tree, it is necessary to just update its own support by increasing it; when we have to add the item *AM_3-A/A*, the function node-creation is called since there is not match between the current item and the current node in the tree; the last item (*AM_5-A/A*) into the transaction, is added as child of the node *AM_3-A/A*.

2.5. Related work

Search for frequent and significant patterns in data has been a central area in data mining and knowledge discovery field. More recently this methodology is largely applied in bioinformatics with different aiming. For lack of space we do not recall here the main association rules algorithm that we discuss more deeply on the DMET-Miner web site: https://sites.google.com/site/dmetminer/related-work. The interested reader may find a complete survey on Naulaerts et al. [13].

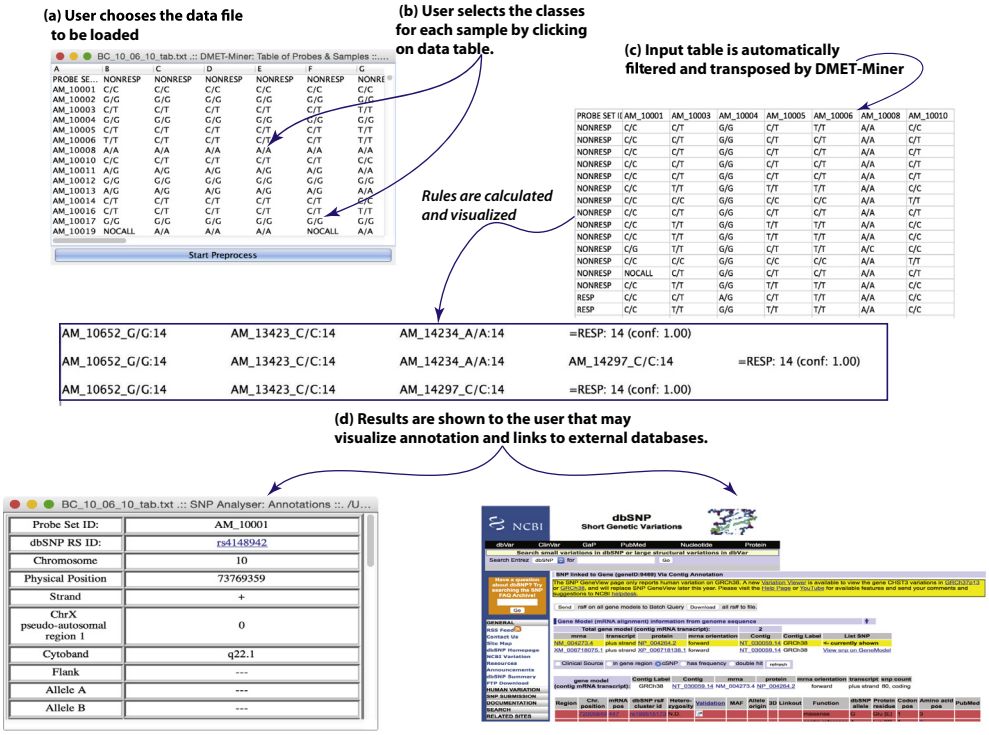


Fig. 5. The main steps necessary to analyze the input DMET dataset are the following: data selection and loading (a); user chooses the sample belonging to *classA* (b); the input dataset is filtered and automatically converted in transactions data (c); finally the tool displays the mined rules to the user offering the possibility to get information from local Affymetrix data or from remote repository such as *dbSNP* and *PharmaGKB*(d).

3. Results

In this section we present the performance evaluation of DMET-Miner and compare it with respect to other well known data mining tools such as *Weka* and *RapidMiner*. We have compared the FP-Growth algorithm implemented in DMET-Miner with the FP-Growth algorithm implemented in Weka (version 3.6.10) and the FP-Growth algorithm available in RapidMiner (version 5.3.013). Output from the DMET platform was preprocessed prior to running on Weka and RapidMiner. No pre-processing was required when using DMET-Miner because DMET-Miner is natively compatible with the DMET platform. All experiments were performed on a MacBookPro with a Pentium i7 2.3 GHz CPU, 16 GB RAM and a 512 GB SSD disk. The Frequent pattern mining algorithm of DMET-Miner was coded using Java 6 technology. As proof-of-principle we report the performance evaluation results of DMET-Miner compared to Weka and RapidMiner using several synthetic DMET datasets. The comparison of the FP-Growth algorithm implemented in the three software tools was tested using five synthetic DMET datasets. We built the synthetic datasets containing the same number of probes as a real DMET dataset (1936 probes) and doubling the number of samples (in the experiments we analyzed five datasets with respectively $25, 50, 100, 200$ and 400 samples for each dataset) grouped into two classes. We generated random datasets that contained significantly different distributions of SNPs.

The datasets contain data related to samples from subjects belonging to two classes: subjects which respond to drugs (*class RESP*) and subjects which do not respond to drugs (*class NONRESP*), simulating a classical *case-control study*. The dimensions of the datasets analyzed range from 300 KB for the dataset with 25 samples to about 3.1 MB for the one with 400 samples. The reported execution times and memory consumption refer to average times and the average memory usage, each value being computed repeating 10 times the measure with the same settings. Tables 5 and 6 show the computation times and the number of mined association rules using as input the dataset with 100 samples for all tools, varying *minimum support* and *confidence*. The computational time consumption during each run has been recorded with a Java and bash-shell script. In order to compare DMET-Miner FP-Growth with Weka FP-Growth and RapidMiner FP-Growth on the same conditions, we have given as input to Weka and RapidMiner the filtered dataset produced by our software. In this way we ensure that the inputs given to DMET-Miner, RapidMiner and Weka have the same dimensions. We point out that the filtering of probes using the Fisher test available in DMET-Miner allows a great reduction in the size of the analyzed dataset (see Table 3). To show the impact of this feature of DMET-Miner, we increased the number of analyzed probes and our results show increases in both execution time and memory usage that result in memory overflow problems shown in Table 4. Therefore using Weka and RapidMiner without filtering becomes impracticable on dataset's similar in size to real datasets (that have at least 1936 probes).

Briefly we used the cleaned dataset produced by DMET-Miner, transforming it into a Comma Separated Value file (CSV) ready to be analyzed with Weka and RapidMiner. We should point out that

Table 3
Number of meaningful probes (rows) after using the Fisher Test Filtering. #*S* indicates the total number of subjects belonging to the database under analysis.

#*S*	#Probes	#Probes using filtering
25	1936	45
50	1936	48
100	1936	57
200	1936	76
400	1936	78

Table 4
Variation of the computational time and memory consumption related with the increase of the number of meaningful probes to analyze. In the table, $\#S$ indicates the total number of subjects belonging to the database under analysis and ΔT_{ms} indicates the run time expressed in milliseconds (ms) to generate association rule.

$\#S$	#Probes	DMET-Miner		RapidMiner		Weka	
		ΔT_{ms}	Memory consumption	ΔT_{ms}	Memory consumption	ΔT_{ms}	Memory consumption
25	45	0.251	24.87 MB	4070.00	944.0 MB	1645.00	304.4 MB
25	100	0.294	26.84 MB	4530.00	1.2 GB	1831.00	672.8 MB
25	200	0.614	28.60 MB	23,000	1.7 GB	3985.00	708.5 MB
25	300	0.604	39.79 MB	∞	*Heap Overflow*	4471.00	746.2 MB
25	350	∞	*Heap Overflow*	∞	*Heap Overflow*	∞	*Heap Overflow*
25	400	∞	*Heap Overflow*	∞	*Heap Overflow*	∞	*Heap Overflow*

the average time needed for this preprocessing step is around 8422.6 ms, that should to be added to the times in Tables 5 and 6. The time cost of the mined rules is approximately proportionate to dataset size, although it is also affected by the complexity of dataset (e.g. the filtering can greatly simplify the dataset). The number of extracted rules in addition to dataset size is also affected by the values used for the Minimum Support and Confidence. In Weka and DMET-Miner the number of mined rules depends by the values of Minimum Support and Confidence, instead in RapidMiner it is possible to set only the Minimum Support. Support and Confidence work as a double filter in Weka and DMET-Miner reducing in a remarkable way the number of rules which will be reported to the users. For this reason, RapidMiner extracts more rules than DMET-Miner and Weka. In particular, the quality of rules produced by the tools, should be taken into account. Even though in some cases Rapid-Miner produces a huge number of rules with respect to Weka and DMET-Miner, their quality is usually poor. In fact, the top rules extracted by Rapid-Miner are trivial rules since they are composed only by one term and thus they are very poor rules from an informative point of view (see Table 7). This feature could be misleading for the user that has to move manually through the rules looking for significant ones. On the other side, Weka and DMET-Miner provide, as top rules, the rules composed by more than one element, thus avoiding the user to manually look for meaningful rules. In Table 7 the first 10 rules extracted by the three tools are conveyed.

Considering the DMET-Miner results (see Tables 5 and 6) it is possible to note that execution times are directly proportional to the number of mined rules and strictly related with the values of confidence and support. On the other hand, considering Weka results (see Table 5), execution time increases when increasing the confidence, independently from the number of mined rules.

Table 7
RapidMiner, Weka and DMET-Miner: first top 10 extracted rules from the dataset composed by 100 samples (Confidence = 60% and Support = 30%). Common rules mined by the different tools, are represented using the same background color.

RapidMiner	Weka	DMET-Miner
AM_13941_C/T	AM_11119=C/A	AM_11119_C/A
	AM_10593=G/C	AM_10593_G/C
AM_12159_A/T	AM_14653=T/G	AM_14653_T/G
	AM_10593=G/C	AM_10593_G/C
AM_15506_T/A	AM_10593=G/C	AM_13854_A/T
	AM_11119=C/A	AM_12136_T/C
AM_15404_A/T	AM_10659=A/C	AM_12254_G/G
	AM_10593=G/C	AM_10524_T/C
AM_14535_C/A	AM_13764=G/T	AM_10329_A/T
	AM_10593=G/C	AM_11313_A/A
AM_12529_T/A	AM_11134=T/T	AM_10202_C/G
	AM_10593=G/C	AM_12323_T/T
AM_10976_A/A	AM_13764=G/T	AM_10106_C/T
	AM_10659=A/C	AM_10524_T/C
AM_10589_A/G	AM_11134=T/T	AM_11346_C/T
	AM_10571=A/A	AM_10524_T/C
AM_15299_A/A	AM_10571=A/A	AM_10072_C/C
	AM_11134=T/T	AM_10524_T/C
AM_15238_A/C	AM_10593=G/C	AM_14368_C/T
	AM_10659=A/C	AM_15439_T/T

Table 5
DMET-Miner versus Weka execution time and number of mined rules when varying the confidence (DMET-Miner and Weka Support = 50%). In the table, $\#S$ indicates the total number of subjects belonging to the database under analysis, $mS\%$ indicates the minimum Support value in percentage, $C\%$ indicates the Confidence value in percentage, $\#R$ indicates the number of mined association rules, ΔT indicates the run time expressed in milliseconds (ms) and δT_U indicates the time in (ms) necessary to generate a single association rule.

$\#S$	DMET-Miner					Weka				
	$mS\%$	$C\%$	$\#R$	ΔT	δT_U	$mS\%$	$C\%$	$\#R$	ΔT	δT_U
100	50	30	100	251.33	2.51	50	30	28	1645.00	58.75
100	50	50	100	255.00	2.55	50	50	28	1693.00	60.46
100	50	70	100	259.00	2.59	50	70	23	1748.00	76.00
100	50	90	1	255.33	255.33	50	90	0	1774.50	–

Table 6
DMET-Miner versus RapidMiner execution time and number of mined rules when varying the support (DMET-Miner Confidence = 60%). In the table, $\#S$ indicates the abbreviation of number of samples that form the dataset, $mS\%$ indicates the minimum Support value in percentage, $C\%$ indicates the Confidence value in percentage, $\#R$ indicates the number of mined association rules, ΔT indicates the run time expressed in milliseconds and δT_U indicates the time in (ms) necessary to generate a single association rule.

$\#S$	DMET-Miner					RapidMiner				
	$mS\%$	$C\%$	$\#R$	ΔT	δT_U	$mS\%$	$C\%$	$\#R$	ΔT	δT_U
100	30	60	589	300.13	0.51	30	–	12,604	4070.00	0.32
100	50	60	100	188.00	1.88	50	–	215	3943.00	18.34
100	70	60	1	178.71	178.71	70	–	10	2701.00	270.1
100	90	60	1	176.52	176.52	90	–	0	2629.00	–

Finally, examining RapidMiner results (see Table 6), it is possible to note that the computational time decreases when the minimum support increases, since the search space contracts for high values of minimum support. Regarding the number of learned rules, while DMET-Miner is able to learn some rules also when confidence is very high (e.g. *confidence* $= 90\%$ see Tables 5 and 6), Weka stops learning rules when *confidence* $> 70\%$ (see Table 5), whereas RapidMiner stops when *support* $> 70\%$ (see Table 6). To better pinpoint out the capability of the three different implementation of FP-Growth algorithms, we measured the execution time (seconds) and the number of association rules at different levels of minimum support ($20\%, 40\%, 60\%, 80\%$, and 100%) and using the 100 samples dataset. The minimum confidence was always set to zero. That is, we required no confidence since in RapidMiner FP-Growth algorithm it is not possible to set the confidence value. Results are summarized in Table 8.

Analyzing the results in Table 8, it seems that RapidMiner outperforms in terms of learned rules DMET-Miner. RapidMiner FP-Growth algorithm iteratively reduces the minimum support until it finds the required number of rules (how delineated in the manual), instead in DMET-Miner FP-Growth algorithm, the minimum support is never decreased. Thus for minimum support equal to 60%, both DMET-Miner and RapidMiner learn only one meaningful rule composed by two terms: $AM\_10976\_A/A, AM\_13941\_C/T$, while the remaining additional 77 rules mined by RapidMiner are trivial ones, because they are composed by only one term. Moreover, when the minimum support is low, due to the huge number of mined rules, the computational time is high. Whereas, when the minimum support is equals to 60% for Weka or 80% for RapidMiner and DMET-Miner, all the algorithms finish within a second, or even less than a second, as DMET-Miner (see Fig. 6).

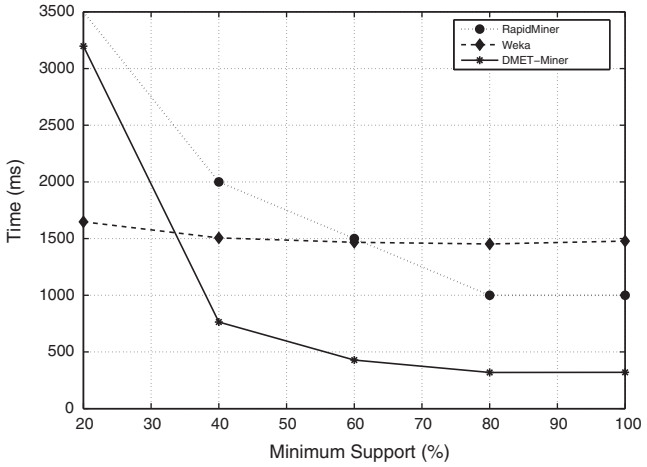


Fig. 6. Execution time of the algorithms varying the minimum support using the 100 samples dataset. The execution time is obtained when varying the value of minimum support. The dotted line (top part of figure) represents the running time of RapidMiner, the dashed line (middle part of figure) represents running time of Weka, the continuous line (bottom part of figure) represents the running time of DMET-Miner.

4. Discussion

DMET-Miner is the first software platform mainly devoted to help researchers to easily extract relevant knowledge related with complex diseases from DMET genotyping data. DMET-Miner has been designed to work well with DMET microarray data and it is based on a customized version of the FP-Growth algorithm able to extract association rules from DMET genotyping data. Customization allows DMET-Miner to achieve better results than Weka and RapidMiner both in terms of memory consumption, execution times and mined association rules. The information about maximum memory occupancy for the three tools applied to different datasets is illustrated in Fig. 7 and summarized in Table 9. The maximum memory occupancy during each run has been recorded with a Java and a bash-shell script. Compared with RapidMiner and Weka tools, DMET-Miner is less expensive in terms of RAM memory usage. Out of these tools, DMET-Miner runs in rather less time than others and the RAM usage increases slowly when increasing the dataset size. On the other side, Weka runs in rather less time than RapidMiner, but the RAM usage for both tools increases when increasing dataset size more than in DMET-Miner, as shown in Fig. 7. In conclusion, the DMET-Miner FP-Growth algorithm is more efficient and requires less memory than the other two tools (see Fig. 7).

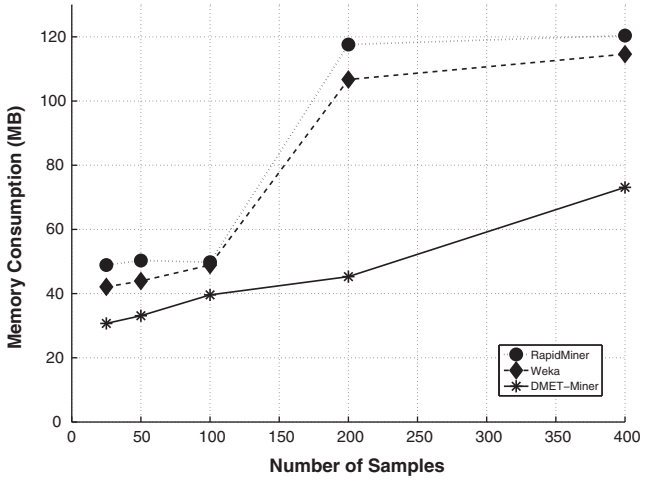


Fig. 7. Memory consumption of RapidMiner, Weka and DMET-Miner when doubling the size of the input dataset. The dotted line (top part of figure) represents the memory consumption of RapidMiner, the dashed line (middle part of figure) represents the memory consumption time of Weka, the continuous line (bottom part of figure) represents the memory consumption of DMET-Miner.

Table 8
DMET-Miner, RapidMiner and Weka execution time and number of mined rules, varying the support (DMET-Miner and Weka Confidence = 0), using the 100 samples dataset. In the table, $mS\%$ indicates the minimum Support value in percentage, $\#R$ indicates the number of mined association rules, ΔT indicates the run time expressed in milliseconds and δT_U indicates the time necessary to generate a single association rule.

$mS\%$	RapidMiner			Weka			DMET-Miner		
	$\#R$	ΔT	δT_U	$\#R$	ΔT	δT_U	$\#R$	ΔT	δT_U
20	255,910	3500	0.01	28	1648	58.86	229,481	3198	0.01
40	1498	2000	1.34	28	1506	53.79	1347	764	0.57
60	78	1500	19.23	0	1467	–	1	428	428
80	0	1000	–	0	1452	–	0	319	–
100	0	1000	–	0	1478	–	0	320	–

4.1. A possible way to read and interpret the mined rules from biomedical researchers

The *one-error*, *one-disease* approach has not held up for complex diseases such as cancer or Alzheimer for example. The causes of complex disorders such as diabetes, heart disease, glaucoma and cancer, however, are much more complex, since they do not have a single genetic cause, but they are likely associated with the effects of multiple factors. Complex disorders are difficult to study and treat because many of the specific factors that cause them are yet unknown. Although some experimental platforms for investigating the genotype of organisms are appearing, the management and analysis of the produced data is not easy. DMET-Miner has been designed to work well with DMET microarray data and uses a customized version of the FP-Growth algorithm able to extract association rules from DMET genotyping data in an efficient way. Identifying those combinations of frequent items (SNPs) and conveying them to researcher in the form of association rules, could aim the researcher to clarify the mechanisms of the diseases, improving the efficiency and quality of health care. Each extracted association rule is a container of multiple mutations (SNPs) that occur within a gene or in a regulatory region near a gene, which play a more direct role by affecting the gene's function, for example. Due to space limitation we only show and discuss two association rules extracted by DMET-Miner. The discusses association rules are the following:

- *Rule_1:*
  AM_10659_A/A:36, *AM_14535_C/A*:37, *AM_10976_A/A*:37, *AM_10589_A/G*:37, *AM_13941_C/T*:40, *AM_12159_A/T*:40;
- *Rule_2:*
  AM_14673_T/C:34, *AM_13871_C/A*:34, *AM_14368_C/T*:35, *AM_14108_C/G*:35, *AM_10072_C/C*:36, *AM_15439_T/T*:38;

An item in the rules is composed from three parts: the probe identifier (probeid), the allele, and the support (frequency), that are respectively *AM_10659*, *A/A*, and 36, for the first item of Rule_1. The *probeid* is an identifier of a specific region of the *DNA* called *Chromosome*, the allele is the value detected by the DMET platform and provided by the DMET-Console tool, and the last value is the support (frequency) of the term within the population. A more easy way to understand the meaning of the rules is to translate each *probeid* with its own gene name. Translating each probe identifier with the associated gene name, allows a researcher to get a more complete insight in the biological functions of the gene that probably is not working properly due to the mutation. The translated versions of the association rules mentioned before, are reported in Tables 10 and 11.

After the translation process, it is more easy to analyze the mined rules (see Tables 10 and 11). A possible way to interpret the rules extracted from the synthetic dataset is given in the following. Considering Rule_1, it should be noted that all mutations refer to chromosomes spatially close together and, in particular, two mutations occur in the same chromosome (see Table 10, Chromosome column). Moreover, mutations have a negative effect on the following biological activities: *synonymous codon*, *nc transcript variant* and *missense* sub-processess of *Translational Process*. The *Translational Process* is the process that makes possible to convert a chain of three nucleotides in an aminoacid. These mutations affect the genes that control the *Translational Process*. A mutation in one of the three genes affects all the others. A mutation into the synonymous codon means a production of a different start or stop signal for the *Translational Process*, augmenting the number of missense and deregulating the transcript variant of a non-coding RNA gene. The rules extracted by DMET-Miner joined with the annotation freely available on the *dbSNP* or *PharmGKB* knowledge bases, should easily allow to know hidden interactions among different genes. Moreover, this new information should help researchers to better understand the dynamics that govern the interaction among genes and consequently, to bring to light the unknown functionality among genes, involved for example in drug metabolization.

Table 9
RapidMiner, Weka and DMET-Miner memory consumption when varying the dataset size. In the table, *#S* indicates the total number of subjects belonging to the database under analysis.

#S	RapidMiner $MemoryUsed_{(MB)}$	Weka $MemoryUsed_{(MB)}$	DMET-Miner $MemoryUsed_{(MB)}$
25	48.9	42.1	30.7
50	50.3	43.9	33.3
100	49.8	48.9	39.6
200	117.6	106.7	39.7
400	120.4	114.6	73.1

Table 10
Rule_1 translation.

ProbeId	DMET detected allele	Gene name	Chromosome	Functional consequence
AM_10659	A/A	SLC15A1	13:98723927	missense
AM_14535	C/A	ABCB4	7:87417435	synonymous codon
AM_10976	A/A	ABCC6	16:16182880	missense
AM_10589	A/G	ATP7B	13:51939130	missense, nc transcript variant
AM_13941	C/T	SLC22A5	5:132388947	synonymous codon, nc transcript variant
AM_12159	A/T	SLCO4A1	20:62657071	missense, nc transcript variant

Table 11
Rule_2 translation.

ProbeId	DMET detected allele	Gene name	Chromosome	Functional consequence
AM_14673	T/C	PPP1R9A	7:95295992	utr variant 3 prime
AM_13871	C/A	HMGCR	5:75360350	utr variant 3 prime
AM_14368	C/T	SLC22A1	6:160139851	cds indel
AM_14108	C/G	PPARD	6:35356964	intron variant
AM_10072	C/C	CYP2C19	10:94781999	splice donor variant
AM_15439	T/T	ATP7A	X:77988686	intron variant, missense

5. Conclusion

Personalized medicine is an ongoing effort in the medical community which aims to realize therapies and drugs tailored to the single patient. The rationale of this interest is based on the consideration that the response to the drugs is strictly related to the genotype of each patient. Such discipline is based on the use of technologies able to investigate the genotype of patients such as the Affymetrix DMET platform. Besides the importance of using the DMET technology into genotype-based personalized medicine, there is a lack of comprehensive tools able to mine efficiently DMET data.

In this paper we presented DMET-Miner, a software platform for the analysis of Affymetrix DMET genotyping data, able to extract relevant knowledge from DMET data in an efficient manner, by computing frequent itemsets and by extracting association rules that link different alleles to clinical conditions.

Preliminary performance evaluation shows how the novel DMET-Miner association rule extraction algorithm outperforms off-the-shelf well known algorithms, as those provided in the Weka and RapidMiner platforms. Moreover, preliminary results on a real DMET dataset, currently under medical investigation, seems to demonstrate the biological relevance of the extracted association rules and the effectiveness of the proposed approach.

Availability of the software

Project home page: https://sites.google.com/site/dmetminer/. **Requirements:** Java 1.6.1 Runtime or higher. **License:** Creative Common License: BY-NC-ND. This version of the software is for academic purposes only.

Conflict of interest

We have none conflict of interest.

Acknowledgements

This work has been partially funded by the following research projects funded by the Italian Ministry of Education and Research (MIUR): "DICET-INMOTO-ORCHESTRA" (PON04a2_D) and "BA2Know-Business Analytics to Know" (PON03PE_00001_1).

References

[1] R. Agrawal, T. Imieliński, A. Swami, Mining association rules between sets of items in large databases, SIGMOD Rec. 22 (2) (1993) 207–216, http://dx.doi.org/10.1145/170036.170072.

[2] N. Alonso, G. Lucas, P. Hysi, Big Data Challenges in Bone Research: Genome-wide Association Studies and Next-generation Sequencing, BoneKEy Reports 4, 2015.

[3] M. Cannataro, P.H. Guzzi, A. Sarica, Data mining and life sciences applications on the grid, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 3 (3) (2013) 216–238.

[4] J. Cheng, Y. Ke, W. Ng, A survey on algorithms for mining frequent itemsets over data streams, Knowl. Inform. Syst. 16 (1) (2008) 1–27, http://dx.doi.org/10.1007/s10115-007-0092-4.

[5] M.T. Di Martino, M. Arbitrio, P.H. Guzzi, E. Leone, F. Baudi, E. Piro, T. Prantera, I. Cucinotto, T. Calimeri, M. Rossi, P. Veltri, M. Cannataro, P. Tagliaferri, P. Tassone, A peroxisome proliferator-activated receptor gamma (pparg) polymorphism is associated with zoledronic acid-related osteonecrosis of the jaw in multiple myeloma patients: analysis by dmet microarray profiling, Br. J. Haematol. 56 (2011) 529–533, http://dx.doi.org/10.1111/j.1365-2141.2011.08622.x.

[6] M.T. DiMartino, M. Arbitrio, E. Leone, P.H. Guzzi, M. Saveria Rotundo, D. Ciliberto, V. Tomaino, F. Fabiani, D. Talarico, P. Sperlongano, P. Doldo, M. Cannataro, M. Caraglia, P. Tassone, P. Tagliaferri, Single nucleotide polymorphisms of ABCC5 and ABCG1 transporter genes correlate to irinotecan-associated gastrointestinal toxicity in colorectal cancer patients: a DMET microarray profiling study, Cancer Biol. Ther. 12 (9) (2011) 780–787 (November 1).

[7] P. Guzzi, G. Agapito, M. Di Martino, M. Arbitrio, P. Tassone, P. Tagliaferri, M. Cannataro, Dmet-analyzer: automatic analysis of affymetrix dmet data, BMC Bioinform. 13 (1) (2012) 258. <http://www.biomedcentral.com/1471-2105/13/258>.

[8] T. Hernandez-Boussard, M. Whirl-Carrillo, J.M. Hebert, L. Gong, R. Owen, M. Gong, W. Gor, F. Liu, C. Truong, R. Whaley, M. Woon, T. Zhou, R.B. Altman, T.E. Klein, The pharmacogenetics and pharmacogenomics knowledge base: accentuating the knowledge, Nucl. Acids Res. 36 (Suppl 1) (2008) D913–D918. <http://nar.oxfordjournals.org/content/36/suppl_1/D913.abstract>.

[9] Y. Hu, E.A. Ehli, K. Nelson, K. Bohlen, C. Lynch, P. Huizenga, J. Kittlelsrud, T.J. Soundy, G.E. Davies, Genotyping performance between saliva and blood-derived genomic dnas on the dmet array: a comparison, PloS One 7 (3) (2012) e33968.

[10] J. Li, L. Zhang, H. Zhou, M. Stoneking, K. Tang, Global patterns of genetic diversity and signals of natural selection for human ADME genes, Hum. Mol. Genet. 20 (3) (2011) 528–540.

[11] G. Lombardi, E. Rumiato, R. Bertorelle, D. Saggioro, P. Farina, A. Della Puppa, F. Zustovich, F. Berti, V. Sacchetto, R. Marcato, et al., Clinical and genetic factors associated with severe hematological toxicity in glioblastoma patients during radiation plus temozolomide treatment: a prospective study, Am. J. Clin. Oncol. doi 10 (2013) 1097.

[12] C. Mizzi, B. Peters, C. Mitropoulou, K. Mitropoulos, T. Katsila, M.R. Agarwal, R.H. van Schaik, R. Drmanac, J. Borg, G.P. Patrinos, Personalized pharmacogenomics profiling using whole-genome sequencing, Pharmacogenomics 15 (9) (2014) 1223–1234.

[13] S. Naulaerts, P. Meysman, W. Bittremieux, T.N. Vu, W. Vanden Berghe, B. Goethals, K. Laukens, A primer to frequent itemset mining for bioinformatics, Briefings Bioinform. (2013). <http://bib.oxfordjournals.org/content/early/2013/10/26/bib.bbt074.abstract>.

[14] E. Rumiato, E. Boldrin, A. Amadori, D. Saggioro, Dmet (drug-metabolizing enzymes and transporters) microarray analysis of colorectal cancer patients with severe 5-fluorouraci-induced toxicity, Cancer Chemother. Pharmacol. 72 (2) (2013) 483–488. <http://dx.doi.org/10.1007/s00280-013-2210-1>.

[15] A. Shiotani, T. Murao, Y. Fujita, Y. Fujimura, T. Sakakibara, T. Kamada, K. Nishio, K. Haruma, 721 novel single nucleotide polymorphism markers for low dose aspirin-associated small bowel bleeding: a dmet microarray profiling study, Gastroenterology 146 (5) (2014) S-126.

[16] T. Sissung, B. English, D. Venzon, W. Figg, J. Deeken, Clinical pharmacology and pharmacogenetics in a genomics era: the dmet platform, Pharmacogenomics 11 (2010) 89–103.

[17] B. Stranger, E. Stahl, T. Raj, Progress and promise of genome-wide association studies for human complex trait genetics, Genetics 187 (2) (2011) 367–383.

[18] S.M. Wakil, C. Nguyen, N.P. Muiya, E. Andres, A. Lykowska-Tarnowska, B. Baz, A.I. Tahir, B.F. Meyer, G. Morahan, N. Dzimiri, The affymetrix dmet plus platform reveals unique distribution of adme-related variants in ethnic arabs, Disease Markers (2015).

[19] X. Wang, L. Liotta, Clinical bioinformatics: a new emerging science, J. Clin. Bioinform. 1 (2011) 1.