Microarray

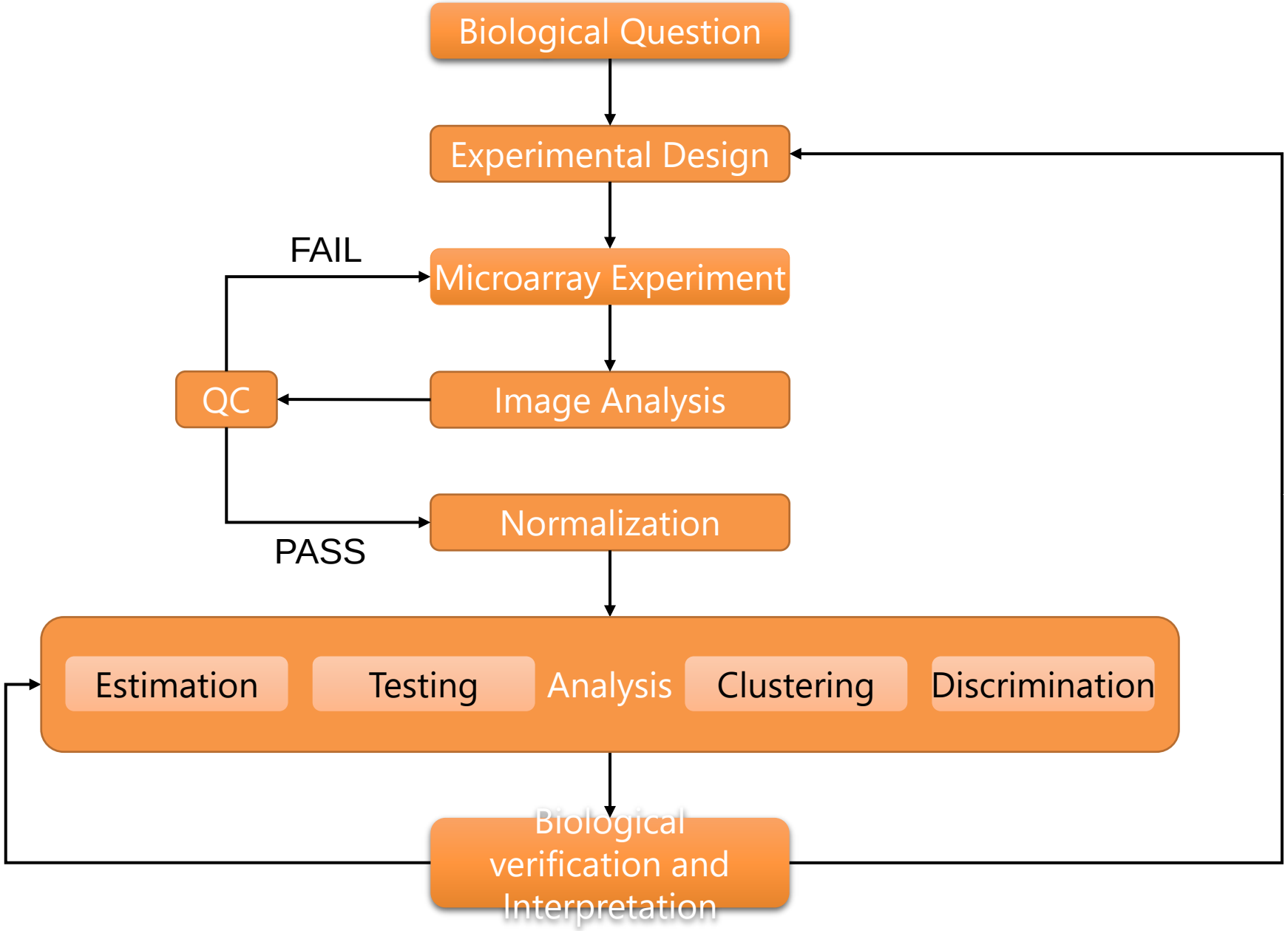
Salvatore Alaimo, MSc.

Email: alaimos@dmi.unict.it

Microarray

- A parte poche eccezioni, ogni cellula del nostro organismo contiene un set completo di cromosomi composti da geni identici.
- In una specifica cellula, solo una parte di questi geni è attiva, e sono proprio i diversi gruppi di geni attivi che conferiscono proprietà specifiche ad ogni tipo cellulare.
- Per "espressione genetica" si intende la produzione di proteina da parte di un gene (la trascrizione delle informazioni contenute sul DNA nell'mRNA che a sua volta viene tradotto nelle proteine che provvedono alle funzioni di base delle cellule).
- Il tipo e la quantità di mRNA prodotto ci dicono quanto un gene sia espresso.
- Ad esempio una alterazione dell'espressione genica può indicare la presenza di una malattia.

Microarray - Workflow

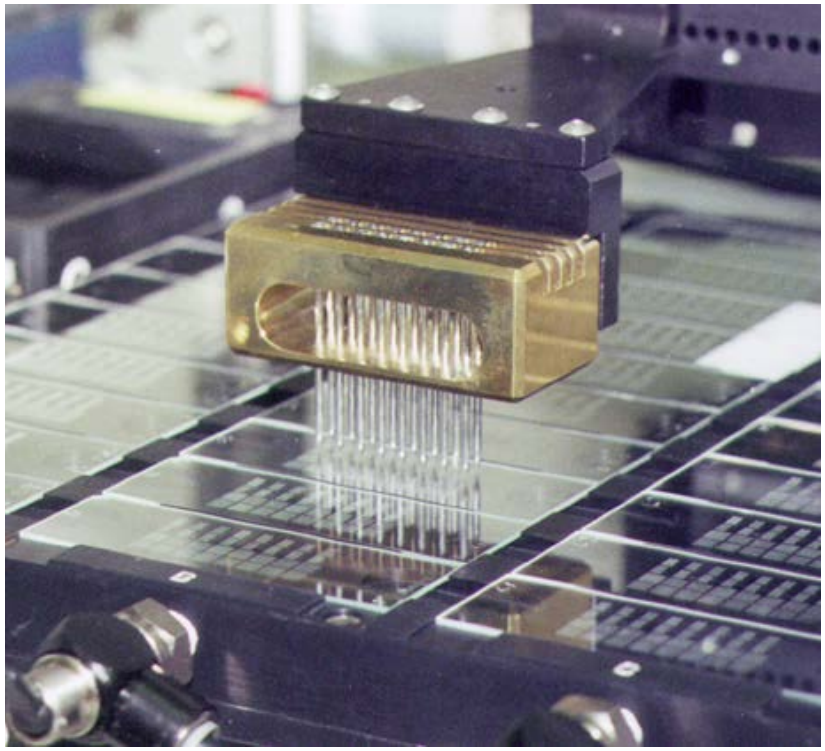


Microarray

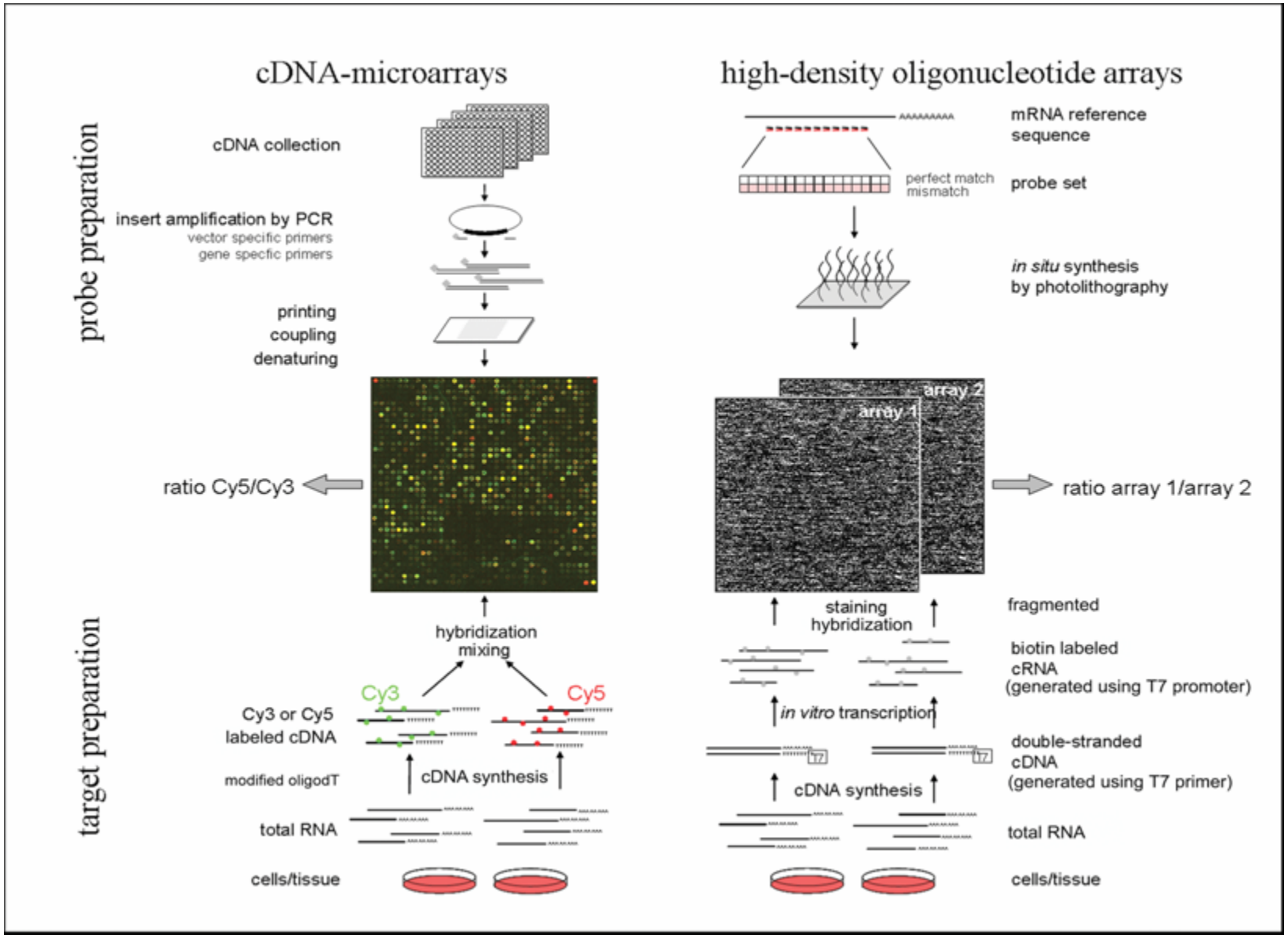
- Analizzare l'espressione genica vuol dire analizzare la quantità di mRNA o di proteine prodotte da una cellula in un particolare momento.
- I Microarray, chiamati anche DNA chip, permettono l'analisi dell'espressione di migliaia di geni con un singolo esperimento.
- Il principio alla base dell'analisi dell'espressione genica consiste nel confronto di campioni diversi, ad esempio tessuti sani o malati per studiare l'espressione genica in una determinata malattia.

Microarray

- Le molecole di mRNA si legano selettivamente, attraverso l'appaiamento delle basi, ad una sequenza di DNA complementare;
- Migliaia di sequenze di DNA a singolo filamento vengono posizionate su una griglia microscopica di pochi centimetri, che funge da supporto per l'appaiamento di molecole di mRNA che vengono poste sulla sua superficie (con l'ausilio di robot);



Microarray


cDNA-microarrays
high-density oligonucleotide arrays
probe preparation
target preparation
cDNA collection
insert amplification by PCR
vector specific primers
gene specfic primers
printing
coupling
denaturing
ratio Cy5/Cy3
hybridization
mixing
Cy3
Cy5
Cy3 or Cy5
labeled cDNA
modified oligodT
cDNA synthesis
total RNA
cells/tissue
mRNA reference
sequence
perfect match
mismatch
probe set
in situ synthesis
by photolithography
array 2
array 1
ratio array 1/array 2
staining
hybridization
fragmented
biotin labeled
cRNA
(generated using T7 promoter)
in vitro transcription
double-stranded
cDNA
(generated using T7 primer)
cDNA synthesis
total RNA
cells/tissue

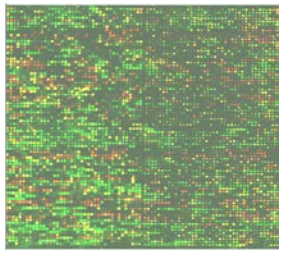
Microarray – cDNA vs high density oligonucleotides

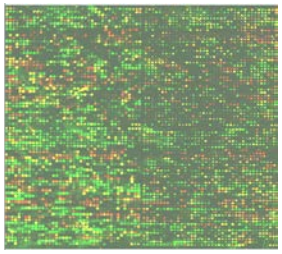
- Lunghezza delle probe ~20nt
- Ogni probe è fissata su un vetro
- Due sample per chip
- Labelling con due coloranti Cy3/Cy5
- Comparazione diretta di due campioni sulla base del modo con cui si ibridizzano alle probe
- Problema 1: cross-hybridization
- Problema 2: riproducibilità

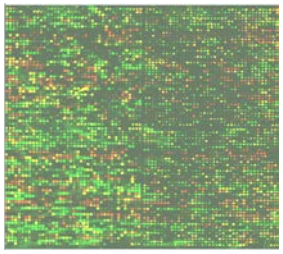
- Lunghezza delle Probe 20-25nt
- Probe sintetizzata in situ
- Un solo sample per chip
- Labelling con un solo colorante
- Un gene, molte probe diverse.
- Problema: cross-hybridization

Microarray

- Come già detto si confrontano le espressioni geniche per correlarle a malattie. Un tipico esempio è quello di confrontare l'espressione genica di un certo numero di geni in diversi campioni (=pazienti) appartenenti a diverse classi (=norm vs cancer).


Array1


Array2


Array3

...

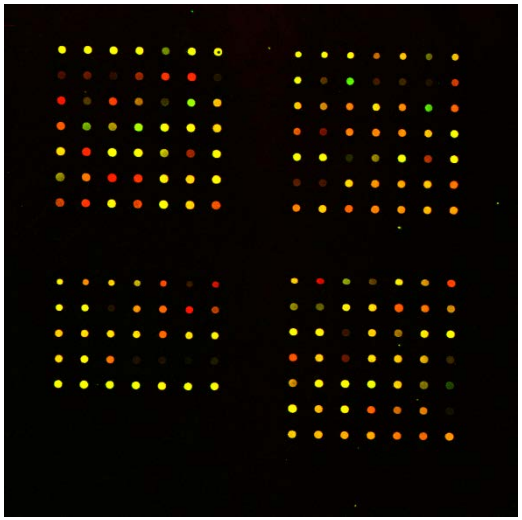
Arrays

	Array1	Array2	Array3	Array4	Array5	...
Gene1	0.46	0.30	0.80	1.51	0.90	...
Gene2	0.10	0.49	0.24	0.06	0.46	...
Gene3	0.15	0.74	0.04	0.10	0.20	...
Gene4	0.45	1.03	0.79	0.56	0.32	...
Gene5	0.06	1.06	1.35	1.09	1.09	...
...	...	...	...	...	...	

rapporto di espressione del gene 5 nel paziente 4

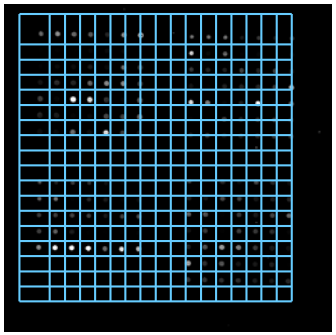
Microarray - cDNA

- Una volta preparato il microarray, gli mRNA relativi a determinati geni sono trattati con dei coloranti fluorescenti (tipicamente Cy3 e Cy5);
- Dei laser applicati al microarray producono una emissione di colori che indicano l'espressione dei mRNA;
- A questo punto viene prodotta una immagine RGB:
  - Rosso per le intensità di Cy5;
  - Verde per le intensità di Cy3;

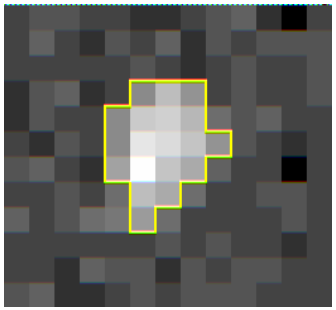


Processamento dell'immagine

- **Addressing/Gridding**: Ad ogni spot è assegnata una coordinata;



- **Segmentation**: Classificazione dei pixel (backgroud/spot);



- **Intensity determination**: Viene misurata l'intensità di ciascuno spot in relazione all'intensità del backgroud;

Microarray – cDNA – Creazione della matrice di espressione

- Microarray rappresentato da una matrice $A=\{a_{ij}\}$ dove:
- $a_{ij} = value_R/value_G$
  - $value_R$= Median (Foreground) - Median (Background);
  - $value_G$= Mean (Foreground) - Mean (Background);

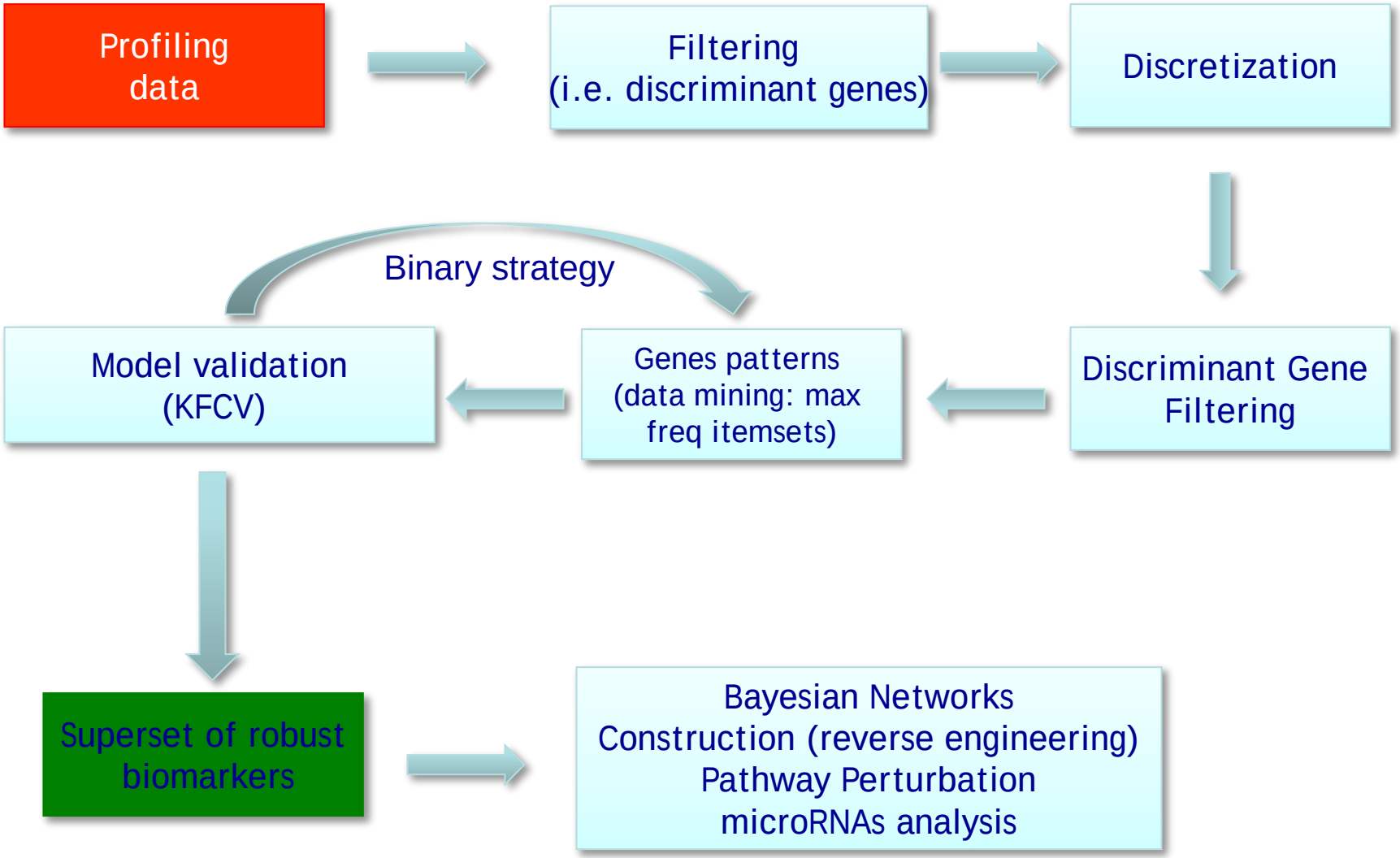
Microarray Data Mining

- Una volta ottenuta la matrice che rappresenta il microarray è possibile applicare tecniche di data mining per:
  - Trovare biomarcatori che permettono di individuare una determinata classe (normal/tumor);
  - Classificare un profilo di espressione genica sconosciuta;
    - Esistono diverse tecniche di data-mining per la classificazione di dati di espressione genica.

Come costruire un classificatore

- INPUT: Matrice di espressione genica (miRNA);
- Ridurre i campioni per classe in modo da equilbrare la generazione dei MFI;
- Filtraggio di geni differentemente espressi nelle varie classi;
- Discretizzazione dei dati;
- Eliminazione dei geni non discriminanti;
- Costruzione di Maximal Frequent Itemset per ciascuna classe;
- Estrazione di regole di associazione utilizzate per classificare samples sconosciuti;
- Validazione del modello tramite KFCV oppure LOOCV;

MIDClass


Profiling data
Filtering (i.e. discriminant genes)
Discretization
Discriminant Gene Filtering
Genes patterns (data mining: max freq itemsets)
Model validation (KFCV)
Binary strategy
Superset of robust biomarkers
Bayesian Networks Construction (reverse engineering)
Pathway Perturbation
microRNAs analysis

MIDClass

- Abbiamo N campioni $\{x_i, y_i\}_{i=1}^{N}$ dove x_i è un vettore M-dimensionale e
- $y_i \in \{0, \dots, k-1\}$ rappresenta la classe di appartenenza.
- I geni (miRNA) sono denotati da $\Phi = \{\Phi_m\}_{m=1}^{M}$ dove $\Phi_m(x_i)$ rappresenta i valori di espressione del campione x per il gene (miRNA) m.

Patients	x_1	x_2	...	...	x_N
Classes	y_1	y_2	...	...	y_N
g_1	$\phi_1(x_1)$	$\phi_1(x_2)$	...	...	$\phi_1(x_N)$
g_2	$\phi_2(x_1)$	$\phi_2(x_2)$	...	...	$\phi_2(x_N)$
...	...	...	...	...	...
...	...	...	...	...	...
g_M	$\phi_M(x_1)$	$\phi_M(x_2)$	...	...	$\phi_M(x_N)$

MIDClass – Discretizzazione

- Al fine di calcolare i Maximal Frequent Itemset dobbiamo prima discretizzare. Ogni valore di espressione discretizzato sarà mappato in un Item rappresentato da un numero intero;
- Differenti metodi di discretizzazione possono influenzare l'accuratezza del metodo
  - supervised-unsupervised, global-local, topdown-bottomup (splitting-merging), etc. etc.;
- Per discretizzare $\Phi = \{\Phi_m\}_{m=1}^{M}$ possiamo usare diversi metodi:
  - Equal Width Interval Bin;
  - Recursive Minimal Entropy Partitioning;
  - Unparametrized Supervised Discretization;
  - Iterative Dicotomizer 3 Discretizer (ID3);

MIDClass – Discretizzazione – Equal Width Interval Bin

- Discretizziamo il range di una variabile continua in B bin (contenitori);
- Dati i livelli d'espressione di un gene Φ_i nel range $\left[v_i^{min}, v_i^{max}\right]$

  l'ampiezza di ciascun bin è impostata a $\delta_i = \frac{v_i^{max} - v_i^{min}}{B}$

  - I limiti di ciascun confine sono impostati a $v_i^{min} + l\delta_i \quad l = 1, \dots, B$
  - Il valore discretizzato del gene i per il paziente j è così assegnato:
    - Sia k il bin in cui cade il valore $\Phi_i(x_j)$
    - Il valore discretizzato sarà $k + (M * i)$

MIDClass – Discretizzazione – Recursive Minimal Entropy e ID3

- Dato l'insieme $S_i = \{\Phi_i(x_1), \dots, \Phi_i(x_N)\}$ di livelli di espressione genica per il gene Φ_i e il confine di partizione T_i, l'entropia della partizione indotta da T_i è

$$E(T_i, S_i) = \frac{|S_i^1|}{|S_i|} Ent(S_i^1) + \frac{|S_i^2|}{|S_i|} Ent(S_i^2)$$

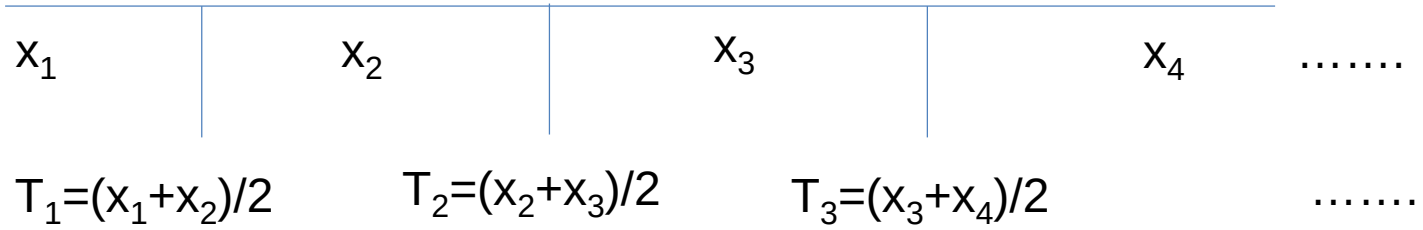
- Il punto di partizione T_i che minimizza l'entropia sarà scelto e il procedimento prosegue ricorsivamente finché non si verifica una condizione di stop:

$$Gain(T_i, S_i) < \frac{log_2(N-1)}{N} + \frac{\Delta(T_i, S_i)}{N}$$

  - dove
    - $Gain(T_i, S_i)$=Ent$(S_i) - E(T_i, S_i)$
    - $\Delta(T_i, S_i) = log_2(3^a - 2) - \left[a \cdot Ent(S_i) - a_1 \cdot Ent(S_i^1) - a_2 \cdot Ent(S_i^2)\right]$
  - con a, a_1, a_2 numero di classi nei set S_i, S_i^1, S_i^2 rispettivamente e N numero di valori discretizzati per il gene S_i.

MIDClass – Discretizzazione – Recursive Minimal Entropy e ID3

- Ordiniamo i valori di $S_i = \{\Phi_i(x_1), \dots, \Phi_i(x_N)\}$
- Siano x_1, x_2,..., x_N i valori ordinati;
- Calcoliamo i punti di mezzo



- Per ciascuno di questi calcoliamo l'entropia e scegliamo come cutting point quello in cui abbiamo minima entropia ottenendo cosi due intervalli;
- Procediamo ricorsivamente sui due intervalli;

- Definiamo:
  - **CutPoint**
    - Valori che delimitano l'intervallo;
  - **Pure Interval**
    - un intervallo i cui valori appartengono alla stessa classe;
  - **Impure Interval**
    - un intervallo i cui valori appartengono a classi miste
  - **Majority class** per un intervallo
    - la classe con più occorrenze nell'intervallo;
  - **Goodness** di un intervallo:
    - $Goodness = \frac{Cardinalità\ della\ classe\ con\ più\ elementi\ nell'intervallo}{1+elementi\ rimanenti\ nell'intervallo}$

- **Algoritmo:**
  - **Calcolo dei cutpoints iniziali**
    - Dividere i valori di un gene in modo da avere il maggior numero di insiemi puri;
      - Problema: si generano molti intervalli!!
  - **Raffinamento**
    - gli intervalli sono ridotti facendo una join tra intervalli adiacenti sulla base della seguente condizione:
      - Se l'intervallo i ha la stessa *majority class* di i+1 e
      - la *goodness* dell'unione è maggiore della media della goodness degli insiemi di partenza.
    - In uno step di raffinamento:
      - La condizione è applicata ad ogni intervallo.
      - La join è eseguita per gli intervalli con goodness migliore
    - Alla fine l'algoritmo produrrà in output l'insieme di intervalli con goodness migliore.

MIDClass – Discriminant Gene Filtering

- Se $\overline{\Phi_m(x_i)}$ è equamente distribuito tra le classi, esso non contribuirà alla classificazione. Per questo motivo calcoliamo

$$s\left(\overline{\Phi_m(x_i)}, k\right) = log_2 \frac{\left|\left\{\overline{\Phi_m(x_l)} : \overline{\Phi_m(x_l)} = \overline{\Phi_m(x_i)}, \overline{\Phi_m(x_l)} \in k, l = 1, \dots, N\right\}\right|}{\left|\left\{\overline{\Phi_m(x_l)} : \overline{\Phi_m(x_l)} = \overline{\Phi_m(x_i)}, \overline{\Phi_m(x_l)} \notin k, l = 1, \dots, N\right\}\right|}$$

- Fissata una soglia **t** diciamo che il gene **m** è discriminante se non esiste una classe **k** per cui $\left|s\left(\overline{\Phi_m(x_i)}, k\right)\right| < t$ o se esiste una classe **k** per cui $s\left(\overline{\Phi_m(x_i)}, k\right) = \infty$
- Se un gene non è discriminante sarà ignorato nella costruzione degli insiemi frequenti.

MIDClass – Discriminant Gene Filtering

- In pratica, dato un valore discretizzato **v** e una classe **k** quello che calcoliamo è:

$$\log_2 \frac{\text{numero di valori discretizzati uguali a v nella classe k}}{\text{numero di valori uguali a v nelle altre classi}}$$

- Tale valore sarà:
  - 0 se v ha la stessa frequenza nelle varie classi;
  - >1 se v compare di più nella classe k rispetto alle altre;
  - <-1 se v compare di più nelle altre classi piuttosto che in k;

MIDClass – Frequent Itemsets

- Uno dei problemi più importanti in Data Mining è trovare regole di associazione:
  - identificare relazioni «interessanti» tra insiemi di oggetti, predicendo inoltre associazioni e correlazioni che possono presentarsi in nuovi dati dello stesso tipo;

- **Market Basket Analysis**: Analizzare i carrelli della spesa per stabilire quali prodotti vengono venduti assieme

- Questo consente di identificare quei prodotti che fanno da traino (e che quindi possono innescare con alta probabilità l'acquisto di altri prodotti).

  - Pannolini → birra; (non vale il viceversa);

- **Marketing**: posizionare in modo opportuno i prodotti negli scaffali;

MIDClass – Frequent Itemsets

- Bisogna affrontare il problema del frequent pattern analysis
  - individuare un pattern (insieme di oggetti) che si presenta frequentemente nei dati
- In generale nel market-basket problem si vogliono estrapolare regole di associazione del tipo:
  - Se un cliente compra x_1, x_2,..., x_k allora probabilmente comprerà anche y
- La probabilità minima che pretendiamo si chiama confidenza

MIDClass – Frequent Itemsets

- Sia $I = \{i_1, i_2, \dots, i_N\}$ un insieme di oggetti e sia **D** un insieme di transazioni su **I**.
- Una transazione è un sottoinsieme di **I**
- Insiemi di oggetti di lunghezza **k** sono chiamati con ***k-itemset***
- Ogni insieme $X \subset I$ ha associato un supporto che indica la frazione di transazioni contenente X
- Un insieme X sarà frequente se ha un supporto superiore a una soglia minima data (*minsupp*)

MIDClass – Frequent Itemsets

- Gli insiemi frequenti ci permettono di costruire delle **regole di associazione**
  - pannolini→ birra

- Le regole di associazione sono da considerarsi come istruzioni IF-THEN

MIDClass – Frequent Itemsets

- Data una regola di associazione X→Y possiamo definire il supporto e la confidenza di tale regola come

$$supp(X \rightarrow Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

$$conf(X \rightarrow Y) = \frac{p(XY)}{p(X)} = p(Y|X)$$

  - dove $X \cap Y$ è l'insieme delle transazioni che contengono sia X che Y e $X \cup Y$ è l'insieme delle transazioni che contengono o X o Y
- In generale il supporto rappresenta la probabilità che una transazione contenga sia X che Y ovvero p(XY)
- La confidenza è una probabilità condizionata che indica quanto robusta è una implicazione (*minconf*)

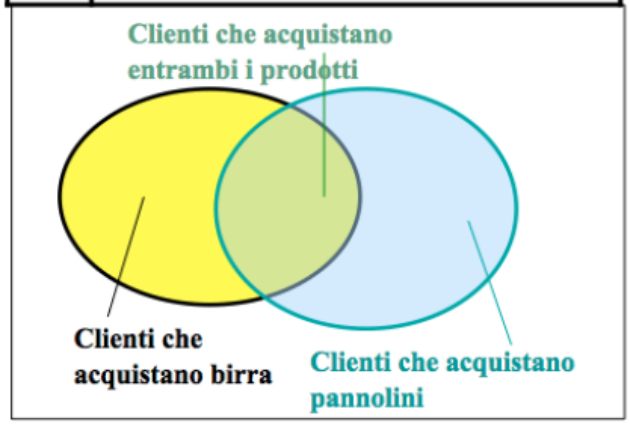
MIDClass – Frequent Itemsets

- Impostiamo *minsupp* = 50% e *minconf*=50%
- I prodotti che superano la soglia minsupp sono:
  - Birra 3;
  - Noccioline 3;
  - Pannolini 4;
  - Uova 3;
  - {birra,pannolini} 3;
- Dall'itemset {birra,pannolini} possiamo tirare fuori le regole di associazione:

$$birra \Rightarrow pannolini \ (60\%, 100\%);$$

$$pannolini \Rightarrow birra \ (60\%, 75\%).$$

Tid	Prodotti acquistati
10	Birra, Noccioline, Pannolini
20	Birra, Caffè, Pannolini
30	Birra, Pannolini, Uova
40	Noccioline, Uova, Latte
50	Noccioline, Caffè, Pannolini, Uova, Latte



Entrambe superano la soglia di minsupp; Tutte le transazioni che contengono birra, contengono anche pannolini (conf=100%); Nel viceversa solo il 75%

MIDClass – A Priori

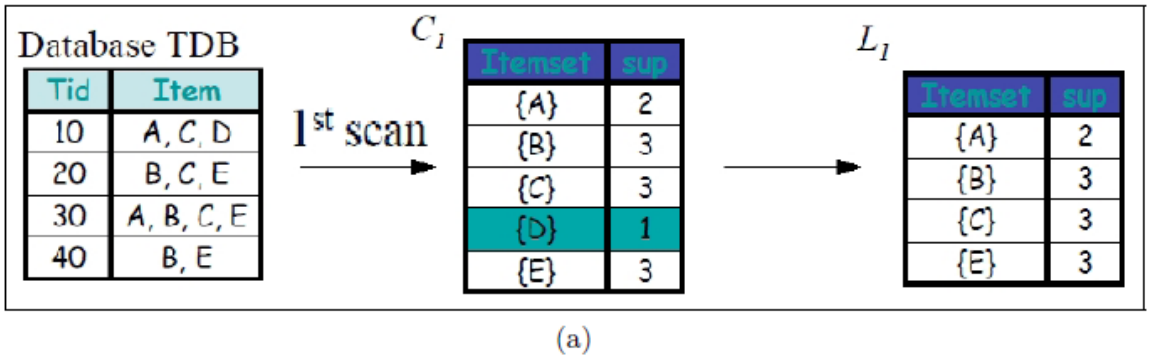
- **Monotonicità:** *Se un insieme S di articoli è frequente, allora anche ogni suo sottoinsieme lo è.*

- L'algoritmo costruisce i singoli insiemi frequenti, a partire da questi costruisce le coppie di insiemi frequenti, dalle coppie costruisce le triple di insiemi frequenti, fino ad arrivare a k-uple di insiemi frequenti in cui non esistono itemset frequenti costituite da k + 1 elementi;

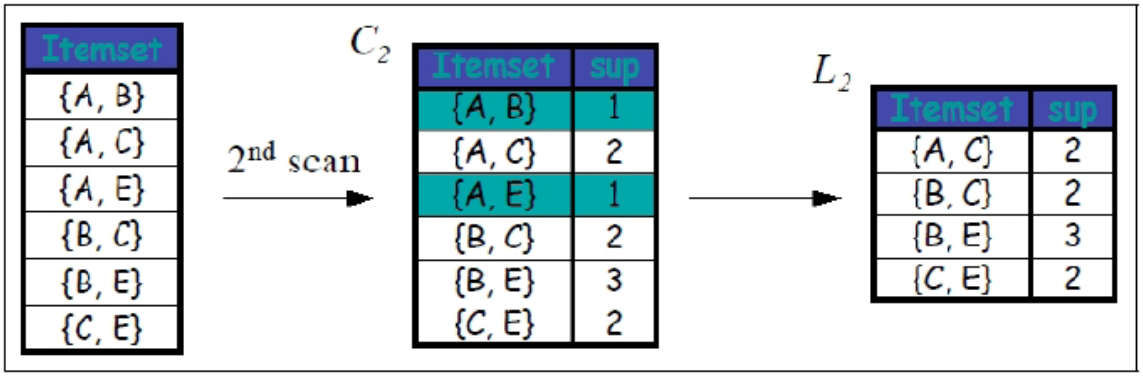
  - $Read(s)$; `/* s soglia di supporto */`
  - $L1 := \{a | a \text{ appare con frequenza} \geq s\}$;
  - $L2 := \{\{a, b\} | a, b \in L1, \{a, b\} \text{ appare con frequenza} \geq s\}$;
  - $L3 := \{\{a, b, c\} | \{a, b\}, \{a, c\}, \{b, c\} \in L2, \{a, b, c\} \text{ appare con frequenza} \geq s\}$;
  - etc ...

MIDClass – A Priori – Esempio

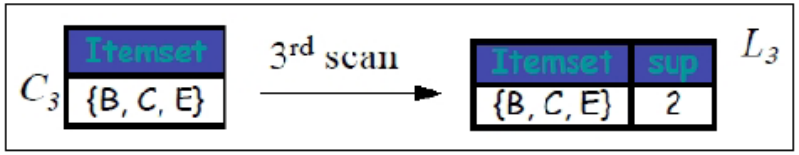
- *Minsupp*=2;
- C_i= candidati;



(a)



(b)



(c)

MIDClass – A Priori – Pseudocodice

$L_1 = \{$ itemset frequenti $\}$;
for $k = 1; L_k \neq \emptyset; k++$ **do**
    C_{k+1} = Candidati generati da L_k;
    foreach *transazione t del database* **do**
        incrementa il conteggio per tutti i candidati in C_{k+1} *che sono contenuti in t*;
    end
    L_{k+1} = candidati in C_{k+1} con supporto minimo s;
end
return $\bigcup_k L_k$;

MIDClass – Insiemi frequenti, chiusi, massimali

- **Insieme frequente**
  - Un Insieme X è frequente se il suo supporto è maggiore di una certa soglia minsup
- **Insieme frequente chiuso**
  - Un insieme frequente è chiuso se non esiste un super-insieme che ha lo stesso suo supporto
- **Insieme frequente massimale**
  - Un insieme frequente è massimale se non esiste un suo super-insieme frequente
- *Gli insiemi massimali sono chiusi, e gli insiemi chiusi sono frequenti*

MIDClass – Insiemi frequenti, chiusi, massimali

- Dato l'insieme di geni discriminanti discretizzato, estraiamo per ogni classe k l'insieme di itemset massimale (MFI);

- Per ogni classe k=0,…,K−1, viene calcolato separatamente l'MFI corrispondente MFI(0),MFI(1),…, MFI(K-1)

- Fissata la classe k MFI(k) sarà composto da un certo numero di itemset frequenti dove il v-esimo sarà della forma mfiv(k)={I1,I2,…,Ij} (ogni item I è un valore discretizzato che indica un gene intervallo);

- mfiv(k) può essere visto come una regola di associazione nella forma:

$$r_v^k : I_i \wedge I_{i+1} \wedge \ldots \wedge I_{j-1} \wedge I_j \Longrightarrow k$$

MIDClass – Insiemi frequenti, chiusi, massimali

- Siano allora date le regole di associazione per ogni classe e un campione sconosciuto (discretizzato):

$$R^k = \{r_1^k, \dots, r_{h_k}^k\} \qquad \bar{x} = \{\overline{x_1}, \dots \overline{x_M}\}$$

- Possiamo allora valutare quante regole sono soddisfatte, anche parzialmente per ogni classe. Fissata la classe k e la regola r_v valutiamo il campione sconosciuto assegnando uno score

$$EVAL(r_v^k, \bar{x}) = \frac{\frac{|r_v^k \cap \bar{x}|}{|r_v^k|} log|r_v^k|}{|R^k|}$$

  - EVAL cerca di tenere in considerazione il numero di item del campione contenute nella regola insieme alla cardinalità della regola (più grande è la cardinalità più alto sarà lo score). Tale score è normalizzato per la cardinalità dell'itemset.
- Lo score finale per una data classe sarà allora dato da:

$$Score(\bar{x}) = \sum_{v=1}^{h_k} EVAL(r_v^k, \bar{x})$$

MIDClass – OVA e AVA

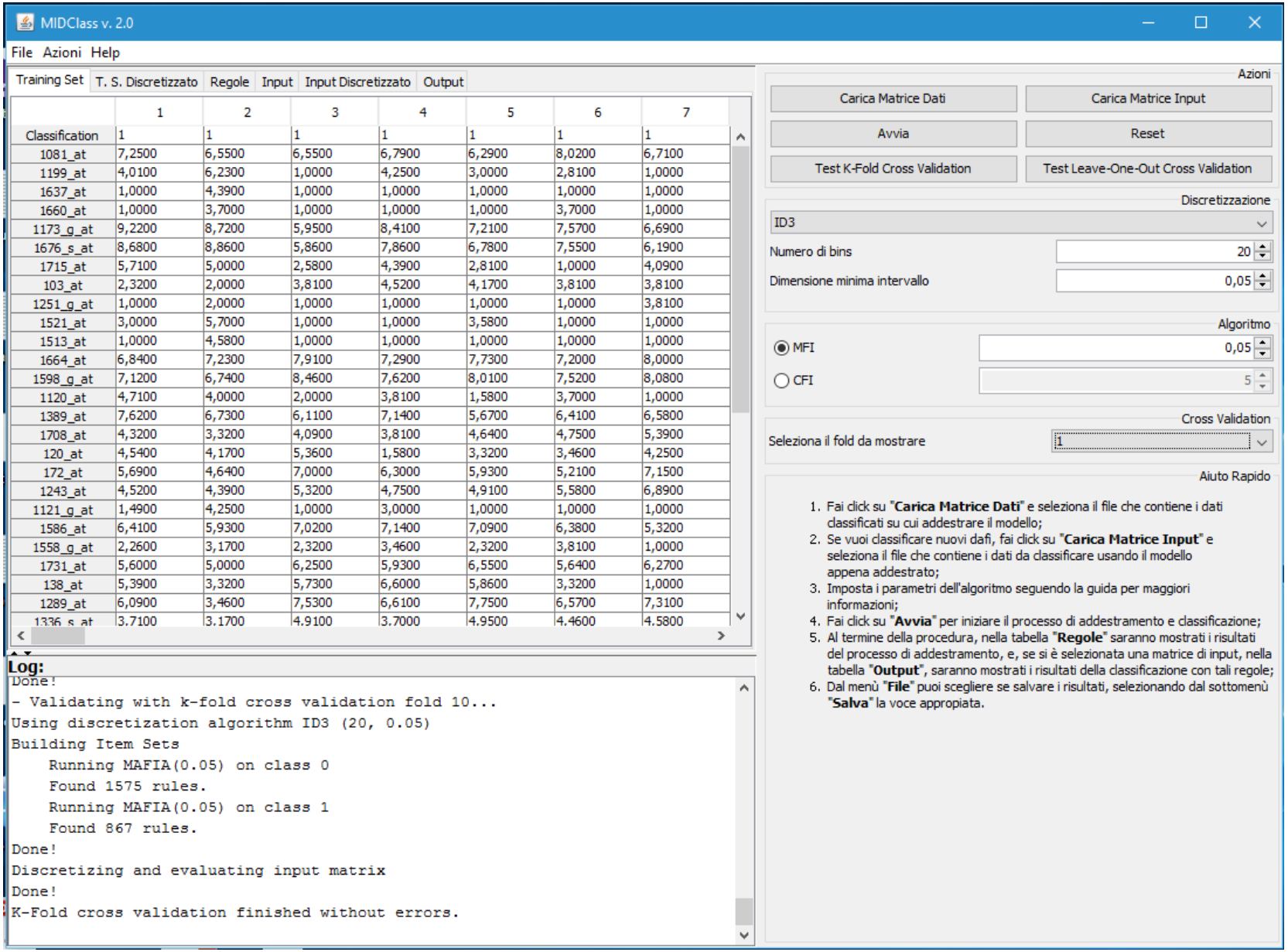
- Quanto illustrato funziona bene se si hanno due classi.
- In presenza di più classi, per massimizzare la qualità dei risultati si possono usare due strategie:
  - OVA: one vs all
    - Date k classi, la strategia OVA costruisce k classificatori uno per ogni classe (gli elementi che non appartengono a quella classe sono tutti assegnati ad una singola classe fittizia) e poi sceglie come risultato quello con massimo supporto.
  - AVA: all vs all
    - Date k classi la strategia AVA consiste nel creare tutti i possibili classificatori binari tra le coppie di k classi. Si sceglie come risultato il consenso tra i classificatori.

MIDClass – Validazione

- k-Fold Cross-Validation:
  - Il training set è diviso in k gruppi distinti;
  - Si usano k-1 gruppi come training set e il gruppo escluso come test input;
  - Processo iterato per ognuna delle k possibili scelte del gruppo tolto dal training-set;
  - Risultato: Media dei risultati;

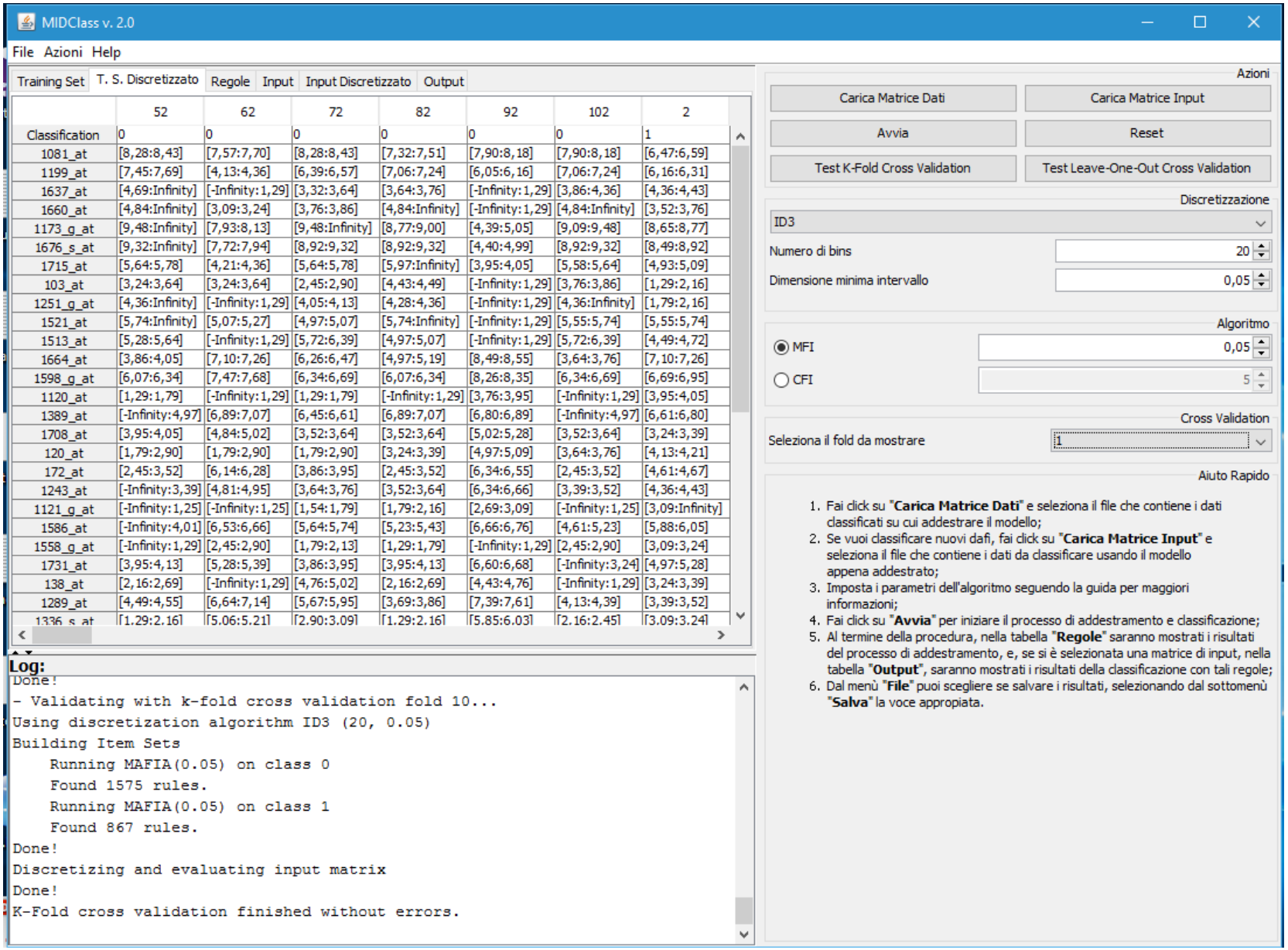
- Leave-one-out Cross-Validation:
  - Si estra un elemento dal set di dati;
  - Il set di dati meno l'elemento verrà usato come training set;
  - L'elemento verrà usato come test input;
  - Processo iterato;

MIDClass – Tool – http://ferrolab.dmi.unict.it/MIDClass.html



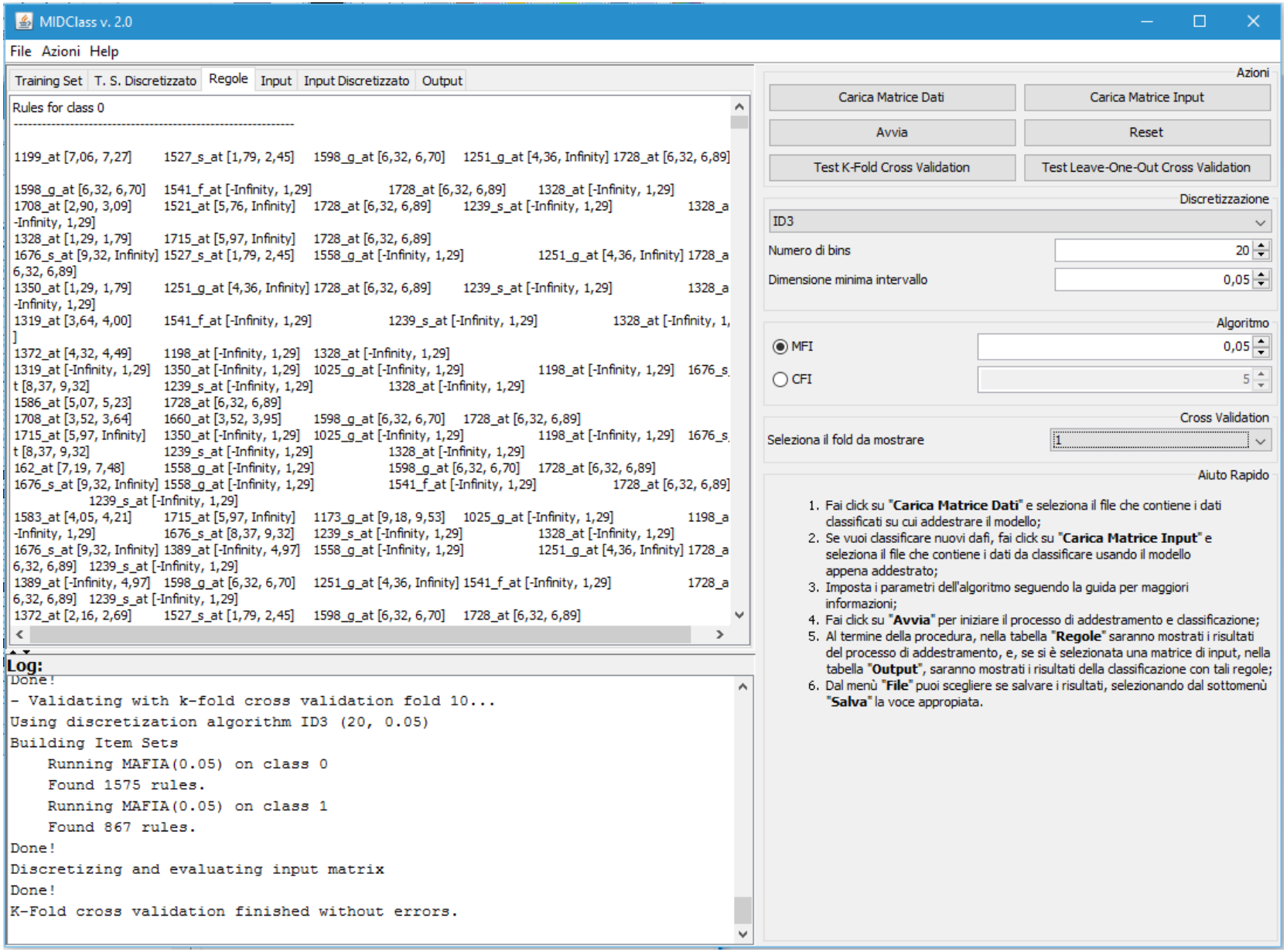
	1	2	3	4	5	6	7
Classification	1	1	1	1	1	1	1
1081_at	7,2500	6,5500	6,5500	6,7900	6,2900	8,0200	6,7100
1199_at	4,0100	6,2300	1,0000	4,2500	3,0000	2,8100	1,0000
1637_at	1,0000	4,3900	1,0000	1,0000	1,0000	1,0000	1,0000
1660_at	1,0000	3,7000	1,0000	1,0000	1,0000	3,7000	1,0000
1173_g_at	9,2200	8,7200	5,9500	8,4100	7,2100	7,5700	6,6900
1676_s_at	8,6800	8,8600	5,8600	7,8600	6,7800	7,5500	6,1900
1715_at	5,7100	5,0000	2,5800	4,3900	2,8100	1,0000	4,0900
103_at	2,3200	2,0000	3,8100	4,5200	4,1700	3,8100	3,8100
1251_g_at	1,0000	2,0000	1,0000	1,0000	1,0000	1,0000	3,8100
1521_at	3,0000	5,7000	1,0000	1,0000	3,5800	1,0000	1,0000
1513_at	1,0000	4,5800	1,0000	1,0000	1,0000	1,0000	1,0000
1664_at	6,8400	7,2300	7,9100	7,2900	7,7300	7,2000	8,0000
1598_g_at	7,1200	6,7400	8,4600	7,6200	8,0100	7,5200	8,0800
1120_at	4,7100	4,0000	2,0000	3,8100	1,5800	3,7000	1,0000
1389_at	7,6200	6,7300	6,1100	7,1400	5,6700	6,4100	6,5800
1708_at	4,3200	3,3200	4,0900	3,8100	4,6400	4,7500	5,3900
120_at	4,5400	4,1700	5,3600	1,5800	3,3200	3,4600	4,2500
172_at	5,6900	4,6400	7,0000	6,3000	5,9300	5,2100	7,1500
1243_at	4,5200	4,3900	5,3200	4,7500	4,9100	5,5800	6,8900
1121_g_at	1,4900	4,2500	1,0000	3,0000	1,0000	1,0000	1,0000
1586_at	6,4100	5,9300	7,0200	7,1400	7,0900	6,3800	5,3200
1558_g_at	2,2600	3,1700	2,3200	3,4600	2,3200	3,8100	1,0000
1731_at	5,6000	5,0000	6,2500	5,9300	6,5500	5,6400	6,2700
138_at	5,3900	3,3200	5,7300	6,6000	5,8600	3,3200	1,0000
1289_at	6,0900	3,4600	7,5300	6,6100	7,7500	6,5700	7,3100
1336_s_at	3,7100	3,1700	4,9100	3,7000	4,9500	4,4600	4,5800

MIDClass – Tool – http://ferrolab.dmi.unict.it/MIDClass.html



	52	62	72	82	92	102	2
Classification	0	0	0	0	0	0	1
1081_at	[8,28:8,43]	[7,57:7,70]	[8,28:8,43]	[7,32:7,51]	[7,90:8,18]	[7,90:8,18]	[6,47:6,59]
1199_at	[7,45:7,69]	[4,13:4,36]	[6,39:6,57]	[7,06:7,24]	[6,05:6,16]	[7,06:7,24]	[6,16:6,31]
1637_at	[4,69:Infinity]	[-Infinity:1,29]	[3,32:3,64]	[3,64:3,76]	[-Infinity:1,29]	[3,86:4,36]	[4,36:4,43]
1660_at	[4,84:Infinity]	[3,09:3,24]	[3,76:3,86]	[4,84:Infinity]	[-Infinity:1,29]	[4,84:Infinity]	[3,52:3,76]
1173_g_at	[9,48:Infinity]	[7,93:8,13]	[9,48:Infinity]	[8,77:9,00]	[4,39:5,05]	[9,09:9,48]	[8,65:8,77]
1676_s_at	[9,32:Infinity]	[7,72:7,94]	[8,92:9,32]	[8,92:9,32]	[4,40:4,99]	[8,92:9,32]	[8,49:8,92]
1715_at	[5,64:5,78]	[4,21:4,36]	[5,64:5,78]	[5,97:Infinity]	[3,95:4,05]	[5,58:5,64]	[4,93:5,09]
103_at	[3,24:3,64]	[3,24:3,64]	[2,45:2,90]	[4,43:4,49]	[-Infinity:1,29]	[3,76:3,86]	[1,29:2,16]
1251_g_at	[4,36:Infinity]	[-Infinity:1,29]	[4,05:4,13]	[4,28:4,36]	[-Infinity:1,29]	[4,36:Infinity]	[1,79:2,16]
1521_at	[5,74:Infinity]	[5,07:5,27]	[4,97:5,07]	[5,74:Infinity]	[-Infinity:1,29]	[5,55:5,74]	[5,55:5,74]
1513_at	[5,28:5,64]	[-Infinity:1,29]	[5,72:6,39]	[4,97:5,07]	[-Infinity:1,29]	[5,72:6,39]	[4,49:4,72]
1664_at	[3,86:4,05]	[7,10:7,26]	[6,26:6,47]	[4,97:5,19]	[8,49:8,55]	[3,64:3,76]	[7,10:7,26]
1598_g_at	[6,07:6,34]	[7,47:7,68]	[6,34:6,69]	[6,07:6,34]	[8,26:8,35]	[6,34:6,69]	[6,69:6,95]
1120_at	[1,29:1,79]	[-Infinity:1,29]	[1,29:1,79]	[-Infinity:1,29]	[3,76:3,95]	[-Infinity:1,29]	[3,95:4,05]
1389_at	[-Infinity:4,97]	[6,89:7,07]	[6,45:6,61]	[6,89:7,07]	[6,80:6,89]	[-Infinity:4,97]	[6,61:6,80]
1708_at	[3,95:4,05]	[4,84:5,02]	[3,52:3,64]	[3,52:3,64]	[5,02:5,28]	[3,52:3,64]	[3,24:3,39]
120_at	[1,79:2,90]	[1,79:2,90]	[1,79:2,90]	[3,24:3,39]	[4,97:5,09]	[3,64:3,76]	[4,13:4,21]
172_at	[2,45:3,52]	[6,14:6,28]	[3,86:3,95]	[2,45:3,52]	[6,34:6,55]	[2,45:3,52]	[4,61:4,67]
1243_at	[-Infinity:3,39]	[4,81:4,95]	[3,64:3,76]	[3,52:3,64]	[6,34:6,66]	[3,39:3,52]	[4,36:4,43]
1121_g_at	[-Infinity:1,25]	[-Infinity:1,25]	[1,54:1,79]	[1,79:2,16]	[2,69:3,09]	[-Infinity:1,25]	[3,09:Infinity]
1586_at	[-Infinity:4,01]	[6,53:6,66]	[5,64:5,74]	[5,23:5,43]	[6,66:6,76]	[4,61:5,23]	[5,88:6,05]
1558_g_at	[-Infinity:1,29]	[2,45:2,90]	[1,79:2,13]	[1,29:1,79]	[-Infinity:1,29]	[2,45:2,90]	[3,09:3,24]
1731_at	[3,95:4,13]	[5,28:5,39]	[3,86:3,95]	[3,95:4,13]	[6,60:6,68]	[-Infinity:3,24]	[4,97:5,28]
138_at	[2,16:2,69]	[-Infinity:1,29]	[4,76:5,02]	[2,16:2,69]	[4,43:4,76]	[-Infinity:1,29]	[3,24:3,39]
1289_at	[4,49:4,55]	[6,64:7,14]	[5,67:5,95]	[3,69:3,86]	[7,39:7,61]	[4,13:4,39]	[3,39:3,52]
1336_s_at	[1,29:2,16]	[5,06:5,21]	[2,90:3,09]	[1,29:2,16]	[5,85:6,03]	[2,16:2,45]	[3,09:3,24]

MIDClass – Tool – http://ferrolab.dmi.unict.it/MIDClass.html



MIDClass – Tool – http://ferrolab.dmi.unict.it/MIDClass.html