Neural Biclustering in Gene Expression Analysis

Neural Biclustering in Gene Expression Analysis

Barbiero Pietro, Ciravegna Gabriele, Elio Piccolo
Politecnico di Torino
Torino, Italy
[pietro.barbiero gabriele.ciravegna]@studenti.polito.it;
elio.piccolo@polito.it

Bertotti Andrea
Università degli studi di Torino, Dipartimento di Oncologia
Candiolo Cancer Institute - FPO, IRCCS, Italy
andrea.bertotti@ircc.it

Cirrincione Giansalvo, Cirrincione Maurizio
University of South Pacific
Suva, Fiji
exin@u-picardie.fr

(Cirrincione Giansalvo contact author)

***Abstract*—Clustering in high dimensional spaces is a very difficult task. Dealing with DNA microarrays is even more difficult because gene subsets are coregulated and coexpressed only under specific conditions. Biclusterng addresses the problem of finding such submanifolds by exploiting both gene and condition (tissue) clustering. The paper proposes a self-organizing neural network, GH EXIN, which builds a hierarchical tree by adapting its architecture to data. It is integrated in a framework in which gene and tissue clustering are alternated and controlled by the quality of the bicluster. Examples of the approach and a biological validation of results are also given.**

Keytopics *— Bioinformatics and Big Data, Gene Pattern Discovery and Identification, Biological Data Mining and Knowledge Discovery, Software Tools and Methods for Computational Biology and Bioinformatics, Microarrays*

Full/Regular Research Paper - CSCI-ISCB

I. BIOLOGICAL INTRODUCTION

Cancer is a heterogeneous disease caused by the sequential accumulation of somatic and stochastic genetic alterations. Each cancer is characterized by a unique set of mutations. Such set of alterations impact both the prognosis and the sensitivity to treatments. Therefore, personalized therapies are required to increase the reliability of prognostic predictions and the efficacy of treatments [1]. In recent years, powerful tools have been developed for biomarker discovery and drug development in oncology, which rely on a technology called Patient-Derived Xenografts (PDXs) [2–4]. PDXs are obtained by propagating surgically derived tumor specimens in immunocompromised mice. Through this, cancer cells remain viable ex-vivo and retain the typical characteristics of different tumors from different patients. Building on these premises, PDXs are extensively exploited to conduct large-scale preclinical analyses for drawing statistically robust correlations between genetic or functional traits and sensitivity to anti-cancer drugs. In this context, we have been collecting metastatic colorectal cancer (mCRC) for the last ten years, and we generated the largest PDX biobank available worldwide in an academic environment. Such resource has been widely characterized at the molecular level [5]-[7] and has been leveraged to reliably anticipate clinical findings [8] with major therapeutic implications. Here it is proposed to exploit available transcriptional data obtained from mCRC PDXs through the Illumina bead array technology [9] to identify and validate novel predictive algorithms to instruct therapeutic decisions.

II. INTRODUCTION

The dataset stemming from the DNA microarrays is composed of the expression of 15396 genes in 146 mCRC murine tissues. Data are preprocessed by the logarithmic normalization and the z-score technique (column statistical scaling) in order to work on the same range and amplifying small distances.

For each tissue two additional quantities are available: a discrete variable describing the cancer response to drugs, as three classes: regressive, stable and worsening cancer; a second continuous variable representing the cancer response to drugs after three weeks, estimated as the difference in size of the tumor.

In sec.III, the computational approach for biclustering is detailed. It is based on the GH EXIN neural network, which is used in pairs for the gene and tissue selection. Sec.IV analyzes the results and sec.V gives an example of biological validation. Sec. VI yields the conclusions.

III. COMPUTATIONAL APPROACH

1. Biclustering

Common requirements in analyzing gene data are the grouping of genes according to their expression under multiple conditions (tissues) and the grouping of conditions based on the expression of a number of genes. These can be achieved by using clustering techniques. However, many activation patterns are common to a group of genes only under specific experimental conditions. Indeed, subsets of genes are coregulated and coexpressed only under certain experimental conditions, but behave almost independently under other conditions. Finding these local expression patterns is the goal of biclustering [10] - [11], also known as two-way clustering or manifold (subspace) clustering, and is the key to uncover unknown genetic pathways. Basically, clustering can be

applied to either the rows or the columns of the data matrix, separately. Biclustering performs clustering in both dimensions, simultaneously. In this work it is achieved by alternating both row and column clustering on projected data derived from the previous steps. It searches for biclusters with constant values, with constant values on rows or columns and with coherent values, respectively. In [11] it is proved that the rank of the corresponding submatrices is less than or equal to three in the noiseless case. Hence, the numerical rank can be used as a figure of merit of the quality of the bicluster. Another index is the *mean squared residue* H_{cc} introduced by Cheng and Church [12], which takes in account the noise in data. It is expressed as:

$$H_{cc} = \frac{\sum_i^{Nr}\sum_j^{Nc} r_{i,j}^2}{Nr * Nc} \qquad (1)$$

where N_c represents the total number of columns of the matrix, N_r represents the total number of rows and $r_{i,j}$ is the residue which is calculated as:

$$r_{i,j} = a_{i,j} - \frac{\sum_k^C a_{i,k}}{C} - \frac{\sum_h^R a_{h,j}}{R} - \frac{\sum_i^R \frac{\sum_j^C a_{i,j}}{C}}{R} \qquad (2)$$

The terms $a_{i,j}$ are the elements of the microarray matrix (rows and columns represent genes and tissues, respectively). C and R are the number of columns and of rows of the bicluster at hand, respectively. The second term is the average value along the ith row, the third term is the average value along the jth column while the last one is the average value within the whole bicluster.

This index decreases as the values in the bicluster tends to be constant, differing for a constant on the rows or differing of a constant on the columns. It goes to zero for the trivial 1x1 bicluster. This fact implies additional controls on the biclusters in order to avoid this drawback.

2. *Neural Framework*

In order to detect biclusters in the gene expression matrix, gene and tissue clustering are alternated. At first a hierarchical clustering is achieved on genes in the tissue space, because it is the lowest dimensional space (in order to avoid the curse of dimensionality, which cannot be avoided if working on the gene space). Then another hierarchical clustering is performed on the tissues in the space of the genes associated to the best leaves produced in the first step (reduced gene space, as an orthogonal projection from the original space). The best leaves of the second step reduce the tissue space for the genes selected after the first step. This corresponds to another orthogonal projection.

Resuming, each cluster (leave) decreases the dimensionality of the problem for the subsequent clustering, whose leaves yield a further dimensionality reduction. Considering that clustering implies a feature selection, this can be viewed as an orthogonal projection of the vectors. Indeed, selecting only some components results in setting the other components to zero. Considering that the basis is canonical, it corresponds to an orthogonal projection into the reduced subspace (cluster).

These two steps, which are illustrated in fig.1, are repeated (alternated projections) until bicluster candidates are identified (see fig.2). There are several ways of estimating a hierarchical tree. Here a novel neural network (GH EXIN) has been devised ad hoc. It will be explained in the next section. The growth of the tree is controlled by the index H_{cc}. However, as seen before, the index tends to zero as the cardinality of the leaves decreases. In order to avoid trivial biclusters, before each clustering step, a check on the minimum number of data in the leaf (Cmin) is performed both in the tissue space and in the gene space (additional check).

The choice of the quality index depends on the goal of the analysis. Other indices can be added (e.g. an index about the shape of the cluster) or replace H_{cc}. However, this choice remains basically heuristic and is an open problem.

Algorithm 1 Neural biclustering

```
biclustering:
GH EXIN on genes
for all leaves do
    if leaf.cardinality ≤ Cmin1 then
        skip leaf
    else
        GH EXIN on the tissues of the leaf (projection)
        for all leaves do
            if projectedLeaf.cardinality ≤ Cmin2 then
                skip leaf
            else
                save projected leaf
                goto biclustering
return
```

Fig.1: Neural biclustering pseudocode

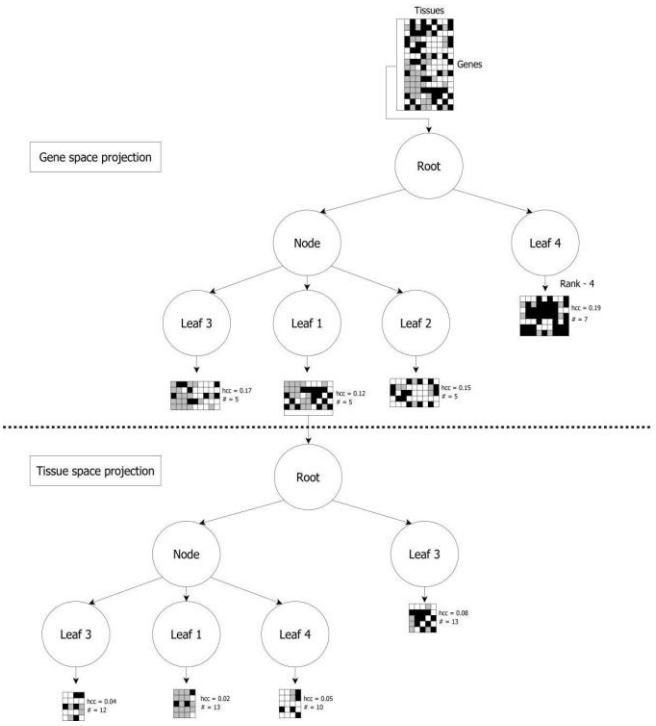

Fig. 2: Neural biclustering scheme: first two steps

3. *The GH EXIN neural network*

The Growing Hierarchical (GH) EXIN neural network is the hierarchical variant of the quantization layer of the GCCA neural network [13]. Each neuron is equipped with a weight vector whose dimensionality is the same of the input space. The architecture is data-driven, in the sense that the number of neurons is determined by the training set by means of node creation or pruning. The weight computation (training) is based on the Soft Competitive Learning (SCL) paradigm, which requires a winner-take-most strategy: at each iteration, the winner and its neighbors change their weights. This law requires the determination of a topology (neighbors) which is achieved by the Competitive Hebbian Learning (CHL) rule, which is used for creating the neuron connections. For the determination of new neurons, thresholds are attached to neurons.

GH EXIN builds a hierarchical (divisive) tree whose vertices correspond to its neurons. At this aim, for each father neuron, the corresponding Voronoi set (set of data represented by the neuron) is estimated and another neural network is trained on this reduced dataset. Its neurons are the sons of the father neuron and determine a subdivision of the father's Voronoi set.

For each leaf, a dedicated neural network is trained. Its initial structure is a *seed*, i.e. a pair of neurons, which are linked by an edge, whose age (attached scalar variable) is set to zero. The initial weight vectors and neuron thresholds are given by heuristics.

For each epoch (presentation in a random way of the whole training set to the network) the basic iteration starts at the presentation of a new data, say x_0. All neurons are ranked according to the Euclidean distances between x_0 and their weights. The neuron with the shortest distance (d_1) is the winner w_1. If its distance is higher than the scalar threshold of the neuron (*novelty test*), a new neuron is created. Otherwise, there is a weight adaptation and a linking phase.

Neuron creation. The weight vector is given by x_0. The new neuron threshold is d_1. It is a *lonely neuron* in the sense that no edges are created.

Adaptation, linking and edge pruning. If a new neuron is not created, it is checked if the winner, whose weight is x_{-1}, and the second winner, whose weight is x_{-2}, are connected by a edge. If there is no edge, these two neurons are linked by an edge (whose age is set to zero) and the same age procedure as in [13] is used as follows. The age of all other links emanating from the *winner* is incremented by one; if a link age is greater than the age_{max} scalar parameter, it is eliminated. Weights are adapted by using SCL [13]: x_{-1} and its direct topological neighbors (there is only one edge between the neuron and the neighbor) are moved towards x_0 by fractions α_1 and α_n (learning rates), respectively, of the vector connecting the weight vector to the datum:

$$\Delta x_{-i} = \alpha_1 (x_0 - x_{-i}) \qquad i = 1 \qquad (3)$$

$$\Delta x_{-i} = \alpha_n (x_0 - x_{-i}) \qquad \text{otherwise} \qquad (4)$$

The thresholds of the winner and second winner are recomputed as the distance to their farthest neighbor. If a lonely neuron wins, or is a second winner, CHL is used for a new edge connecting it.

Neuron pruning and Voronoi re-estimation. At the end of each epoch, if a neuron remains unconnected (no neighbors), two cases are possible.

- If it is not lonely, it is pruned.
- If not, it is pruned, but the associated datum is analyzed, by a new ranking of all the neurons of the network (i.e. also the neurons of the neural networks of the other leaves of the hierarchical tree).

If it is outside the threshold of the winner (i.e. the hypersphere whose center is the winner weight and radius the associated threshold), it is labelled as *outlier* and pruned. If, instead, it is inside, it is assigned to the winner's Voronoi set. If this winner belongs to the network of another leaf, it is re-allocated to the associated Voronoi set.

Stop criterion. Each leaf neural network is controlled by H_{cc}, because it is searching for biclusters (it is estimated by using the data of each Voronoi set). In particular, the training epochs are stopped when the estimated value of this parameter falls below a percentage of the value for the father leaf.

Algorithm 2 GH EXIN

```
1: Root Initialization
2: if it exists at least an heterogenous leaf then
3:     for all the heterogenous leaves do
4:         create 2 children nodes of the leaf
5:         while avg(childrenheterogenity) ≥ Hp * parentheterogenity do
6:             for all the parent data do
7:                 delete previous assignment
8:                 find winner
9:                 find 2nd winner
10:                if winner.T ≤ dist(winner, datum) then
11:                    update winner position
12:                    update nieghbours position (SCL)
13:                    create or update edge winner - 2nd winner (CHL)
14:                    update neighbours ages
15:                    update winner.T
16:                else
17:                    create a new child node coincident with parent datum
18:            for all created leaves (edge pruning) do
19:                for all edge of the node do
20:                    if edge.age ≥agemax then
21:                        delete edge
22:            for all created leaves (node pruning) do
23:                if leaf has no neighbours then
24:                    pOutiler ← pOutlier + leaf.points
25:                    leaf is deleted
26:            for all potential outliers (data pruning) do
27:                find winner
28:                if winner belongs to the same subnet then
29:                    if winner.T ≤ dist(winner, datum) then
30:                        Outlier ← Outlier + leaf.data
31:                    else
32:                        datum is assigned to the winner
33:                else
34:                    datum is assigned to the winner
35:        if there exists more than one connected subtree in the leaf then
36:            for all connected subtrees do vertical growth
```

Fig.3: GH EXIN pseudocode

At the end of each training, Voronoi sets are estimated for each neuron. They are used as training set for the new leaves (new neural networks). This technique builds a vertical growth of the tree. The horizontal growth is generated by the neurons of each network. However, a simultaneous vertical and horizontal growth is possible. At the end of a training, the trees created by the neuron edges are checked. If subtrees (connected trees) are detected, each cluster is considered as a father, by estimating the centroid of the cluster (vertical growth) and the associated neurons as the corresponding sons (horizontal growth).

The whole neural approach requires two groups of user dependent parameters:

1. The GH EXIN parameters, i.e. the two learning rates and the scalar age_{max} for edge pruning; the two rates are constant values for SCL. However, they can be made decreasing in time and automatically scaled by using the Voronoi cardinality (conscience). The last one has to be lowered if more edges (and neurons) have to be pruned. In a sense, it controls, in an indirect way, the leaf cardinality.
2. The biclustering quality indices, i.e. the percentage of H_{cc}, its maximum value and the minimum cardinality of leaves. They control the search and require a deep analysis (out of the scope of the paper).

IV. Analysis of the Results

Each time the neural biclustering algorithm runs the GH EXIN, a tree is built, either in the gene or in the tissue space, as it is shown in fig.4, for the gene clustering in the higher-level leaves. The validity of the leaves is tested and possibly GH EXIN is called again in the corresponding projected space of each leaf. This procedure is recursively repeated until the cardinality of the tissues or the cardinality of the genes of a leaf is under the minimum threshold. At this point the leaf is saved and the algorithm continues by processing the other leaves. The order in which leaves are processed depend on their ranking, based on their H_{cc} value

Low values of H_{cc} associated to an acceptable cardinality do not imply a final bicluster has been detected, above all for the presence of high noise in data. An additional analysis is required, which depends on several considerations. Here, the final leaves are studied from two different points of view: parallel coordinates and singular value decomposition (SVD), for the analysis of the numerical rank of the submatrices associated to the biclusters.

Parallel coordinates are a powerful way of visualizing high-dimensional data. This kind of data visualization was invented during the 19th century and sharpened by Wegman in 1990 [16]. A point in n-dimensional space is represented as a polyline with vertices on equally spaced parallel axes each of one representing a feature; the position of the vertex on the i-th axis corresponds to the i-th coordinate of the point. In this way, all axes can be visualized in the same plot, unlike the classical orthogonal framework which can be visualized only for at most three coordinates. Coherent groups of polylines represent correlated variables, i.e. clusters.

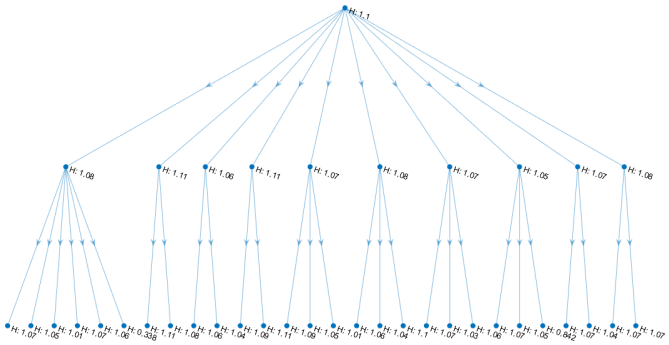

Fig.4: GH EXIN hierarchical tree (first levels)

Fig. 5 shows this kind of plot by visualizing genes as samples (colored polylines) and murine tissues as features (parallel vertical axes) on a leaf of GH EXIN in the gene space, whose characteristics are shown in the top line of the figure.. Blue polylines represent all genes available in the dataset, while red polylines stand for genes collected in the 19th gene cluster. The red grouping of polylines show coherency, which confirms the quality of gene clustering. A similar validation analysis is used after the GH EXIN clustering in the tissue space which is run after projecting the Voronoi set of the 19th gene leaf (cluster).

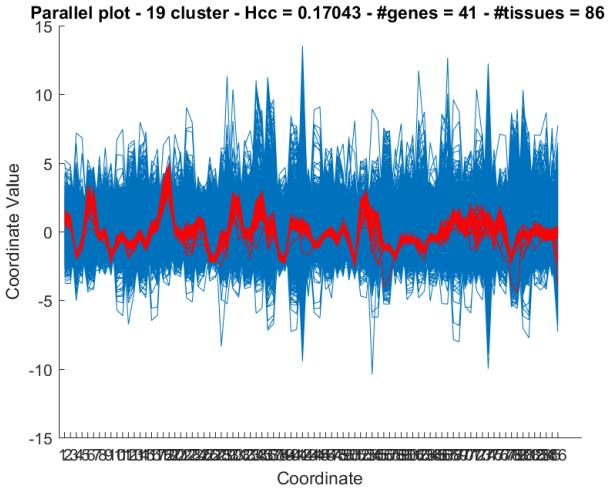

Fig.5: Parallel coordinates of a cluster of gene

Figs. 6 and 7 show two parallel coordinate plots in which vertical axes (here visualized as the corresponding abscissas in the coordinate axis) represent the 41 genes belonging to cluster 19, while polylines stand for murine tissues. In particular, blue polylines represent all the tissues and red ones the tissues grouped in the bicluster. The difference between the two images consists in a different setup of a parameter of the algorithm, Cmin2, which regulates the maximum number of tissues accepted in a bicluster. In the first case a higher value of the parameter is set, in order to find a bigger bicluster. However, both pictures show an excellent bicluster coherency revealing the goodness of GH EXIN as a tool for biclustering.

The biclusters shown in both figures are coherent additive values biclusters in which the values vary both according to the rows (the axes in this case) and according to the columns (the polylines). This can be inferred from the pictures, because a difference is present between two gene expressions on different polylines but along the same axes, but this difference remains stable along the polyline. The same is also valid between two gene expressions on different axes of the same polylines.

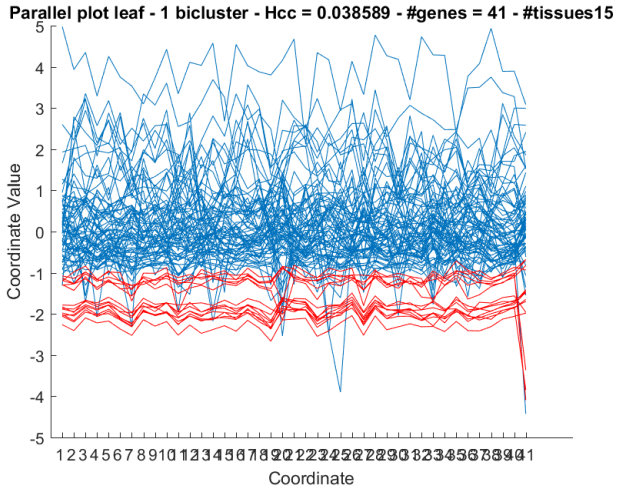

Fig. 6: Parallel coordinates of a bicluster

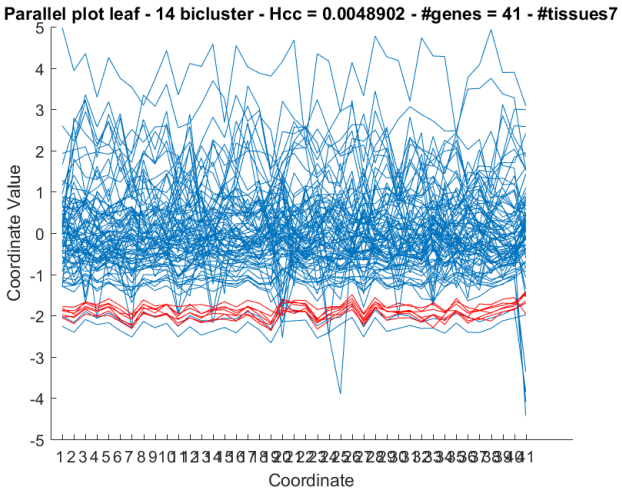

Fig. 7: Parallel coordinates of a smaller bicluster

This visualization tool can be considered as a first validation of the quality of the leaves. A second technique, based on the theory in [11], can be performed analyzing the singular values of the resulting bicluster matrices. According to the theory, in case of noiseless data, biclusters with constant values and with constant values on rows or columns have rank one, while biclusters with coherent values have rank three. The difficulty raises in case of noise, because there are no more zero singular values. Indeed, the size of the last values increases with the level of noise. It then becomes a problem in numerical rank estimation. The SVD of the matrix of the bicluster in fig.6 has the first two singular values (42.5 and 4.5) well separated from the other ones (the third one is equal to 1.5), considering also that the matrix has been scaled in the preprocessing stage). This result represents the sum of two biclusters of rank one, certainly, considering the associated parallel plot, two constant row biclusters. Indeed, fig. 6 shows two clusters (coherent polylines, whose thickness depends on noise level). Hence, it can be deduced that a further clustering (and projection) is needed in order to have a single bicluster. Instead, the SVD of the matrix of the bicluster in fig.7 (see fig.8) has only the first singular value (32.5) well separated from the remaining ones (the second one is equal to 1.1). As also confirmed in fig.7, it represents a constant row bicluster. This result does not require a further analysis.

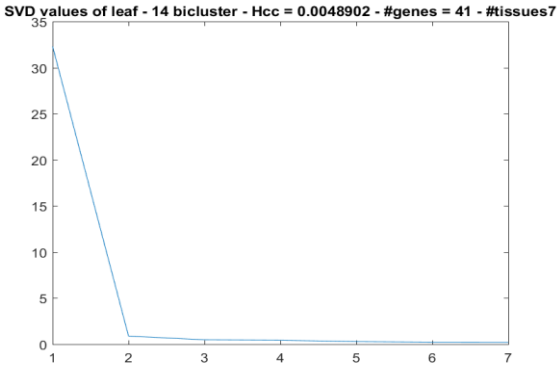

Fig. 8: Singular values for leaf 14

Leaf node	Cardinality		H-biclustering index
Class	*genes*	*tissues*	*H*
1	54	8	0.027
1	58	6	0.059
2	8	8	0.027
2	8	6	0.029
2	71	5	0.031
2	5	5	0.039
3	7	8	0.025
3	7	13	0.025
3	19	6	0.026
3	19	5	0.026
3	41	13	0.028
3	19	8	0.029

TABLE I. BEST LEAVES IN TERMS OF BICLUSTERING QUALITY

In order to better analyze genetic expressions common for different patients, the dataset has been divided into three parts (classes). This division follows the murine tissues response to anti-cancer drugs. At the end. three datasets have been

derived, one for the mice which started recovering after three weeks of treatments, a second one for the mice which had a stable situation and at last one also for the case in which drugs had no effect and the cancer kept growing.

This type of division has been maintained also in the summary table 1, where it has been reported the information about the cardinality of the biclusters, both in the tissue and in the gene space and the value of the H_{cc} index. In the table there are only the best biclusters for each class, ranked according to the class and the H_{cc} index.

V. Biological Validation

As last step, as a biological feedback (validation), the scientific relevance of the selected genes has been taken in account. Among all the biclusters found, the one that grouped the most interesting genes in the cancer field has been the one that also had the lowest H_{cc} index value. Indeed, the 7 genes present in the bicluster are the following:

- "CSAG1", "CSAG3", "CSAG3A", which belong to the same CSAG family. These genes are well known in literature as associated with chondrosarcomas, but they are also present in normal tissues. Furthermore, CSAG3 and CSAG3A are gene coding the "Chondrosarcoma-associated gene 2/3 protein" which is a "drug-resistance related protein, its expression is associated with the chemotherapy resistant and neoplastic phenotype. May also be linked to the malignant phenotype." [14]
- "MAGEA2", "MAGEA3", "MAGEA12", "MAGEA6", which belong to the same MAGEA family. These genes are melanoma antigens which "Reduce p53/TP53 transactivation function" and also "Represses p73/TP73 activity". [15] Both p53 and p73 are tumor suppressor proteins which regulate cell cycle and induct apoptosis.

The relevant issue is that these gene families are not only important by themselves, but this analysis suggests that, at least in the observed condition, they may also coregulate each other. It is also important to notice that this bicluster phenomenon has been observed within the tissues belonging to the third class, the one where tissues unable to respond to drugs are present.

VI. Conclusion

GH EXIN is a powerful tool which is basically a vector quantizer whose architecture depends on a data-driven threshold. It has very few user dependent parameters and creates a hierarchical tree whose leaves depend on the choice of the quality index, as a stopping criterion. The choice of H_{cc} determines its use for searching biclusters. Two GH EXIN networks have been coupled in order to work on projected subspaces of the original microarray matrix. The outputs of this approach are not necessarily the final biclusters, mainly because of noise, but also in case of overlapping subsets. Hence, a post-processing is needed for refinement: it can be a visualization analysis, e.g. parallel coordinate, or an additional biclustering, or a linear or nonlinear projection [17], [18]. Also the algebraic study of the numerical rank of the subarrays can help in extracting more knowledge from data. Here, a biological validation has been presented for illustrating the quality of the results.

Future work will deal with a neural pattern recognition step for the classification of bicluster genes according to their impact on the cancer growth, and a modeling of the corresponding process.

VII. References

[1] de Bono, J. S. & Ashworth, "A. Translating cancer research into targeted therapeutics", Nature 467, 543–549 (2010).

[2] Hidalgo, M. et al., "Patient-derived Xenograft models: An emerging platform for translational cancer research", Cancer Discov. 4, 998–1013 (2014).

[3] Tentler, J. J. et al., "Patient-derived tumour xenografts as models for oncology drug development", Nat. Rev. Clin. Oncol. 9, 338–350 (2012).

[4] Byrne, A. T. et al., "Interrogating open issues in cancer precision medicine with patient derived xenografts", Nat. Rev. Cancer (2017). doi:10.1038/nrc.2016.140

[5] Bertotti, A. et al., "A molecularly annotated platform of patient- derived xenografts ('xenopatients') identifies HER2 as an effective therapeutic target in cetuximab-resistant colorectal cancer", Cancer Discov. 1, 508–523 (2011).

[6] Zanella, E. R. et al., "IGF2 is an actionable target that identifies a distinct subpopulation of colorectal cancer patients with marginal response to anti-EGFR therapies", Science translational medicine, 7, (2015).

[7] Bertotti, A. et al. The genomic landscape of response to EGFR blockade in colorectal cancer. Nature 526, 263–7 (2015).

[8] Sartore-Bianchi, A. et al., "Dual-targeted therapy with trastuzumab and lapatinib in treatment-refractory, KRAS codon 12/13 wild-type, HER2-positive metastatic colorectal cancer (HERACLES): a proof-of-concept, multicentre, open-label, phase 2 trial", Lancet Oncol. 17, 738–746 (2016).

[9] Illumina. Array-based gene expression analysis. Data Sheet Gene Expr. (2011). at http://res.illumina.com/documents/products/datasheets/datasheet_gene_exp_analysis.pdf

[10] Madeira S. and Oliveira A., "Biclustering algorithms for biological data analysis: a survey", IEEE Trans. on Computational Biology and Bioinformatics, vol.1, no.1, pp.24-45, 2004.

[11] P. Barbiero, A. Bertotti, G. Ciravegna, G. Cirrincione and E. Piccolo, "Neural hierarchical biclustering", sent to Bioinformatics in 2017.

[12] Cheng Y. and Church G., "Biclustering of expression data", Proc. Eight International Conference Intelligent Systems for Molecular Biology (ISMB), pp.93-103, 2000.

[13] G. Cirrincione, M. Cirrincione, R. Kumar, E. Pasero and V. Randazzo, "Growing Curvilinear Component Analysis (GCCA) for Stator Fault Detection in Induction Machines", Italian Workshop on Neural Networks (WIRN 2017), June 14- 16, 2017, Vietri sul Mare (SA), Italy.

[14] http://www.genecards.org/cgi-bin/carddisp.pl?gene=CSAG3

[15] http://www.genecards.org/cgi-bin/carddisp.pl?gene=MAGEA2

[16] Wegman, Edward J., "Hyperdimensional data analysis using parallel coordinates", Journal of the American Statistical Association 85.411, pp. 664-675, 1990.

[17] P. Barbiero, A. Bertotti, G. Ciravegna, G. Cirrincione , E. Pasero and E. Piccolo, "Unsupervised Gene Identification in Colorectal Cancer", Italian Workshop on Neural Networks (WIRN 2017), June 14- 16, 2017, Vietri sul Mare (SA), Italy.

[18] P. Barbiero, A. Bertotti, G. Ciravegna, G. Cirrincione , E. Pasero and E. Piccolo, "Supervised Gene Identification in Colorectal Cancer", Italian Workshop on Neural Networks (WIRN 2017), June 14- 16, 2017, Vietri sul Mare (SA), Italy