A COMPARATIVE ANALYSIS OF mRNA EXPRESSION FOR 33 DIFFERENT CANCERS, PART I: THE LUST ALGORITHM

J. B. NATION, GORDON OKIMOTO, TOM WENSKA, ASMEETA ACHARI, TRISTAN HOLMES, JENNA MALIGRO, GORDON OKIMOTO, TOM WENSKA, TAMMY YOSHIOKA, EMORY ZITELLO

Abstract. The Lattice Up-Stream Targeting (LUST) algorithm is a discrete mathematical method for analyzing expression data. It uses a variation of association rules to find groups of genes whose expression is correlated. These sets of genes are called *metagenes*, as they are associated with a common biological process or function. Metagenes can be refined to smaller subsets called *signatures* that represent the entire metagene.

This study uses the LUST algorithm to find metagenes and predictive signatures for the 33 different types of cancer in the TCGA database, based on mRNA expression data. This allows us to identify the metagenes that are common to multiple cancers, and those that occur with only one or two types. Knowing which metagenes are associated with a particular cancer enables us to determine factors that are significant in determining the progress of the disease, which are then candidates for further study.

This first part concentrates on the mathematical background for the LUST algorithm. A second part will present the biological results, analyzing the different metagenes that arise.

1. Introduction

Each type of cancer involves multiple biological processes, some of which are common across cancers, and some of which occur in only one or a few types of cancer. These processes are reflected in mRNA expression data, found in data bases such as TCGA (The Cancer Genome Atlas). In this paper we describe an algorithm to identify some of those processes, and thereby compare different types of cancers. With a slight modification, the same algorithm can be used to generate predictive signatures.

The LUST (Lattice Up-Stream Targeting) algorithm is a discrete method that uses a variation of association rules to find clusters of

Date: August 2, 2019.

genes with similar expression patterns, and within those clusters selects groups of genes that maximize some given objective function. The objective function might represent for example the degree of interaction between the genes (as reflected in the expression data), or some clinical outcome such as survival.

It should be emphasized that we are comparing gene expression for tumors, not tumor vs. normal. We are looking for genes whose expression varies significantly in cancer patients, with the intention of using this information to tailor treatment based on genetic signatures.

Not every metagene produced by the algorithm need be related to the disease. For example, one metagene contains only genes from the male Y chromosome, and just indicates the sex of the patient. But most are related to immune response or cell growth and division in some way.

In fact, we will apply the algorithm twice for each type of cancer in this analysis. In the first pass, LUST is applied to the TCGA mRNA expression data, with an objective function that measures the degree of interaction between the genes. This allows us to identify groups of genes that are part of the same biological process, e.g., immunity or cell division or metabolism, that have sufficient variation across the samples. These we will refer to as *metagenes.* Knowing the metagenes that occur in the gene expression for a particular type of cancer allows us to determine the biological factors that affect the progress of the disease, using tools such as GeneAnalytics or IPA (Ingenuity Pathway Analysis, QIAGEN).

Figure 1 uses a heatmap to illustrate the idea of a metagene, which is a group of genes whose expression levels are coordinated across the samples.

In the second pass, the algorithm is applied to each metagene with an objective function that measures survival. This refines the metagenes to smaller predictive subsets, which we will refer to as *signatures.* By comparing the predictive power of the various signatures for a given cancer, we can determine factors that affect the aggressiveness of the disease.

These signatures may also suggest treatment options. Generally this will require a more detailed analysis of the signatures with regard to clinical data than is done here, but let us consider an elementary application. It is known that immune checkpoint expression itself need not predict response to immunotherapy; see e.g. Meng *et al.* [20] and Section **??**. If however we have a good predictive signature for a given cancer, and the patients in the good prognosis group show significantly differentiated expression for a particular immune checkpoint

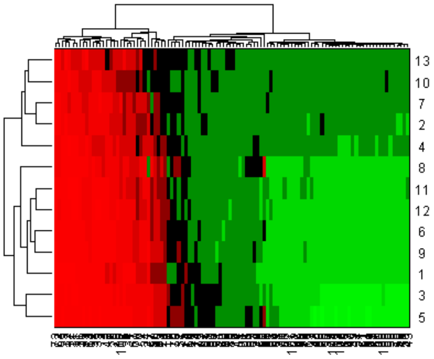


Figure 1. Heatmap of a (small) metagene. The rows represent the genes in the metagene, while the columns represent samples. Most patients show a similar expression level for all the genes in the group, either over-expressing or under-expressing simultaneously.

gene from the patients in the poor prognosis group, then that signature may possibly serve as a biomarker for immunotherapy blockading that checkpoint. On the other hand, if all the good predictive signatures show no differentiated expression on the target, then blockading that checkpoint will likely not significantly affect outcomes. (More accurately, that signature is not a biomarker indicating which patients might benefit from the treatment in question.) This idea is refined in Okimoto *et al.* [26].

Our top-down approach, using the LUST algorithm to find metagenes and signatures, is meant to identify those factors in cancer that warrant further analysis. It is, if you will, looking at the forest rather than the trees, with the objective of indicating what parts of the forest might prove productive. These results indicate candidate focal points for further mathematical, laboratory and clinical investigation, but a high-level overview does not substitute for detailed analysis.

The different metagenes found are presented in our GitHub, and will be discussed in Part II. This first part concerns the mathematical basis for the LUST algorithm, which could be applied to other diseases as well.

This paper presents both the LUST algorithm and the results of applying it to TCGA cancer data. Some readers may wish to skim or skip the mathematical details in Sections 2–6. The LUST programs are written in MATLAB (MathWorks, Natick, MA, USA).

2. Data setup

TCGA mRNA expression and clinical data were obtained from the Broad Institute *via* the Firehose GDAC portal. We used the normalized gene expression files sequenced by Illumina HiSeq, which reports expression levels for 20,531 genes. Samples from surrounding tissue were removed, so that for each patient there was one record, representing tumor tissue. Following a standard protocol, the expression data is log transformed, quantile normalized, and row centered.

The expression data is represented as a $20,531 \times N$ real matrix $\mathbf{E}$, where N is the number of samples. The number of samples varied from 36 for cholangiocarcinoma to 533 for kidney cancer (KIRC), but most were in the range of 150 to 300 samples (see Appendix 3). Clinical data is downloaded for each patient, though for this initial study we used only survival times and censoring information.

The data portal also contains microRNA expression and DNA methylation data. The LUST program can use any combination of these, but so far we have done only a few trial runs with microRNA and methylation. In the long run, these should be included, and perhaps other variables as well.

The next step is to discretize the expression data in $\mathbf{E}$ into a matrix $\mathbf{M}$ with entries $+1$, 0 and -1. The *density* of the discretized expression matrix $\mathbf{M}$ is the fraction of nonzero entries. The density is controlled by an internal threshold variable φ. Entries in the expression matrix that are greater than φ will be replaced by $+1$, representing a high expression level. Entries that are less than $-\varphi$ will be replaced by -1, representing a low expression level. The remaining entries are marked 0, representing a normal expression range. The desired density D is obtained automatically by adjusting φ using the secant method.

Because the data has been centered and quantile normalized, if the density is D, then we expect roughly a fraction D/2 of the entries to be $+1$ and roughly D/2 entries to be -1. For example, if $D = .50$, then $+1$ represents the top quartile of expression, while -1 represents the bottom quartile.

For the purpose of comparison, in the current analysis we used a common value of $D = .50$ for all the cancers. Other values between $D = .40$ and $D = .60$ did not effect significant changes in the results. However, for any particular study, one might vary the density to optimize the results, especially if the number of samples is small.

3. The Lattice Upstream Targeting Algorithm

There are any number of methods available to cluster data sets: hierarchical cluster methods [12, 17, 23], modularity [24, 22], information theoretic clustering [28, 29], association rules [4, 5, 8, 14, 15]. One can also try lattice theoretic techniques, such as Formal Concept Analysis [7, 8, 13], the D-Basis [1, 2, 3, 32], Boolean implications [30, 31, 34], or reference to biological resources such as GeneAnalytics or IPA. *The object is always to group genes into equivalence classes in terms of function.*

One can then think of the regulation of one gene by another as inducing a partial order on these equivalence classes. The LUST algorithm uses these ideas to produce metagenes and signatures. The expression data is first discretized into values of $+1$, 0 and -1 representing over-expression, a normal range, and under-expression, as previously described. The genes are then clustered, using a variation on association rules, into (possibly overlapping) groups with similar expression patterns. Each of these groups can then be refined by restricting them to genes with a common upstream regulator.

Consulting IPA and other genetic resources can be done at any stage of the process, but it is particularly useful for determining the biological relevance of the metagenes and signatures.

3.1. Specifications.

As indicated above, there are generally two parts to our analysis. The LUST algorithm can be run on a large set of expression data to find metagenes, or it can be run on the expression data for a metagene to produce smaller signatures. We will refer to these successive runs as Part I and Part II. In either case, the algorithm will generate many results (groups of genes), which are ranked using an *objective function*. The first run is unsupervised, with an objective function that depends only on the expression data. The second part is supervised, in the sense that the objective function is determined by clinical outcomes (in our case survival). The objective functions currently used are described in Section 3.5.

The input of the LUST algorithm consists of the following.

- Data which can be a combination of gene expression, microRNA expression, methylation, and possibly other variables.
- The values of parameters *density* and *conftol* to adjust the sensitivity of the algorithm.
- For Part II only, clinical data, such as survival time from diagnosis and censorship status, age, stage, treatments. Other clinical data, such as time to recurrence, could be used as well.

The output of the algorithm is:

- Groups of genes (metagenes or signatures), ranked using the objective function.
- For Part II only, a model based on each signature, e.g., a score placing patients into high-risk and low-risk groups.
- For Part II only, Kaplan-Meier survival curves based on the model, or some other measure of the predictive ability of the signature.

Not all the groups obtained need be related to the disease and/or clinical outcomes, and some will be more relevant than others, but the results can be analyzed to identify signatures of interest.

3.2. **Regulation, equivalence and basic groups.** Now let us describe the actual LUST algorithm. Assume the density D has been fixed. Recall that we have the parameter *conftol* at our disposal.

For a row (gene) X, let X^+ denote the set of columns (samples) that are marked $+1$, and let X^- denote the set of columns that are marked -1. We say that X *regulates* Y, and write $X \to Y$, if the following hold:

$$(1)\quad \frac{|X^+ \cap Y^+|}{|X^+|} \geq \mathit{conftol}$$

$$(2)\quad \frac{|X^- \cap Y^-|}{|X^-|} \geq \mathit{conftol}$$

In words, X regulates Y if the conditional probability that a patient over-expresses Y, given that the patient over-expresses X, is at least *conftol*, and likewise for under-expression.

Caveat. The term *regulates* should be interpreted with caution. We write $X \to Y$ to mean that X and Y satisfy the above relations, in this data set. It *might* be that X regulates Y biologically, or X and Y could be regulated by a third factor, while a few such relations may be random coincidence in a large data set. (However, as we will see in Section 6, this is rare for the values of the parameters normally used.) The premise is that if X *does* strongly regulate Y biologically, then it will be reflected as regulation in the data.

Now we say that gene X is *equivalent* to gene Y, and write $X \approx Y$, if $X \to Y$ and $Y \to X$ both hold. This means that X and Y are acting in concert, and heuristically are part of some common process. Note that equivalence is not in general an equivalence relation, as it need not be transitive.

(A variation of the program puts $X \to Y$ when either (1) and (2) hold, or when the expression of X and Y is *negatively* correlated, so that both of the following hold:

(3) $\dfrac{|X^+ \cap Y^-|}{|X^+|} \geq \mathit{conftol}$

(4) $\dfrac{|X^- \cap Y^+|}{|X^-|} \geq \mathit{conftol}$

The results in this paper use only the original version of the algorithm, with (1) and (2).)

3.3. **Forming Groups.**

The LUST algorithm begins by calculating:

(1) for each gene X, a list of all genes Y such that $X \to Y$,
(2) for each gene X, a list of all genes Y such that $X \approx Y$.

The algorithm also produces a list of *nonplayers*. Typically, anywhere from about 50% to 90% of the genes will have the property that neither $X \to Y$ nor $Y \to X$ holds for any other gene Y. These genes will not belong to, nor regulate, any group. While they may play some role in the overall biology, it is not manifest in the data.

Now we form groups of genes as follows. Initially, the groups consist of a gene X and all the genes equivalent to it,

$$F_X = \{Y : Y \approx X\}.$$

These groups may overlap substantially. Overlapping groups are *merged* according to the following scheme: if a larger group contains at least *overlappercent* (a parameter with default value 0.6 for Part I and 0.5 for Part II) of the genes of a smaller group, then the groups are combined. This step is normally performed just once, so as not to extend the transitivity too far. When the merging is complete, the groups may still overlap somewhat, or even one group contain the another. That is to be expected.

Very small groups, with at most some predetermined number genes, can optionally be discarded at this point to speed up the algorithm.

3.4. **Refinement using upstream regulators for signatures.**

This step, the refinement of groups using up-stream regulators, is not essential to the algorithm and could be omitted (despite the name of the algorithm). In practice, we *do* omit it in the first run of the algorithm to find metagenes, but always include it in the second run to find signatures from metagenes. Our experience has been that it gives tighter, more focused signatures than with the step omitted.

Each group G obtained in the first (clustering) step can be refined using an upstream regulator. Thinking of the $\to$ relation as a quasi-order, the idea is to restrict our attention to those genes in a group that have a common regulator. Towards this end, we need a way to select the regulator candidates for each group.

It turns out to be useful to assign each gene a score which measures its overall effectiveness as a regulator of the whole set of genes. This should depend on the number of genes that X regulates, for those genes Y that it regulates the number of times that both are positive or both negative, and the fraction of the time that Y agrees with X. The following score has all those properties:

$$\begin{aligned} s_X &= \frac{1}{N} \cdot \sum_{X \to Y} \left(|X^+ \cap Y^+| + |X^- \cap Y^-| \right) \frac{|X^+ \cap Y^+| + |X^- \cap Y^-|}{|X^+| + |X^-|} \\ &= \frac{1}{N} \cdot \sum_{X \to Y} \frac{(|X^+ \cap Y^+| + |X^- \cap Y^-|)^2}{|X^+| + |X^-|} \end{aligned}$$

where N is the total number of samples (patients or columns).

Now let G be a group obtained in the first step, and choose a value of the parameter *noregs* that determines the number of regulators to be kept for each such group. (The default value is *noregs* $= 5$.) For every gene X not in G, consider the set $G_X = \{X\} \cup \{Y \in G : X \to Y\}$. To every such gene we assign a score which measures how much X regulates G:

$$p_{XG} = \frac{|G_X|}{|G|}(1 + s_X).$$

If *noregs* $= k$ and $X_1, \ldots, X_k$ are the regulators of G with the k highest scores p_{XG}, then the groups $G_{X_1}, \ldots, G_{X_k}$ are kept for further testing. This is done for each of the original groups G.

The intent is that restricting the group to genes with a common upsteam regulator will further isolate the biological processes going on. The groups G_X will generally be smaller than G, and some of them will separate the survival curves as well as or better than G.

Of course, a larger value of *noregs* slows down the algorithm, but it should be large enough that the last few groups obtained are not significant.

3.5. **Objective functions.** The LUST algorithm will generate many groups that are candidates for metagenes or signatures. These candidates are ranked using an *objective function* that we seek to maximize.

There are two basic types of objective function. For Part I, we regard a metagene as a directed graph, with edges determined by the relation

$X \to Y$. The objective function for Part I should be a graph-theoretic measure of the probability of obtaining a set of vertices of that size and density of edges. An alternative would be an information-theoretic measure. *The objective function for metagenes does not depend on clinical outcomes*, but only on the expression data.

For the signatures of Part II, on the other hand, the objective function should depend on clinical outcomes. For this study we used a score based on survival time from date of diagnosis, unadjusted for covariate factors. There are many other clinical objective functions of interest, including survival times adjusted for covariates or disease-free survival (time to recurrence). The factor of interest may also depend on the aggressiveness of the cancer in question.

Part I. Suppose we are given a metagene M with n genes. Regarding M as a directed graph, let $|E|$ be the number of arrow relations (edges) $X \to Y$ between genes (vertices) of M. (Here $X \approx Y$ counts as two directed edges.) A complete directed graph on n vertices would have $n(n-1)$ edges. Hence the *edge density* of M is given by

$$\delta(M) = \frac{|E|}{n(n-1)} \ .$$

The objective function should be increasing in both the size n of the metagene and its edge density. A simple objective function with this property is

$$f(M) = n \cdot \frac{|E|}{n(n-1)} = \frac{|E|}{n-1}$$

and this is what we used for Part I.

There is another way of looking at it. We want to consider the probability that a random set of n genes has $|E|$ or more arrows. Let p be the density of arrows for the entire set of $N = 20,531$ genes, i.e., p is the total number of arrows divided by $N(N-1)$. Then p represents the probability that $X \to Y$ holds for a random pair of genes from the entire set. Hence the expected number of arrows for an n-gene subset is given by $\mu = p \cdot n(n-1)$, with variance $\sigma^2 = p(1-p) \cdot n(n-1)$. So the z-score in this distribution for a set of n genes with $|E|$ edges is given by

$$z = \frac{|E| - pn(n-1)}{\sqrt{2p(1-p)n(n-1)}} \ .$$

However, in general $n \ll N$ and $p \ll 1$, while for a "good" (i.e., dense) metagene $|E| \gg \mu = pn(n-1)$. In that case

$$z \doteq \frac{|E|}{\sqrt{2p(1-p)n(n-1)}}.$$

This is proportional to $\dfrac{|E|}{\sqrt{n(n-1)}}$ which for large n is approximately $\dfrac{|E|}{n-1} = f(M)$. Thus $f(M)$ is a measure of the probability that the edge density of a set of n genes is at least $\delta(M)$.

Part II. Recall that *signatures* are subsets of metagenes, and for Part II we want an objective function that measures the clinical relevance of the signature.

The groups thus formed correspond to signals in the expression data, which may or may not be related to clinical outcomes. In order to determine which groups are associated with survival, for each group G_X we form the submatrix $\mathbf{E}_{G_X}$ of the original (undiscretized) expression matrix whose rows correspond to genes in the group G_X, and columns to patients. We then test whether this group of genes predicts survival on this data set, using the eigen-survival analysis from [25], described in Appendix 1.

Briefly, given a signature G_X, the eigen-survival analysis produces a predictive score for each patient, which is a linear combination of that patient's expression values for the genes in G_X. Once the predictive scores are computed, the top and bottom quartiles are identified. For each of these we calculate the Kaplan-Meier expected survival curves, and measure their separation using the logrank and Cox tests. Each of these produces a p-value for the separation. Let p_1 be the p-value for separating the survival curves using the logrank test, and let p_2 be the p-value for Cox regression. Then the *Fisher score* for the group is given by $F(G_X) = -\log p_1 - \log p_2$. (For convenience, we use the natural logarithm in our calculations, so that if $p_1 = p_2 = .05$, then $F(G_X) = 6$.)

The signatures G_X can then be ranked according to their Fisher scores. This provides a fairly straightforward measure of how well the survival curves are separated, with a larger score indicating more separation.

To study disease-free survival, we need only change the terminal event from *death* to *death or recurrence.* Of course, either the logrank or Cox test could be used separately, and to adjust for covariate factors we would use the latter. Other possible objective functions might be the difference in mean survival time between the good and poor responders, or the difference in 5-year survival rates, but these measures are more sensitive to sample size and censoring effects.

4. Part I: Finding and ranking metagenes

Now let us describe the process to find metagenes for a particular type of cancer. Our first task is to choose the value of the parameter *conftol*. Since the algorithm will be used to compare different types of cancer, we need a uniform protocol. On the other hand, the number of samples varies considerably, and data with a smaller number of samples generally requires a larger value of *conftol* to get comparable results (because they more easily generate arrow relations). Also, the variability of expression differs from cancer to cancer. Still, taking these factors into consideration, the optimum value of *conftol* was between 0.7 and 0.8 for every cancer considered. That being the case, we used *conftol* $= 0.75$ uniformly for our comparisons.

As with density, for any particular study one should try several values of *conftol* and compare the results. Lowering *conftol* increases the sensitivity of the algorithm, and as long as values below 0.5 are avoided, false discoveries are not an issue (see Section 6). Lower values of *conftol* yield larger versions of the same metagenes, plus some additional metagenes that often have little or no clinical significance. This may or may not be desirable, depending on one's point of view: the central contributors to the metagene will persist through a range of values of the parameter.

Now the LUST program is run on the expression matrix $\mathbf{E}$ for the type of cancer being considered. The output consists of the following:

(1) a list of the genes in each of the 32 largest groups $G_1, \dots, G_{32}$,
(2) a table giving the size of the groups $|G_i|$ and their intersections $|G_i \cap G_j|$,
(3) the value $f(G_i)$ of the objective function on each group.

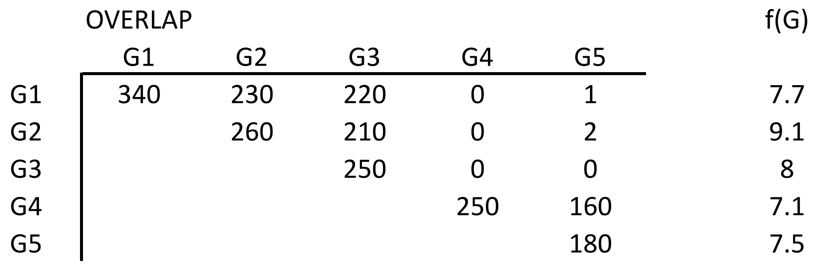
Some comments are in order. The groups in (1) are the groups *after* merging, listed in the order based on their size *before* merging, with their scores on the objective function given in a separate table. This is not crucial but works well, and makes sure that all the large single-gene generated groups G_X are included. The parameter 32 in (1) is clearly arbitrary. The idea is to include enough groups so that the last few are much less significant than the first few, and choosing 20 groups was not enough for that criterion. An argument could be made to include a few more, and the parameter is easily adjusted.

Now we form a *pseudo-equivalence* by setting $G_i \equiv G_j$ if $G_i \cap G_j$ is large. This is again not a transitive relation. Mimicking what was done with merging, we can say that the overlap is "large" if $|G_i \cap G_j|$ is at least some fixed fraction κ of the lesser of $|G_i|$ and $|G_j|$. In fact, this partitioning is easy to do by hand, and has not been built into the program yet, though it should and eventually will be.

From each pseudo-equivalence class, choose a representative G_k that maximizes the objective function $f(G)$. These representatives G_k will be the *metagenes* for this cancer.

Figure 2 illustrates this process for a collection of 5 groups.

In practice, we obtain from 4 to 8 metagenes in this way for each type of cancer. These can be ranked according to their values with the objective function $f(G)$.



	OVERLAP					f(G)
	G1	G2	G3	G4	G5	
G1	340	230	220	0	1	7.7
G2		260	210	0	2	9.1
G3			250	0	0	8
G4				250	160	7.1
G5					180	7.5

Figure 2. Schematic output of the LUST algorithm, Part I. The overlaps show that group G1 contains 340 genes, G2 contains 260 genes, and their intersection $G1 \cap G2$ contains 230 genes, etc. The overlaps indicate that groups G1, G2 and G3 represent the same biological process or pathway; we would choose G2 as the representative metagene since it has the highest score on the objective function $f(G)$. Groups G4 and G5 are from a different process; for this set of candidate metagenes we would choose G5 as the representative based on its score.

5. Part II: Finding and ranking signatures

Now we want to find within each metagene a signature that is predictive of survival, if possible. To achieve this, we run the LUST algorithm again, with different input, objective function, and output.

- The input is the expression matrix $\mathbf{E}_G$ consisting of the rows of the original expression matrix corresponding to the genes in the metagene G, and the clinical data, in this case the survival times and censoring information.

- The objective function is the Fisher score: for a subset $S \subseteq G$, this is $f(S) = -\log(p_1) - \log(p_2)$ where p_1 and p_2 are the p values for logrank and Cox modeling, respectively. (This is one of several options to measure clinical outcomes.)
- The output is a list of the top signatures contained in G, ranked by their Fisher scores. In addition to the list of genes in the signature, we generate the Kaplan-Meier survival curves and a model (see Appendix 1).

There are many factors that affect the predictive quality of a signature, including the aggressiveness of the disease, the probability of early detection, the effectiveness of treatments. The Fisher score also depends on the number of samples, so comparison across cancers must take this into account. A method to standardize the p-scores to account for variations in sample size is given in Appendix 3. One may also want to output other relevant information for the signature; see Section **??**.

Rather than choose a single value of *conftol*, one can run the program for several values, and combine the signatures obtained with their ranks into a pool, and then choose the best Fisher score from all possibilities. For the current study, we used *conftol* = 0.66, 0.70 and 0.74.

6. False discovery rate

The calculation of a good estimate for the false discovery rate is an elementary statistics problem. Fix the density D, and let $p = \frac{D}{2}$. For convenience of notation, let γ denote *conftol.* Consider an $m \times n$ matrix with entries $+1$, -1, 0 assigned from uniform probability distributions of densities p, p, $1-2p$ respectively. We first determine $\text{prob}(X \to Y)$ for a given pair of rows X, Y. Let

$$\begin{aligned} g(n,a,b,p) &= \binom{n}{a,b} p^{a+b}(1-2p)^{n-a-b} \\ h(a,c,p) &= \binom{a}{c} p^{c}(1-p)^{a-c}. \end{aligned}$$

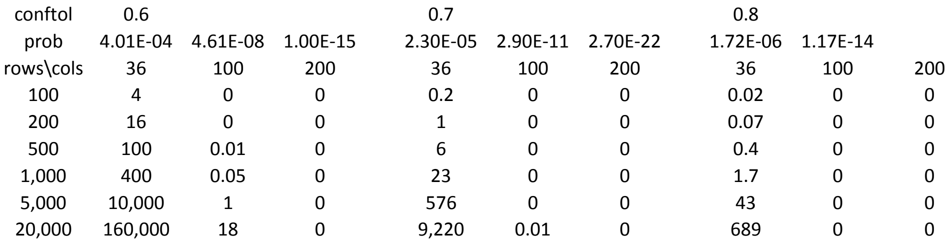
Thus $g(n,a,b,p)$ is the probability that row X will have a entries $+1$ and b of -1. Likewise, $h(a,c,p)$ is the probability that row Y will have c entries of $+1$ in a given set of a columns. Combining these, we see that

$$\begin{aligned}\operatorname{prob}(X \to Y) &= \sum_{1\le a,b\le n} g(n,a,b,p)\left(\sum_{\substack{a\ge c\ge \gamma a\\ b\ge d\ge \gamma b}} h(a,c,p)\cdot h(b,d,p)\right)\\ &= \sum_{1\le a,b\le n} g(n,a,b,p)\left(\sum_{a\ge c\ge \gamma a} h(a,c,p)\right)\left(\sum_{b\ge d\ge \gamma b} h(b,d,p)\right).\end{aligned}$$

The expected number of relations $X \to Y$ over all pairs of the m rows is then given by $E = m(m-1)\cdot \operatorname{prob}(X \to Y)$.

Testing on permuted data matrices confirms that these estimates are very accurate, except for values of *conftol* below 0.4, which is far below the values used in practice.

And the expected number of random arrows is small. The table in Figure 3 gives the estimated number of arrows in a random matrix of various sizes for the range of values of *conftol* that are normally used.



conftol	0.6			0.7			0.8		
prob	4.01E-04	4.61E-08	1.00E-15	2.30E-05	2.90E-11	2.70E-22	1.72E-06	1.17E-14	
rows\cols	36	100	200	36	100	200	36	100	200
100	4	0	0	0.2	0	0	0.02	0	0
200	16	0	0	1	0	0	0.07	0	0
500	100	0.01	0	6	0	0	0.4	0	0
1,000	400	0.05	0	23	0	0	1.7	0	0
5,000	10,000	1	0	576	0	0	43	0	0
20,000	160,000	18	0	9,220	0.01	0	689	0	0

FIGURE 3. Expected number of arrow relations $X \to Y$ for random matrices of various sizes and various *conftol* values. For example, with *conftol* $= 0.7$, in a 200×36 random matrix, the probability of any particular $X \to Y$ is 2.3×10^{-5} and the expected number of arrows is 1.

For Part I, finding metagenes, a typical real expression data matrix of size $20,000 \times 100$ with *conftol* $= 0.7$ (the least value used for Part I), will generate 150,000 to 200,000 arrows. The expected number of random arrow relations is 0.01. When there are more columns, the expected number of random arrows is less.

The worst-case scenario we encounter is with cholangiocarcinoma, for which there are only 36 patients in the TCGA data. For a $20,000 \times 36$ matrix, the expected number of random arrows is 9,220. However, the real data matrix with *conftol* $= 0.7$ yields just over 830,000 arrows, so the fraction of random arrows is still negligible.

For the expression matrices used in Part II, finding signatures, there are even fewer random arrows. We conclude that the *false discovery rate* is effectively zero, for both parts.

Simulations give some idea of the sensitivity of the LUST algorithm. For these simulations, we created a $5,000 \times 120$ signal matrix $\mathbf{S}$ with a step-signal in the first 200 rows consisting of 30 entries of $+1$, 30 entries of -1, and 60 zeros. To this was added a Gaussian noise matrix to form $\mathbf{M} = \mathbf{S} + a\mathbf{N}$, using the variable a to adjust the signal-to-noise ratio (SNR). The LUST algorithm was then applied to $\mathbf{M}$, with the objective of identifying the 200 rows containing the signal.

With *conftol*=0.7 and SNR=$-10\,db$, the algorithm found on average 188 rows out of the 200. The performance deteriorated rapidly with the SNR, so that at SNR=$-12.5\,db$, the algorithm found an average of 4 rows, and for SNR=$-15\,db$ it found none.

Lowering conftol improves the sensitivity. With *conftol*=0.6, the algorithm found 200 at SNR $= -10\,db$, an average of 196 rows at SNR=$-12.5\,db$, and an average of 50 rows at SNR= $-15\,db$.

In the preceding cases, as predicted, there were zero false rows discovered.

With *conftol*=0.5, the algorithm found 200 at SNR $= -10\,db$ and $-12.5\,db$, still with no false discoveries. But at SNR=$-15\,db$, it found an average of only 199 rows in the signal, and there were an average of 4 (out of 4800 possible) false positives.

Of course, we don't know the real nature of biological signals in expression data (except that it is almost surely not a single step-signal embedded in Gaussian noise). But this does tell us that the signals we have found in the expression data, in the form of metagenes and signatures, are quite strong.

7. Results: metagenes

In this section let us consider how one should interpret metagenes. Lists of the genes contained in the metagenes found by the LUST algorithm are given in Appendix 2.

A metagene is collection of genes acting in concert. The probability of this happening for a large number of samples without their being part of some common process (or interacting pathways) is low, as evidenced by the false discovery rate. Moreover, these genes are expressed differentially in the samples. Thus the metagene points to some feature or factor that varies in cancer patients. (It will not in general distinguish tumor from normal.) This may reflect

- aggressiveness of the tumor,

- proliferation,
- patient response to the disease,
- some factor unique to the organ in question (e.g., lipid metabolism in HCC, digestive functon or insulin regulation in pancreatic cancer), or
- none of the above.

The significance of the metagene can be measured by

(1) the objective function $f(M)$ from Section 3, or
(2) how well the signatures obtained from the metagene separate the survival curves (or some other clinical response).

These are rather different measures, which are rather loosely correlated. One function of signatures is to identify the patients that have the factor in question, e.g., impaired immune response or metabolic dysfunction.

If signatures from the metagene separate the Kaplan-Meier survival curves, then there is *something* there that needs to be understood. *Caveat:* None of our current analysis addresses mutations or methylation or microRNA regulation of gene expression. These factors are relevant and should be included in later studies. The LUST algorithm supports multiple data types.

Sometimes a collection of genes that is a single group for some cancers can split into two or more metagenes for others, with the parts being disjoint or nearly so. This is particularly true for metagene A, related to immune regulation, as illustrated for ovarian cancer in Appendix 2. Let us call those parts A1 , A2 and A3. When we look at the groups from Part I that are candidates for metagene A, several options occur.

- Some groups are a mixture of A1, A2 and A3.
- Some groups contain A2 and A3 combined, but separate from A1.
- Sometimes all three parts form distinct groups.

Whether we regard this situation as one metagene with three parts, or three related and slightly overlapping metagenes, is a matter of convenience.

How or whether the split occurs differs for the various cancers.

- The groups from Part I for metagene A are all mixed, with no pronounced split, in bladder and uterine cancer.
- There is a large combined metagene, along with smaller split versions, for ovarian, cervical, prostate and liver CHOL cancers.

- The groups are split into A1 versions and combined A2-A3 versions in pancreatic, colon, liver HCC, kidney KIRC cancers and uterine sarcoma.
- The groups are all heavily type A2 or A2-A3 in kidney KICH, kidney KIRP, rectum and colorectal cancers. (Colon cancer could be put into this group, but it has a small A1 group also.)

In all cases, even when the metagene A is mixed, its signatures tend to come from one part or the other.

Nor do we have a good description of these two parts in terms of function, though it is easy enough to identify certain subsets. For example, A1 contains the SLAM (signaling lymphocytic activation molecule) family genes CD2, CD48, LY9, SH2D1A, SLAMF1, SLAMF6 and SLAMF7. Part A2 contains genes from the tyrosine kinase pathway: BTK, LCP2, PTPRC, SIGLEC7, SIGLEC9, SIGLEC10 and (in most versions) VAV1, as well as several genes from the Fc gamma receptor (FCGR) and G protein receptor (GPR) families. This is an area that warrants further investigation; see the analysis of immune response in HCC by Sia *et al.* [33].

Metagene R for stomach cancer also shows a distinct split into two parts, with part R2 being the stomach cancer version of metagene C.

It can be instructive to run the LUST algorithm on subsets of patients meeting a certain criterion. We have only just begun doing this systematically, using stage as a measure of early and late disease.

For example, we divided liver cancer (HCC) patients into two groups:

(1) stage 1 tumors weighing at most 500 grams,
(2) stage 2 or stage 3 tumors.

Patients with large stage 1 tumors or stage 4 were omitted. Our data represents expression levels for each patient at the time of resection, but gene expression may well change as the disease progresses. Looking at different stages amounts to using *stage* as a surrogate for *time*.

The first distinction was simply in the number of genes that were significantly differentially expressed between long-term and short-term survivors in each group. For the Stage 1 group, 67 of the 20,531 genes in our data set were differentially expressed with a p-value less than .001. Since there are 157 patients in the Stage 1 group and 193 patients in Stage 2–3, we convert $p \leq .001$ to the corresponding level for 193 samples (as discussed in Appendix 3), which is $p \leq .000264$. For the Stage 2–3 group, 1143 of the 20,531 genes were differentially expressed with a p-value less than .000264.

The two groups, Stage 1 and Stage 2–3 also had a noticeably different ranking of metagenes. Metagene A (regulating immune response) was most prominent for Stage 1 tumors, and gave the best prognostic signatures. Metagene B (cell division and mitosis) was a distant second. For Stage 2–3 tumors, the results were reversed: Metagene B was the more prominent and predictive signature. Based on other work by our group, we suspect that metabolic function becomes increasingly disregulated as the disease progresses [26].

Early diagnosis of HCC is perhaps the primary factor for patient outcome with liver cancer [18]. However,he survival distribution for stage 1 HCC patients is bimodal: roughly 20% of these patients die within the first two years after diagnosis, another 10% during the next two years, while 70% live at least four years, often much longer. A 44-gene signature based on metagene A identifies the stage 1 short-term survivors fairly accurately [21]. If the poor prognosis group could be recognized, then these patients would be candidates for alternate treatment.

On the other hand, when we analyzed stage 1 vs. stage 2–3 cervical cancer, or stage 2 vs. stage 4 bladder cancer, there were no appreciable differences in the metagenes or signatures. These disparities in behavior beg explanation.

8. Conclusion

The LUST algorithm uses a variation of association rules to cluster genes that have similar expression patterns. We have applied the algorithm to mRNA expression data from TCGA for sixteen different types of cancer.

For each type, the algorithm is applied twice. The first time the algorithm is applied to the entire expression matrix, and uses a graph-theoretic objective function to rank the groups obtained (Part I). This pass identifies and ranks a small set of *metagenes* associated with the cancer. The second time the algorithm is applied to the expression matrix for each metagene (Part II). This pass is supervised by survival, and uses the Fisher score to rank the results. This pass identifies small predictive *signatures* contained in the metagene.

Some metagenes recur consistently across different cancers, while others are particular to a cancer or set of cancers. This was to be expected; the point is that the algorithm can find them mathematically. These metagenes represent factors that differentiate the nature of the tumor, either in terms of aggressiveness or patient response. (Other methods can be used to distinguish tumor profiles from normal ones.) Recognizing the biological processes corresponding to each metagene can help us understand the functioning of the tumor.

The signatures derived from the metagenes indicate the effect of those metagenes on the progress of the disease, such as survival time. In some cases, signatures could potentially be used as biomarkers to guide treatment.

Analysis of mRNA expression data paints a broad picture of tumor biology. The LUST algorithm can be applied to combinations of mRNA expression, microRNA, methylation, or other continuous data. Other methods may be more appropriate for binary data, such as mutations. For studies aimed at finding the mutations driving oncogenesis and proliferation across multiple cancers, see [6, 9, 10, 35].

9. Appendix 1: Eigen-survival analysis

We want to measure the expression level for a given signature in a way that can be tied to survival. This measure should provide a score which will divide the patients into high-risk and low-risk groups, or place them on a scale according to risk. There are several options, but we use the following eigen-signature scheme [25].

Let $\mathbf{M}$ be the expression matrix for the signature, with rows corresponding to the variables in the signature and columns to patients. Write the singular value decomposition of $\mathbf{M}$ in the outer product form

$$\mathbf{M} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T.$$

Each right singular vector $\mathbf{v}_i$ is then tested for association with the survival data using Kaplan-Meier (KM) analysis with log-rank testing, and Cox regression modeling with age as a covariate. To accomplish this, we interpret the components of $\mathbf{v}_i$ as “predictive scores” for each patient and sort the patients by this score to identify those that fell in the top and bottom quartiles of scores. A given v_i was called significant if and only if differences in survival between patients in top and bottom quartiles based on $\mathbf{v}_i$ are significant in both the KM and Cox regression models with a p-value of 0.05 or less. Given that at least one such $\mathbf{v}_i$ exists, we define

$$\begin{aligned}
&B = \{j \mid \mathbf{v}_j \text{ is significant in both the KM and Cox models}\},\\
&U = \{\mathbf{u}_j \mid j \in B\},\\
&V = \{\mathbf{v}_j \mid j \in B\}.
\end{aligned}$$

Then we take the *predictive vector* to be a linear combination of the vectors in V,

$$\mathbf{w} = \sum_{j \in B} \operatorname{sign}(j) \sigma_j \mathbf{v}_j$$

where $\operatorname{sign}(j) = \pm 1$ was chosen so that $\mathbf{w}$ is significantly associated with survival in both the KM and Cox regression models. (Note that the singular value decomposition is determined only up to a choice of sign. In the linear combination for the predictive vector, the signs should be chosen so that they associate with survival in the same way, e.g., the top quartile should associate with longer survival for each j.) To extract the predictive vector from the SVD, we compute

$$\mathbf{w}^T = \sum_{j \in B} \operatorname{sign}(j) \mathbf{u}_j^T \mathbf{M}$$

or equivalently

$$\mathbf{w} = \sum_{j \in B} \operatorname{sign}(j) \mathbf{M}^T \mathbf{u}_j .$$

In other words, the j-th entry of the predictive vector vector $\mathbf{w}$, associated with survival of the j-th patient, is the dot product of the j-th column of $\mathbf{M}$, which consists of the measurements of the variables in the signature for that patient, with the weighting vector $\mathbf{z} = \sum_{j \in B} \operatorname{sign}(j) \mathbf{u}_j$. This provides a "score" for each sample, *viz.*, for patient j the score is the j-th entry of $\mathbf{w}$.

To compute a predictive score for a set of new patients not included in the original samples, let $\mathbf{M}'$ be a matrix with columns that represent expression values of the signature variables for those patients. Then form the predictive vector

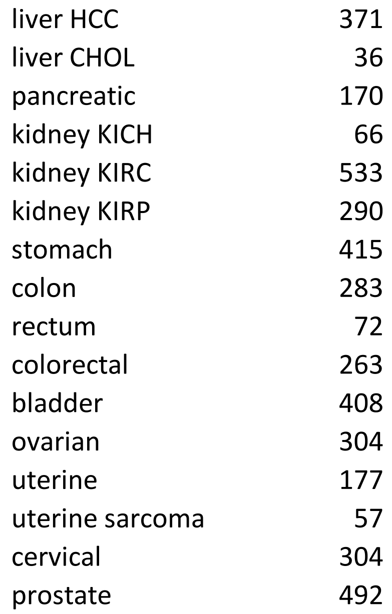
$$\mathbf{w}' = \sum_{j \in B} \operatorname{sign}(j) (\mathbf{M}')^T \mathbf{u}_j$$

which transforms the columns of the matrix $\mathbf{M}'$ into a vector of predictive scores for these patients. If KM and Cox regression analysis indicates that $\mathbf{w}'$ is associated with overall survival, then we conclude that the signature is a robust predictor of survival that is capable of generalizing to new patients that were unseen during discovery.

10. Appendix 2: Adjusting p-values for sample size

Figure 4 gives the number of samples used in this study for each type if cancer.

We would like to compare signatures across cancers, but the p-values for the Kaplan-Meier curves depend on the number of samples. For this purpose, we convert the p-values to the value that would be obtained with 200 samples for a distribution with the same means and standard deviations for the upper and lower quartiles. It is not hard to write a program to convert to standardized p-values. Table 1 gives conversions to this standard for a range of values.



liver HCC	371
liver CHOL	36
pancreatic	170
kidney KICH	66
kidney KIRC	533
kidney KIRP	290
stomach	415
colon	283
rectum	72
colorectal	263
bladder	408
ovarian	304
uterine	177
uterine sarcoma	57
cervical	304
prostate	492

Figure 4. Number of samples in TCGA mRNA expression data for each type of cancer.

Based on the table, one can see that a good estimate can be obtained using the formula

$$p_{200} = q_n s_n^{-\log_{10} p} \tag{\dagger}$$

with the values of q_n and s_n provided in Table 2.

References

[1] Adaricheva, K., Nation, J.: Discovery of the D-basis in binary tables based on hypergraph dualization. Theoretical Computer Science 458(B), 307–315 (2017).

[2] Adaricheva, K., Nation, J., Rand, R.: Ordered direct implicational basis of a finite closure system. Discrete Applied Math. 161, 707–723 (2013).

[3] Adaricheva, K., Nation, J., Okimoto, G., Adarichev, V., Amanbekkyzy, A., Sarkar, S., Sailanbayev, A., Seidalin, N., Alibek, K.: Measuring the implications of the D-basis in analysis of data in biomedical studies. Proceedings of ICFCA-15, Nerja, Spain, Springer, 2015, 39–57.

[4] Agrawal, R., Amieliński, T., Swami, A.: Mining association rules between sets of items in large databases. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, SIGMOD '93, p. 207.

[5] Agrawal R., Srikant, R.: Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), Santiago, Chile, September 1994, 487–499.

$n \diagdown p$	$1e-2$	$1e-3$	$1e-4$	$1e-5$	$1e-6$	$1e-7$
50	$2.6e-7$	$4.7e-11$	$7.2e-15$	$1.0e-18$	$1.3e-22$	$1.7e-26$
100	$2.7e-4$	$3.3e-6$	$3.8e-8$	$4.2e-10$	$4.6e-12$	$5.0e-14$
200	$1e-2$	$1e-3$	$1e-4$	$1e-5$	$1e-6$	$1e-7$
300	$3.5e-2$	$7.2e-3$	$1.5e-3$	$3.1e-4$	$6.5e-5$	$1.4e-5$
400	$6.9e-2$	$2.0e-2$	$5.9e-3$	$1.8e-3$	$5.4e-4$	$1.7e-4$
500	$1.0e-1$	$3.7e-2$	$1.4e-2$	$5.2e-3$	$2.0e-3$	$7.7e-4$

Table 1. Table for converting p-values for survival curves (using quartiles) from n samples to a standard 200-sample value. For example, if the separation with 50 samples has a p-value of .01, then 200 samples with the same means and standard deviations for both low and high quartiles would have a p-value approximately 2×10^{-7}. On the other hand, a p-value of .01 on 500 samples would be equivalent to 0.1 on 200 samples. By converting to a standard size, we can compare separation p-values (and by extension Fisher scores) for signatures from data sets with different numbers of samples.

n	q_n	s_n
50	1.28	$1.54e-4$
100	2.15	$1.15e-2$
200	1	0.1
300	0.820	0.206
400	0.760	0.297
500	0.735	0.371

Table 2. Parameter values for formula (†) to convert p-values to a standardized value for 200 samples.

[6] Bailey, M.H., Tokheim, C., Porta-Pardo, E., Ding, L., et al.: Comprehensive characterization of cancer driver genes and mutations. Cell 173, 371–385 (2018), doi: 10.1016/j.cell.2018.02.060.

[7] Carpineto, C., Romano, G.: GALOIS: An order-theoretic approach to conceptual clustering. Proceedings of the 10th International Conference on Machine Learning, Amherst, 33–40 (1993).

[8] Carpineto, C., Romano, G.: Concept Data Analysis: Theory and Applications, Wiley, West Sussex (2004).

[9] Creixell, P., Reimand, J., Haider, S., Wu, G., Shibata, T., Vazquez, M., Mustonen, V., Gonzalez-Perez, A., Pearson, J., Sander, C., Raphael, B.J., Marks, D.S., Quellette, B.F.F., Valencia, A., Bader, G.D., Boutros, P.C., Stuart, J.M., Linding, R., Lopez-Bigas, N., Stein, L.D.: Pathway and network analysis of cancer genomes. Nat. Methods 12, 615–621 (2017), doi: 10.1038/nmeth.3440.

[10] Ding, L., Wendl, M.C., McMichael, J.F., Raphael, B.J.: Expanding the computational toolbox for mining cancer genomes. Nat. Rev. Genet. 2014 Aug 15, 556-570, doi: 10.1038/nrg3767.

[11] Dobruck, J., Daneshmand, S., Fisch, M., Lotan, Y., Noon, A.P., Resnick, M.J., Shariat, S.F., Ziotta, A.R., Borjian, S.A.: Gender and bladder cancer: a collaborative review of etiology, biology, and outcomes. Eur. Urol. 69(2), 300–310 (2016), doi: 10/1016/j.eururo.2015.08.037.

[12] Fortunado,S.: Community detection in graphs. Physics Reports 486, 75–174 (2010).

[13] Ganter, B., Wille, R.: Formal Concept Analysis - Mathematical Foundations. Springer, New York (1999).

[14] Hájek, P., Havel, I., Chytel, M.: The GUHA method of automatic hypothesis determination. Computing 1, 293–308 (1966).

[15] Hájek, P., Feglar, T., Rauch, J., Coufal, D.: The GUHA Method, Data Preprocessing and Mining, Dababase Support for Data Mining Operations. Springer, New York (2004).

[16] Kim, S.E., Paik, H.Y., Yoon, H., Lee, J.E., Kim, N., Sung, M.K.: Sex- and gender-specific disparities in colorectal risk. World J. Gastroenterol. 21(17), 5167–5175 (2015), doi: 10.3748/wjg.v21.i17.5167.

[17] Lancichinetti, A., Fortunado, S.: Community detection algorithms: a comparative analysis. Physical Review E, 80(5):056117 (2009).

[18] Llovet, J., Zucman-Rossi, J., Pikarsky, E., Sangro, B., Schwarz, M., Sherman, M., Gores, G.: Hepatocellular carcinoma. Nat. Rev. Dis. Primers (2016 Apr 14), doi: 10.1038/nrdp.2016.18.

[19] Lucca, I..F, Klatte, T., Fajkovic, H., de Martino, M., Shariat, S.E.: Gender differences in ncidence and outcomes of urothelial and kidney cancer. Nature Reviews Urology 12, 585—592 (2015), doi:10.1038/nrurol.2015.232

[20] Meng, X., Huang, Z., Teng, F., Xing, L., Yu, J.: Predictive biomarkers in PD-1/PD-L1 checkpoint blockade immunotherapy. Cancer treatment reviews 41, 868–876 (2015); doi: 10.1016/h.ctrv.2015.11.001.

[21] Nation, J.: Identifying stage 1 hepatocellular carcinoma patients with poor prognosis. Draft available at math.hawaii.edu/∼jb/.

[22] Newman, M.E.J.: Modularity and community structure in networks. Proc. of the National Acad. of Sciences 103, 8577–8582 (2006).

[23] Newman, M.E.J.: Networks: an introduction. Oxford University Press, Oxford (2010).

[24] Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E, 69(2):026113 (2004).

[25] Okimoto, G., Zeinalzadeh, A., Wenska, T., Loomis, M., Nation, J., Fabre, T., Tiirikainen, M., Hernandez, B., Wong, L., Kwee, S.: The joint analysis of multiple, high-dimensional data types using sparse matrix factorizations of

rank-1 with applications to ovarian and liver cancer. BioData Mining, 2016, 9:24, doi: 10.1186/s13040-016-0103-7.

[26] Okimoto, G., Wenska, T., Zitello, E.: Warburgian tumors in hepatocellular carcinoma and cholangiocarcinoma. Preprint.

[27] Rampersaud, E.N., Klatte, T., Bass, G., Patard, J.J., Bensaleh, K., Böhm, M., Allhoff, E.P., Cindolo, L., De La Taille, A., Mejean, A., Soulie, M., Bellec, L., Christophe Bernard, J., Pfister, C., Colombel, M., Belldegrun, A.S., Pantuck, A.J., George, D.: The effect of gender and age on kidney cancer survival: younger age is an independent prognostic factor in women with renal cell carcinoma. Urol Oncol. 32(1) 30 (2014), doi: 10.1016/jurolonc.2012.10.012.

[28] Rosvall, M., Bergstrom, C.T.: An information-theoretic framework for resolving community structure in complex networks. Proc. of the National Acad. of Sciences 104, 7327–7331 (2007).

[29] Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. Proc. of the National Acad. of Sciences 105, 1118–1123 (2008).

[30] Sahoo, D., Dill, D.L., Gentles, A.J., Robert Tibshirani, R., Plevritis, S.K.: Boolean implication networks derived from large scale, whole genome microarray datasets. Genome Biology 9:R157 (2008); doi 10.1186/gb-2008-9-10-r157.

[31] Sahoo, D., Seita, J., Bhattacharya, D., Inlay, M., Weissman, I., Plevritis, S.K., Dill, D.L.: MiDReG: A method of mining developmentally regulated genes using boolean implications. Proc. Nat. Acad. Sci. 107(13): 5732–5737 (2010); doi 10.1073/pnas.0913635107.

[32] Segal, O., Cabot-Miller, J., Adaricheva, K., Nation, J., Sharaphudinov, A.: The bases of association rules of high confidence. Proc. DTNM 2018. Draft available at math.hawaii.edu/∼jb/.

[33] Sia, D., Jiao, Y., Martinez-Quetglas, I., Kuchuk, O., Villacorta-Martin, C., Castro de Moura, M., Putra, J., Camprecios, G., Bassaganyas, L., Akers, N., Losic, B., Waxman, S., Thung, S.N., Mazzaferro, V., Esteller, M., Friedman, S.L., Schwartz, M., Villanueva, A., Llovet, J.M.: Identification of an immune-specific class of hepatocellular carcinoma, based on molecular features. Gastroenterology (2017), doi: 10.1053/j.gastro.2017.06.007.

[34] Sinha, S., Tsang, E.K., Zeng, H., Meister, M., Dill, D.L.: Mining TCGA Data Using Boolean Implications. PLOS One July 23, 2014; doi 10.1371/journal.pone.0102119.

[35] Tamborero, D., Gonzalez-Perez, A., Perez-Llamas, C., Deu-Pons, J., Kandoth, C., Reimand, J., Lawrence, M.S., Getz, G., Bader, G.D., Ding, L., Lopez-Bigas, N.: Comprehensive identification of mutational cancer driver genes across 12 tumor types, Scientific Reports 3, 2650 (2013), https://digitalcommons.wustl.edu/open_access_pubs/4316.

[36] Yang, D., Hanna, D.L., Usher, J., LoCoco, J., Chaudhari, P., Lenz, H.J., Setiawan, V.W., El-Khoueiry, A.: Impact of sex on the survival of patients with hepatocellular carcinoma: a surveillance, epidemiology, and end results analysis. Cancer 120, 3707–3716 (2014), doi: 10.1002/cncr.28912.

Department of Mathematics, University of Hawai'i, Honolulu, HI 96822, USA

Email address: jb@math.hawaii.edu

Northeastern University, Boston, MA 02115 USA
Email address: `achari.a@husky.neu.edu`

University of Hawai‘i, Honolulu, HI 96822 USA
Email address: `tristanh314@math.hawaii.edu`

University of Hawai‘i, Honolulu, HI 96822 USA
Email address: `jmaligro@hawaii.edu`

University of Hawai‘i Cancer Center, Honolulu, HI 96813, USA
Email address: `gokimoto@cc.hawaii.edu`

SNR Analytics, Kaneohe, HI 96744, USA
Email address: `twenska@gmail.com`

University of Hawai‘i, Honolulu, HI 96822 USA
Email address: `tammyky@hawaii.edu`

University of Hawai‘i, Honolulu, HI 96822 USA
Email address: `ez9@hawaii.edu`