Joint Grid Discretization for Biological Pattern Discovery

Jiandong Wang
Department of Computer Science
New Mexico State University
Las Cruces, New Mexico, USA
wangjd24@nmsu.edu

Sajal Kumar
Department of Computer Science
New Mexico State University
Las Cruces, New Mexico, USA
sajal49@nmsu.edu

Mingzhou Song*
Department of Computer Science
Molecular Biology Graduate Program
New Mexico State University
Las Cruces, New Mexico, USA
joemsong@cs.nmsu.edu

ABSTRACT

The complexity, dynamics, and scale of data acquired by modern biotechnology increasingly favor model-free computational methods that make minimal assumptions about underlying biological mechanisms. For example, single-cell transcriptome and proteome data have a throughput several orders more than bulk methods. Many model-free statistical methods for pattern discovery such as mutual information and chi-squared tests, however, require discrete data. Most discretization methods minimize squared errors for each variable independently, not necessarily retaining joint patterns. To address this issue, we present a joint grid discretization algorithm that preserves clusters in the original data. We evaluated this algorithm on simulated data to show its advantage over other methods in maintaining clusters as measured by the adjusted Rand index. We also show it promotes global functional patterns over independent patterns. On single-cell proteome and transcriptome of leukemia and healthy blood, joint grid discretization captured known protein-to-RNA regulatory relationships, while revealing previously unknown interactions. As such, the joint grid discretization is applicable as a data transformation step in associative, functional, and causal inference of molecular interactions fundamental to systems biology. The developed software is publicly available at https://cran.r-project.org/package=GridOnClusters

CCS CONCEPTS

• **Theory of computation → Unsupervised learning and clustering**; • **Applied computing** → *Bioinformatics*; Biological networks.

KEYWORDS

Grid discretization, Pattern discovery, Clustering, Functional chi-squared statistics, Single-cell sequencing

ACM Reference Format:
Jiandong Wang, Sajal Kumar, and Mingzhou Song. 2020. Joint Grid Discretization for Biological Pattern Discovery. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '20), September 21–24, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3388440.3412415

*Corresponding author

BCB '20, September 21–24, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7964-9/20/09…$15.00
https://doi.org/10.1145/3388440.3412415

1 INTRODUCTION

Discretization converts analog to digital signals. It is most often used to represent measurements from the physical world such as a biological system. Most discretization methods are univariate, such as equal-bin-width quantization, optimal univariate k-means [23, 25, 29], maximum entropy quantization [15], and maximum likelihood quantization [24]. Multivariate discretization methods are also developed, including vector quantization [11] and sequential multivariate quantization [18]. These methods optimize on the fidelity of a discretized signal to the original signal, often measured by squared errors.

The utility of discretization has progressed from signal representation to pattern discovery. Notably, certain model-free methods for pattern discovery rely on discrete representation of signals to rid the process of any parametric assumption. They include mutual information and the classical Pearson's chi-squared test of association [19], the recent functional tests [17, 32, 35], and discrete causal inference methods such as causal inference by stochastic complexity [2] and hidden compact representation [3]. These model-free methods detect symmetric, functional, or causal dependencies between variables without assuming a mathematical model for a relationship. They are used to study biological interactions from measurements of molecular abundance such as transcriptome or proteome data [6, 13, 16, 28, 34, 35]. However, univariate discretization, often used before these methods, is ineffective if marginal distributions do not reflect patterns in the joint distribution.

To transform analog to digital signals for model-free pattern discovery, we present a joint grid discretization method that preserves clusters in the original data. Instead of using the mainstream squared error criterion, the method finds a grid to preserve the relative positions of clustered patterns in the data. The grid is then used to discretize the data for downstream analysis, such as constructing a contingency table.

We benchmarked joint grid discretization in contrast to marginal methods by simulation studies and single-cell multiomic datasets of multiple phenotype acute leukemia [10]. The performance is measured by the adjusted Rand index to evaluate the strength of cluster preservation. The functional chi-squared statistic (FunChisq) and mutual information score [14] are also used to indicate how well a discrete function is present in discrete data. Joint grid discretization demonstrates advantages in maintaining cluster patterns and improving functional pattern discovery accuracy over marginal methods. On a subset of the single-cell leukemia dataset, novel protein-RNA interactions are revealed via the strength of discrete

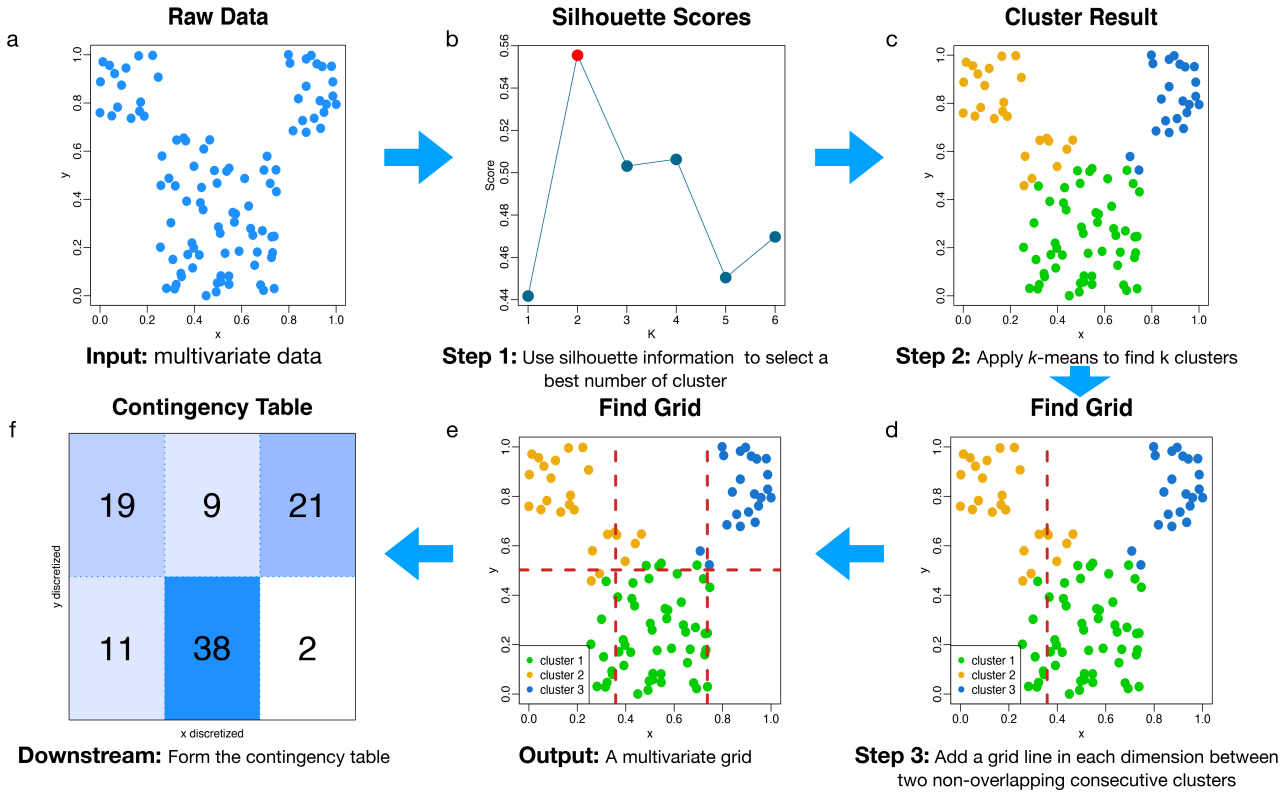


Figure 1: Overview of joint grid discretization. (a) The input is multivariate continuous data. (b) Step 1, silhouette information is used to select a best number of clusters k. (c) Step 2, data are clustered by k-means. (d) Step 3, hyperplanes (grid lines in 2D) cutting each dimension are found to preserve the clustering. (e) The output is a multivariate grid. (f) For downstream analysis, a contingency table is formed by discretizing the input continuous data using the grid.

functional pattern. Its success on noisy single-cell data suggests a broad range of possibilities in studying complex biological systems where model-free approaches are beneficial for discovery of previously unknown mechanisms.

2 METHODS

2.1 Overview of joint grid discretization

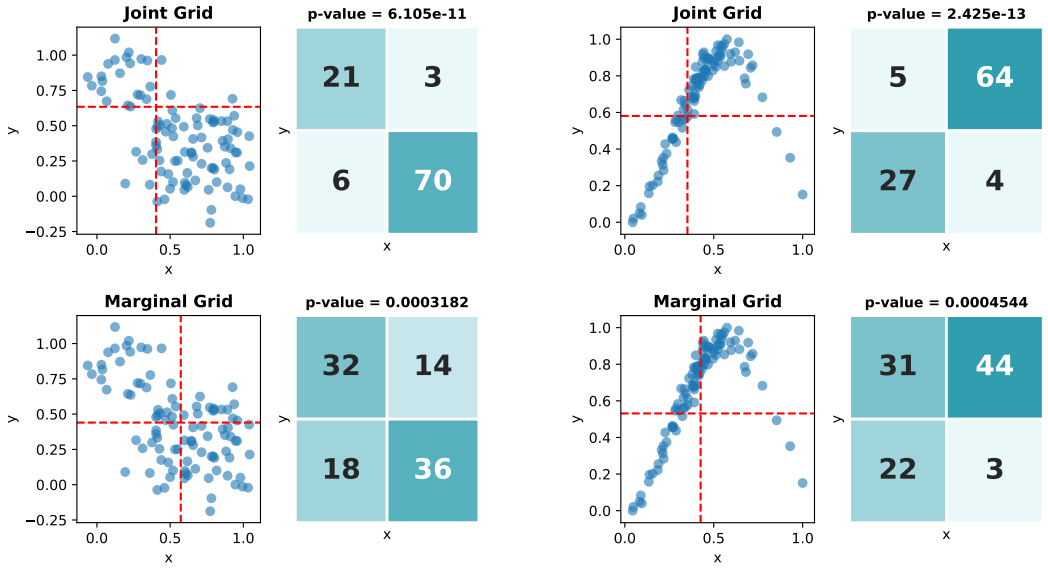
To capture multivariate patterns, we introduce a joint grid discretization algorithm whose main steps are summarized in Figure 1. Previous joint grid finding approaches are based on likelihood or entropy but not based on cluster preservation, while marginal grids obtained by discretizing each dimension separately are blind to joint patterns.

Figure 2 illustrates the advantage of joint discretization. Both joint and marginal grid discretization are applied on bivariate data on two functional patterns of a chessboard and a noisy curve. Here, the marginal distributions of both dimensions are uniform. The joint grid captured the patterns visibly better than the marginal grid, and also better by the lower p-values of FunChisq obtained from the contingency tables built from discretized data. As uniform marginal distributions do not carry any information regarding a joint pattern, the marginal grid is nearly blind to the joint patterns in both cases. The examples thus provide an intuition why joint grid discretization preserves a joint pattern much better than marginal grid discretization.

The joint grid discretization method is given as Algorithm 1 JOINT-GRID-DISCRETIZATION. Its input is d-dimensional multivariate data set X of n points. There are three main steps in the algorithm. The first step is to perform multivariate k-means clustering; the second step is to select an optimal number of clusters by silhouette information; and the last step is to find hyperplanes (grid lines in 2D) perpendicular to the axes that best separate the projection of clusters onto each dimension. The time complexity of the joint grid discretization algorithm is $O(k_{\max} n^2 + d(n+k)\log n)$, where

Algorithm 1 JOINT-GRID-DISCRETIZATION(X)

```
for k = 2 to k_max
    C(k) = Perform k-means clustering on X with k clusters
    if k == 2 or clustering C(k) increases average silhouette
    width over C*
        C* = C(k)
G = FIND-GRID(X, C*)
return grid G
```



A chessboard pattern A noisy curve

Figure 2: The advantage of joint over marginal grid discretization on a chessboard pattern and a noisy curve. The corresponding contingency tables are shown to the right of each pattern. p-values are obtained by the FunChisq test on the contingency tables.

the first term is the time for the clustering and k selection steps, and the second term is the time needed for finding a grid.

2.2 Finding homogeneous clusters

We first cluster the input multivariate continuous data. The goal is to represent the data in a way such that each cluster is isotropic, meaning points within a cluster are homogeneous without a strong direction. Such clustering allows one to capture global changes across homogeneous clusters instead of hiding them inside one large heterogeneous cluster. This is often critical for downstream pattern discovery. We use k-means clustering in the Euclidean space to achieve this step.

We use the average silhouette width of a clustering to select an optimal number of clusters. Silhouette information characterizes the separation across clusters versus the compactness within each cluster. Unlike other internal cluster quality measures, silhouette is not biased to a large number of clusters making it suitable for selecting the best number of clusters. For a given clustering, the average silhouette width is in the range of $[-1, 1]$. The greater the width, the better is the clustering. We select the best k-means clustering over a various number of clusters from 2 to $k_{\max}$, as specified by the user. A clustering with the greatest average silhouette width is selected as the optimal clustering to find a grid.

2.3 Determining a grid that preserves clusters

To identify a grid that preserves a given clustering, we design Algorithm 2 Find-Grid. Its input is the original continuous data and a clustering of the data. Its output is a grid. It projects the data and clustering to each dimension to determine whether a hyperplane (line in 2D) is necessary between consecutive clusters

Algorithm 2 Find-Grid(X, C)

```
1  n_1, ..., n_k = number of points in clusters 1, ..., k in C
2  for each dimension d of X
3      Y_1, ..., Y_k = dimension d coordinate vectors of k clusters
       in ascending order by medians
4      for j = 1 to k − 1
5          W = Intersect(Y_j, Y_{j+1})
6          line = Find-Line(W, 0, |W|, n_j, n_{j+1})
7          Append line to dimension d in G
8  return grid G
```

and if so the location of a hyperplane. For two dimensional input data, the grid is composed of lines; for higher dimensions, the grid is a collection of hyperplanes.

We find grid lines in each dimension by minimizing line crossings between projections of consecutive clusters. The rationale is that cluster projection already reflects the joint pattern. Working with projections preserves the joint patterns. We first decide whether two cluster projections have sufficient separation by their overlap. We define the error rate e_j as the percentage of points in cluster j that cross the grid line away from the cluster median. If the sum of the error rates of two neighboring projected clusters $e_j + e_{j+1}$ is more than 50%, we declare the two clusters overlap in the current dimension and no grid line is added to separate these two clusters. Otherwise, we add a grid line at a location where the error rates of the two neighboring clusters are about equal. In the algorithm, ϵ is the machine precision, which we set to 0.005.

To speed up error rate calculation, we design an index-based search strategy given as Algorithm 3 Find-Line. Direct computation of error rate e is expensive since it has to be calculated twice in each dimension for each cluster by going through the entire cluster to obtain the number of misplaced points. The search algorithm based on index can speed up this process. The main idea is to use an index to provide information about the number of misplaced points. As the number is fixed for each position, we calculate it only once for repeated use. The index tells us how many points are misplaced for two consecutive clusters at the position we examined. We create a data structure called W to maintain the index information.

Algorithm 3 Find-Line($W, left, right, size_1, size_2$)

```
line_index = ⌈(left + right)/2⌉
line = W[0][line_index]
Calculate error rates e1 and e2 for cluster1 and cluster2
err_sum = e1 + e2; err_diff = |e1 − e2|
if left == right − 1 or line_index == right or err_diff < ϵ
    // base case:
    if err_sum > 0.5
    // Clusters j and j + 1 overlap in dimension d
        return null
    elseif err_sum < 0.5
    // Clusters j and j + 1 do not overlap in dimension d
        return line
if e1 > e2 // search the right side
    return Find-Line(W, line_index, right, size1, size2)
elseif e1 < e2 // search the left side
    return Find-Line(W, left, line_index, size1, size2)
```

We design Algorithm 4 Intersect to calculate this index for two given clusters. The idea is illustrated in Figure 3. We first merge two consecutive clusters and index points by the decreasing order for cluster 1 and increasing order for cluster 2. Then we extract points in cluster 1 and cluster 2 in between the cluster medians. In case the line overlaps a point in the data, we calculate mid-point for every pair of consecutive points and use right side index for cluster 1 and left side index for cluster 2, as index will indicate how many points in that cluster cross the line. The following theorem justifies searching between cluster medians.

Theorem 2.1. *Given two multisets S_1 and S_2, each containing n_1 and n_2 real numbers, respectively. Let α_1 and α_2 be the medians of S_1 and S_2, respectively. We assume $\alpha_1 \le \alpha_2$ without loss of generality. Let t be a real number. Using t as a decision boundary, we define the error rates for S_1 and S_2 by*

$$e_1 = |\{x | x \ge t, x \in S_1\}|/n_1$$

and

$$e_2 = |\{x | x \le t, x \in S_2\}|/n_2$$

If $t < \alpha_1$ or $t > \alpha_2$, then the sum of error rates $e_1 + e_2 \ge 50\%$.

Proof. By the definition of error rate and as t increases (moving from left to right), e_1 decreases from 100% to 0 and e_2 increases from 0 to 100%. When $t = \alpha_1$, it is always true that $e_1 \ge 50\%$; when $t = \alpha_2$, we always have $e_2 \ge 50\%$. Thus, if $t < \alpha_1$ or $t > \alpha_2$, either $e_1 \ge 50\%$ or $e_2 \ge 50\%$. As the error rate is always non-negative, we can conclude that if $t < \alpha_1$ or $t > \alpha_2$, then the sum of error rates $e_1 + e_2 \ge 50\%$. □

Algorithm 4 Intersect(Y_1, Y_2)

```
Merge and sort points in the two clusters
Create an index data structure based on decreasing order for Y1
and increasing order for Y2
Extract points and corresponding indices between the medians
of Y1 and Y2
Calculate mid-points for every two consecutive points and
assign the corresponding index
return mid-points and their indices
```

To find a grid line in a given dimension, we perform binary search between projections of two consecutive cluster medians. We evaluate error rates of the two clusters if the line is put at the middle of two end points. Let e_1 and e_2 be the error rates for the left and right neighboring clusters. We repeatedly shrink the search interval by half until either $e_1 + e_2 > 50\%$, $e_1 == e_2$, or the interval width is zero. The binary search strategy is correct because the error rate difference of $e_1 - e_2$ will not increase as the line position increases in the given dimension.

The runtime of Find-Grid includes both sorting and search, giving rise to a time complexity of $O(dn \log n)$.

2.4 Forming a contingency table by a grid

For discrete pattern discovery, data are discretized by the grid. Algorithm 5 Form-Contingency-Table(Z, G, d) produces a contingency table, whose columns are dimension d of Z as the dependent variable, and rows are all remaining dimensions of Z as independent variables. The contingency table can be used for the FunChisq test, Pearson's chi-squared test, or mutual information calculation.

Algorithm 5 Form-Contingency-Table(Z, G, d)

```
Let L_d be the number of intervals in dimension d in grid G
Discretize Z_d by G to Y of s = L_d levels
Discretize Z\{Z_d} by G to X of r = ∏_{j=1, j≠d}^{D} L_j levels
Form a table T of dimensions r × s
Map each point (x, y) ∈ (X, Y) to T using grid G
return contingency table T
```

2.5 Functional chi-squared statistics

One downstream analysis of discretization is functional pattern discovery. Given n observations of two discrete variables, X with r unique levels and Y with s unique levels, the FunChisq statistic measures the functional dependency $X \to Y$ in an $r \times s$ contingency table with X as rows and Y as columns defined as

$$\chi^2_f(X \to Y) = \sum_{i=1}^{r} \sum_{j=1}^{s} \frac{[n_{ij} - (n_{i\cdot}/s)]^2}{n_{i\cdot}/s} - \sum_{j=1}^{s} \frac{[n_{\cdot j} - (n/s)]^2}{n/s} \quad (1)$$

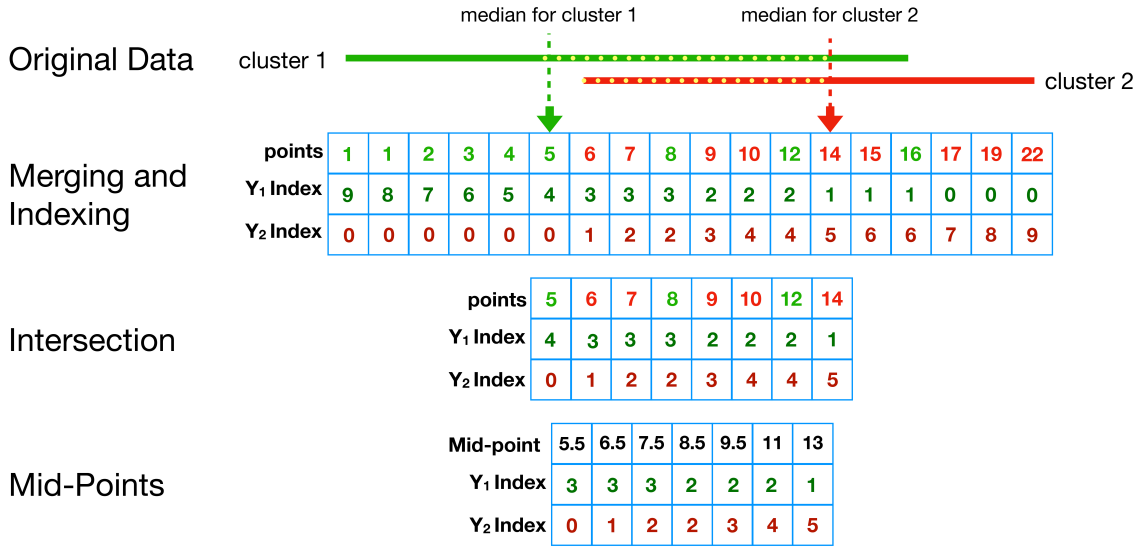


Figure 3: Data structures used in the INTERSECT algorithm. The green solid line indicates cluster 1 and the red solid line is cluster 2. The two vertical dashed lines show the medians of each cluster.

where n_{ij} is joint frequency at row $X = i$ and column $Y = j$, $n_{i\cdot}$ is the sum of row $X = i$ and $n_{\cdot j}$ is the sum of column $Y = j$. FunChisq follows the chi-squared distribution asymptotically with $(r-1)\cdot(s-1)$ degrees of freedom under the null hypothesis of X and Y being independent when Y is uniformly distributed. Other mathematical properties of FunChisq are derived in [17, 32, 35].

3 RESULTS

3.1 Simulation studies

3.1.1 Simulated patterns. To evaluate the joint grid discretization method, we generated 15 functional patterns as shown in Figure 4, where each pattern is either a discrete or a continuous function. The patterns shown do not carry noise. Various levels of Gaussian noise are later added in the simulation study. Each pattern contains 500 sample points. The first eight patterns are highly discrete and the last seven are smooth functions. Zero-mean Gaussian noise at standard deviation levels from 0 to 150% of the signal standard deviation is added to each signal.

3.1.2 Using adjusted Rand index to measure cluster preservation. We first compare the joint grid (GOC) and three marginal discretization methods for cluster preservation in our simulation studies. For all four methods, we use the same table dimension as determined by the joint grid. The marginal grid discretization is performed using the `discretize` function in the R package 'arules' [12] with '`cluster`'—univariate k-means (KMEANS), '`frequency`'—maximum entropy (FREQ) and '`interval`'—equal bin width (INT) strategies. Marginal discretization was performed on each dimension to obtain a grid. This grid is used to generate a contingency table. Then we calculate the adjusted Rand index (ARI) score to evaluate the performance of each method. The (unadjusted) Rand index computes a similarity measure between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clustering. ARI is the corrected-for-chance version of the Rand index. Such a correction for chance establishes a baseline by using the expected similarity of all pair-wise comparisons between clusterings specified by a random model [21, 27]. The ARI of the four methods are shown in Figure 5. The higher the ARI, the better preservation of the original clusters. From Figure 5, we observe that joint grid discretization method has better ARI scores at most noise levels, capturing the clusters more accurately than marginal discretization methods.

3.1.3 Using the FunChisq test and mutual information to measure functional dependency. Next, we applied the FunChisq test and mutual information on the contingency table obtained from the grid to calculate a p-value and a score, respectively, to measure the functional dependency. Distributions of FunChisq p-values and mutual information of the four methods are shown in Figure 6. On functional patterns at low noise levels, joint grid discretization outperformed marginal grid discretization in returning smaller median p-values and mutual information scores. On noisy functional patterns which are getting closer to independent patterns, all methods returned p-values and mutual information close to one. Figure 6 suggests that the joint grid method did better then the marginal methods at most noise levels.

3.1.4 Functional relationships versus independent patterns. We also evaluated if joint grid, along with three marginal discretization strategies promote functional over independent patterns at increasing levels of noise. For each functional pattern at each noise level, an independent pattern was generated by two random normal variables with their mean and standard deviation same as the variables in the functional pattern. Figure 7(a) and (b) show the area under the receiver operating characteristic (AUROC) curve and area under the precision recall (AUPR) curve over increasing noise for four discretization strategies obtained using FunChisq. Figure 7(c) and

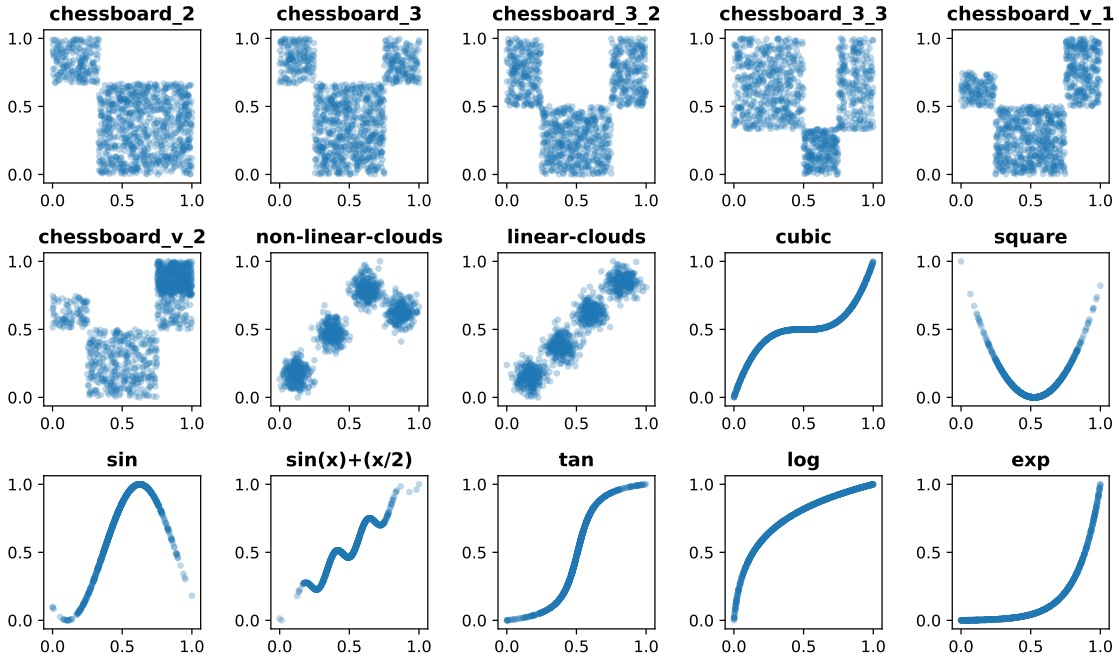


Figure 4: Fifteen test cases representing discrete and smooth functional patterns to evaluate the performance of discretization. In these patterns, Y (the vertical axis) is a function of X (the horizontal axis).

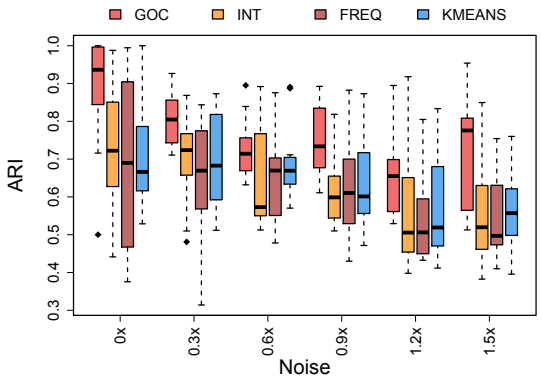


Figure 5: Cluster preservation by joint and marginal discretization methods on simulated data. The horizontal axis represents the noise level percentage at multiples of signal standard deviation. The vertical axis is ARI score.

(d) used mutual information instead. Joint grid (GOC) is the overall top performer until noise level 0.8×, after which the performance becomes unstable as functional patterns approach independence. K-means clustering (KMEANS) is second best, followed by maximum entropy (FREQ) and equal bin width (INT) that are relatively similar.

3.2 Discovering novel combinatorial protein to RNA interactions in leukemia cells

Greenleaf and colleagues [10] generated multi-omic single-cell datasets containing proteome and transcriptome of cells of healthy developing blood and cells of mixed-phenotype acute leukemia (MPAL). Using single-cell RNA sequencing (scRNA-seq) [33], they measured 20,287 RNA transcripts across 53,638 cells. They also measured 21 surface proteins across 52,886 cells using single-cell antibody-derived tag sequencing (scADT-seq) [26].

Following the steps in [10], we processed both datasets by first removing zero variance cells. We also removed zero variance RNAs, while no proteins of zero variance were found. After filtration, the scRNA-seq data had 17,451 RNAs and scADT-seq data had 21 proteins across 52,885 matched single-cells. Next, we transformed scADT-seq using the centered log ratio (CLR) as explained in [26]. We filtered scRNA-seq using $\log_2$ transformation after normalizing it to count per thousand (CPT) and adding one, given by: $R'_{ij} = \log_2\left(1 + \left(R_{ij} \cdot 10,000 / R_{\cdot j}\right)\right)$, where R_{ij} is the count of RNA i in cell j and $R_{\cdot j}$ is the sum of unique molecular identifiers in cell j.

We evaluated if the four discretization strategies are able to capture and prefer known combinatorial protein to RNA interactions over shuffled independent patterns. We obtained 551 known protein-RNA interactions from Pathway Commons [4] that have documented experimental evidence for physical interactions. Among the 551 interactions, 176 RNAs were influenced by two or more proteins. For each of the 176 RNAs, we obtained its two most 'active' proteins by selecting ones with the highest non-zero median absolute deviation (MAD) across samples, scaled by the number

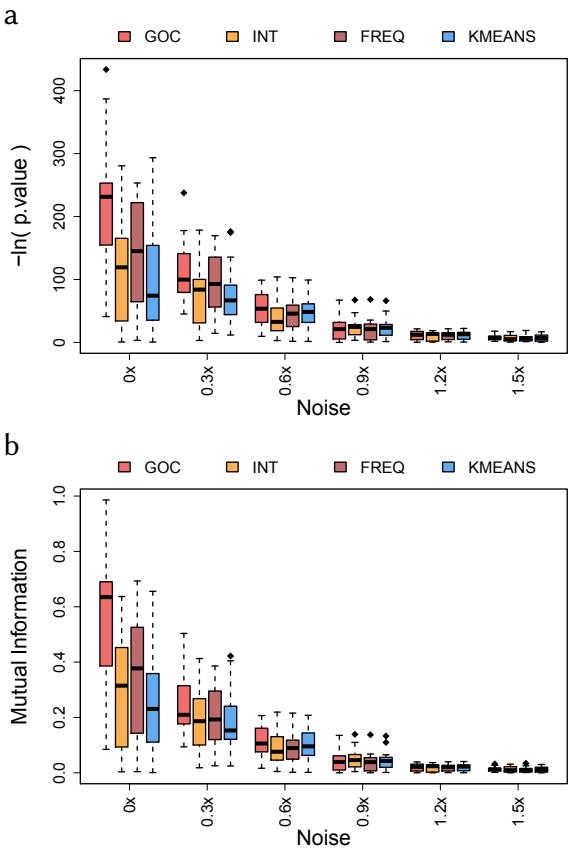


Figure 6: The performance of joint and marginal discretization methods on simulated functional patterns. The horizontal axis represents the noise level percentage at multiples of the signal standard deviation. The vertical axis is (a) FunChisq (− ln p-value) and (b) mutual information. In both cases, a high value suggests a strong functional pattern.

of non-zero single-cells. Thus, we evaluated 176 protein × protein → RNA interactions versus 176 independent patterns by means of mutual information and FunChisq p-value. The independent patterns were generated by asynchronously shuffling the single cells in the original interaction.

For joint grid, each interaction was jointly discretized to obtain a grid G. Algorithm 5 Form-Contingency-Table was then used on the three dimensional Z containing two proteins and one RNA. For marginal discretization, all proteins and RNAs were individually discretized with the same number of levels as that obtained by joint grid. Independent patterns were discretized in the same manner. When discretizing, we only considered samples that were non-zero across the two proteins and the RNA.

Figure 8 shows the ROC and PR curves generated using FunChisq, GOC (AUROC=0.89, AUPR=0.93) is the top performer followed by the KMEANS and FREQ (AUROC=0.87, AUPR=0.92) performing equally and INT (AUROC=0.86, AUPR=0.91) performing slightly worse than others. A similar trend is seen in Figure 8(c) and (d) generated using mutual information, where GOC (AUROC=0.81,

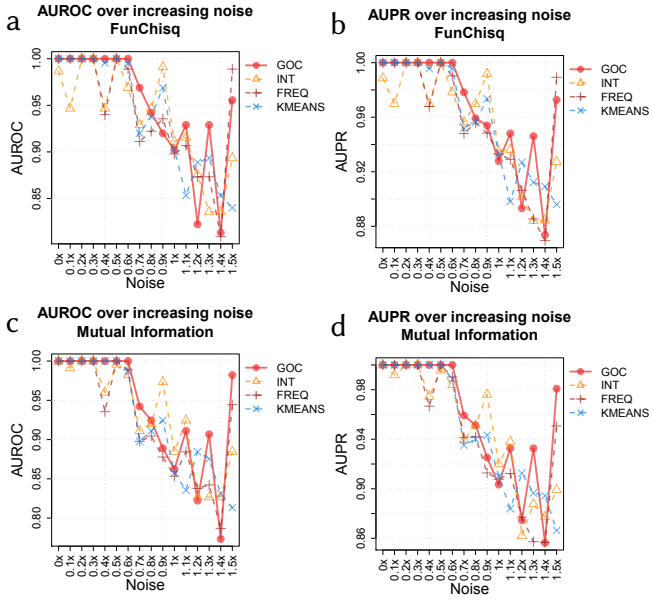


Figure 7: Detecting functional versus independent patterns. (a,c) AUROC and (b,d) AUPR over increasing noise obtained via FunChisq and mutual information, respectively.

AUPR=0.77) is the overall top performer, followed by KMEANS and FREQ (AUROC=0.81, AUPR=0.75) with INT (AUROC=0.81, AUPR=0.73) coming in last.

3.2.1 Cluster preservation among combinatorial interactions. Figure 9 shows the ARI distribution for all 176 protein × protein → RNA interactions, ordered by median ARI. KMEANS (median ARI=0.450) is the best performer followed by GOC (median ARI=0.434). INT (median ARI=0.420) with a large dynamic range comes in third while FREQ (median ARI=0.174) performed poorly.

3.2.2 Discovering putative protein-RNA interactions. To discover putative protein-RNA interactions that may play a role in leukemia, we applied the four discretization methods to a subset of the leukemia dataset and computed four FunChisq p-values. Four surface marker proteins CD14, CD3D, CD19, and CD8A with known roles in healthy blood development [10] were selected. 1,000 RNAs with the highest non-zero MAD scaled by the number of non-zero single-cells were also selected. Thus, we computed the FunChisq p-values for 4000 interactions under the four discretization schemes.

Figure 10 shows the joint kernel densities of top four patterns ordered by their aggregate ranks of FunChisq p-value on four discretization schemes. Each subplot also shows the grid lines and FunChisq p-value for each discretization strategy. CD3D (CD3d Molecule) indirectly regulates *B2M* (β-2-Microglobulin) in Figure 10(a) as CD3D protein governs T-cell development [22] as a part of T-cell receptor (TCR) complex, while the TCR stimulus (or CD3/CD28 costimulus) induces USF1 [9] which in turn regulates *B2M* transactivation in hematopoietic cells [8]. In Figure 10(b), CD14 (CD14 Molecule) may have regulatory impact on *FTL* (Ferritin Light Chain) as soluble CD14 (sCD14) in patients on hemodialysis showed positive

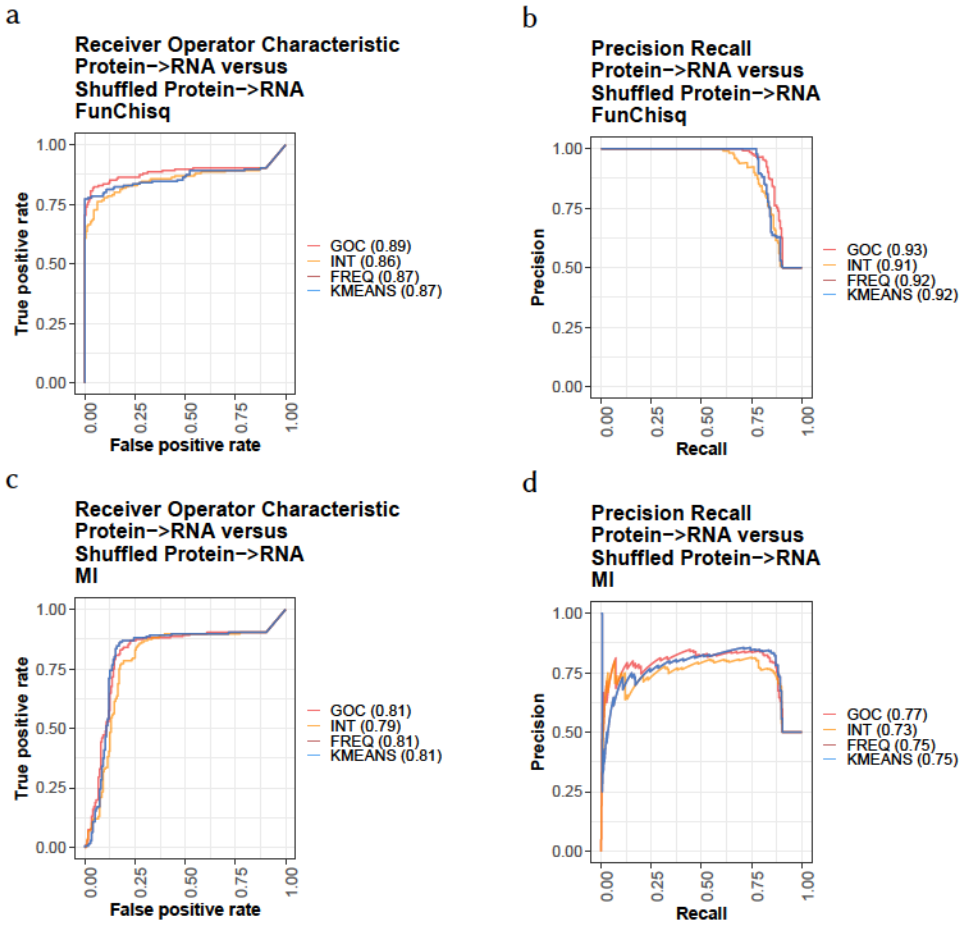


Figure 8: Performance on Leukemia dataset. (a),(c) ROC and (b),(d) PR using FunChisq and mutual information, respectively.

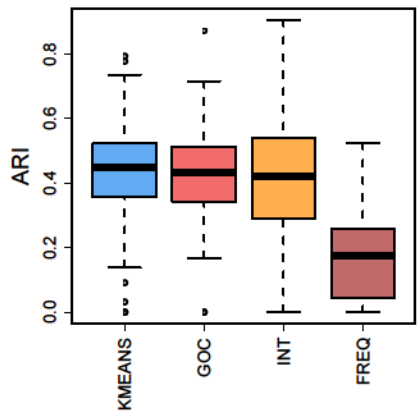


Figure 9: Performance of joint and marginal discretization methods on leukemia datasets expressed through adjusted Rand index distribution.

correlation with serum ferritin encoded by *FTL* [22], among other biochemical variables [20]. In Figure 10(c), CD14 and *S100A6* (S100 Calcium Binding Protein A6) differentially express in porcine alveolar macrophages' response to *Haemophilus parasuis* in pigs [30]. Additionally, they are also found to differentially express together in rat blood under heat stress [5]. In Figure 10(d), CD14 indirectly regulates *CTSS* (Cathepsin S) by initiating a pathway that induces SRC-family kinase [31]. SRC in turn, is known to control *CTSS* [7]. In each case, the grid lines of GOC are appropriately placed boosting the FunChisq p-value, while other methods either cut in or around a dense region, except KMEANS in Figure 10(d). In Figure 10(a), the INT placement of the grid lines makes an otherwise functional pattern, non-functional.

4 DISCUSSION

To create contingency tables that can represent global patterns within the data, we developed a joint grid discretization approach. Our findings suggest that it is more sensitive to functional patterns than a marginal grid approach, but treats independent patterns equally with the marginal grid approach.

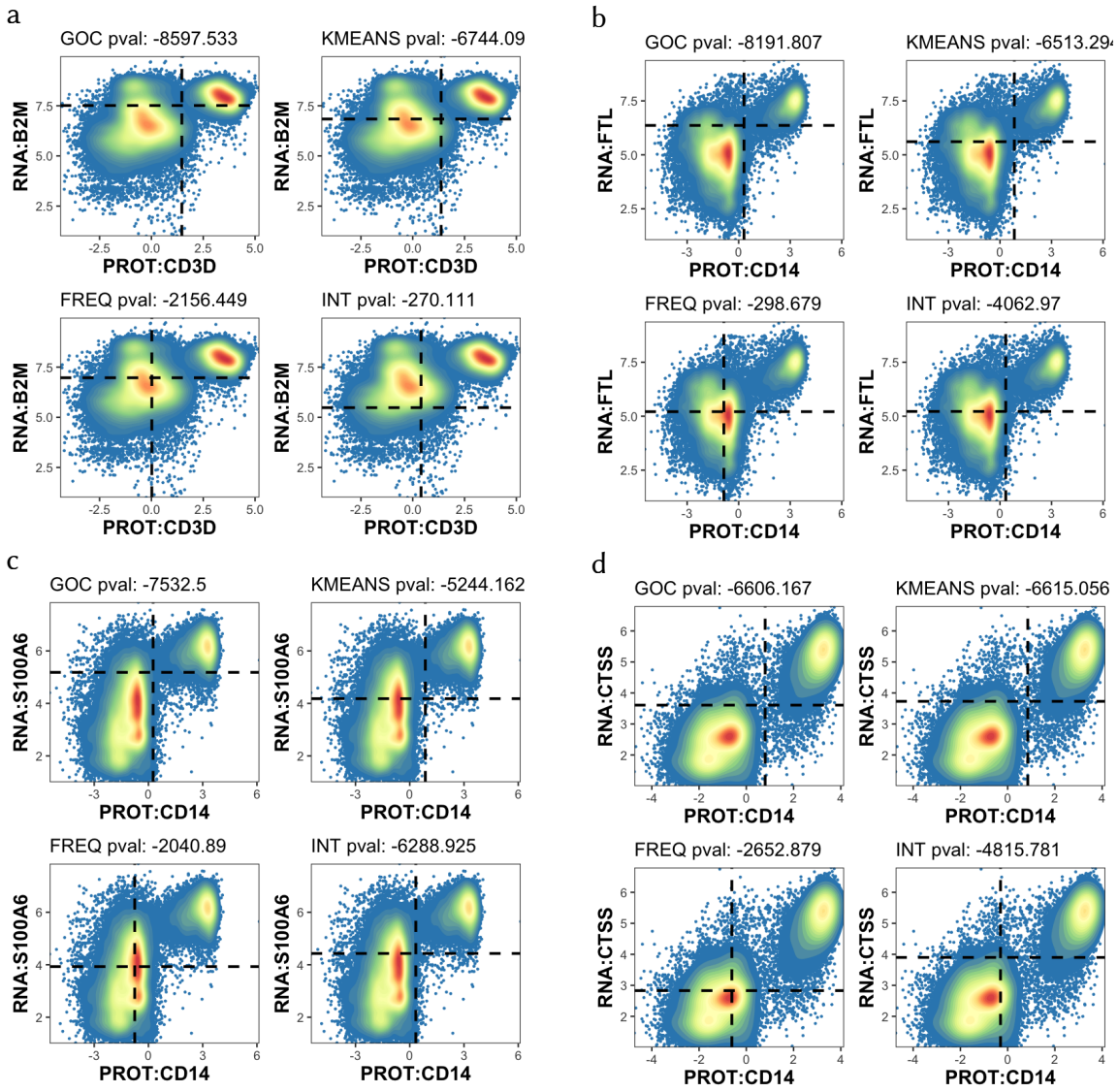


Figure 10: Discretization on top four putative patterns. (a) CD3D→B2M, (b) CD14→FTL, (c) CD14→S100A6 and (d) CD14→CTSS. Each point represents a cell and warmer colors (approaching red) indicate higher cell concentration. Each plot contains four subplots showing the dashed grid lines obtained by GOC, KMEANS, FREQ and INT and their respective FunChisq $\ln p$**-value.**

Vector quantization, consistent with k-means clustering, returns decision boundaries not parallel to the axes of each variable as in grid discretization. A grid parallel to axes is often required for model-free pattern discovery such as in detecting gene-gene interactions.

Grid discretization is different from grid-based clustering such as the CLIQUE method [1]. The goal of the former is to identify a grid whose cells cover homogeneous clusters formed by the data; the goal of the latter is to group cells in a fine grid into clusters efficiently in high dimensions.

We removed cell and gene outliers but did not normalize the single-cell leukemia data by library size as typically done. The primary reason is that all library size normalization methods we tried introduced artifacts of perfect lines in gene-gene scatter plots. Additionally, we do not observe obvious systematic effects due to library size in top gene-gene interactions. Artifacts introduced during library size normalization do raise a concern for further studies.

Even top known gene-gene interaction patterns appear much noisier in single-cell (Figure 10) than bulk RNA-seq data we have seen in the past. Such data quality is not desirable but unavoidable due to current challenges in single-cell instrumentation. But it does support the use of a discretization method to capture only globally important variations rather than local details, which may benefit downstream model-free pattern discovery tasks.

5 CONCLUSIONS

A joint grid discretization algorithm is introduced and demonstrated for its desirable performance in preserving clusters and functional patterns in the data. It is unconventional as its focus is not to minimize the squared errors between the discretized points and their continuous originals. Thus its capacity is not in the numerical precision as needed in modeling, but in locking in the global patterns required for model-free pattern discovery. It is thus beneficial to a broad range of biology applications with a goal to discover from noisy data hidden biological mechanisms.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1661331, and also by Specialty Crop Research Initiative Grant No. 2016-51181-25408/Project Accession No. 1009971 from the USDA National Institute of Food and Agriculture.

REFERENCES

[1] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. 2005. Automatic subspace clustering of high dimensional data. *Data Mining and Knowledge Discovery* 11, 1 (2005), 5–33.

[2] Kailash Budhathoki and Jilles Vreeken. 2017. MDL for causal inference on discrete data. In *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 751–756.

[3] Ruichu Cai, Jie Qiao, Kun Zhang, Zhenjie Zhang, and Zhifeng Hao. 2018. Causal discovery from discrete data using hidden compact representation. In *Advances in Neural Information Processing Systems*. 2666–2674.

[4] Ethan G Cerami, Benjamin E Gross, Emek Demir, Igor Rodchenkov, Özgün Babur, Nadia Anwar, Nikolaus Schultz, Gary D Bader, and Chris Sander. 2011. Pathway Commons, a web resource for biological pathway data. *Nucleic Acids Research* 39, suppl 1 (2011), D685–D690.

[5] Jinhuan Dou, Adnan Khan, Muhammad Zahoor Khan, Siyuan Mi, Yajing Wang, Ying Yu, and Yachun Wang. 2020. Heat Stress Impairs the Physiological Responses and Regulates Genes Coding for Extracellular Exosomal Proteins in Rat. *Genes* 11, 3 (2020), 306.

[6] E Dvořáková, S Kumar, J Kléma, F Železný, K Drbal, and M Song. 2019. Evaluating Model-free Directional Dependency Methods on Single-cell RNA Sequencing Data with Severe Dropout. In *Proceedings of International Conference on Bioinformatics Research and Applications*. Seoul, South Korea, 55–62.

[7] Jaya Gautam, Suhrid Banskota, Hyunji Lee, Yu-Jeong Lee, Yong Hyun Jeon, Jung-Ae Kim, and Byeong-Seon Jeong. 2018. Down-regulation of cathepsin S and matrix metalloproteinase-9 via Src, a non-receptor tyrosine kinase, suppresses triple-negative breast cancer growth and metastasis. *Experimental & Molecular Medicine* 50, 9 (2018), 1–14.

[8] Sam JP Gobin, Paula Biesta, and Peter J Van den Elsen. 2003. Regulation of human $\beta 2$-microglobulin transactivation in hematopoietic cells. *Blood, The Journal of the American Society of Hematology* 101, 8 (2003), 3058–3064.

[9] Aparna Godavarthy, Ryan Kelly, John Jimah, Miguel Beckford, Tiffany Caza, David Fernandez, Nick Huang, Manuel Duarte, Joshua Lewis, Hind J Fadel, et al. 2020. Lupus-associated endogenous retroviral LTR polymorphism and epigenetic imprinting promote HRES-1/Rab4 expression and mTOR activation. *JCI Insight* 5, 1 (2020).

[10] Jeffrey M Granja, Sandy Klemm, Lisa M McGinnis, Arwa S Kathiria, Anja Mezger, M Ryan Corces, Benjamin Parks, Eric Gars, Michaela Liedtke, Grace XY Zheng, Howard Y Chang, Ravindra Majeti, and William J Greenleaf. 2019. Single-cell multiomic analysis identifies regulatory programs in mixed-phenotype acute leukemia. *Nature Biotechnology* 37, 12 (2019), 1458–1465.

[11] Robert Gray. 1984. Vector quantization. *IEEE ASSP Magazine* 1, 2 (1984), 4–29.

[12] Michael Hahsler, Sudheer Chelluboina, Kurt Hornik, and Christian Buchta. 2011. The arules R-Package Ecosystem: Analyzing Interesting Patterns from Large Transaction Datasets. *Journal of Machine Learning Research* 12 (2011), 1977–1981. http://jmlr.csail.mit.edu/papers/v12/hahsler11a.html

[13] Sajal Kumar, Hua Zhong, Ruby Sharma, Yiyi Li, and Mingzhou Song. 2018. Scrutinizing functional interaction networks from RNA-binding proteins to their targets in cancer. In *IEEE International Conference on Bioinformatics and Biomedicine*. Madrid, Spain, 185–190. https://doi.org/10.1109/BIBM.2018.8621502

[14] David J C MacKay. 2003. *Information Theory, Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, UK.

[15] David G. Messerschmitt. 1971. Quantizing for maximum output entropy (Corresp.). *IEEE Transactions on Information Theory* 17, 5 (Sep. 1971), 612–612. https://doi.org/10.1109/TIT.1971.1054681

[16] Hien H Nguyen, Susan C Tilton, Christopher J Kemp, and Mingzhou Song. 2017. Nonmonotonic Pathway Gene Expression Analysis Reveals Oncogenic Role of p27/Kip1 at Intermediate Dose. *Cancer Informatics* 16 (11 2017), 1176935117740132. https://doi.org/10.1177/1176935117740132

[17] Hien H. Nguyen, Hua Zhong, and Mingzhou Song. 2020. Optimality, Accuracy, and Efficiency of an Exact Functional Test. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. 2683–2689. https://doi.org/10.24963/ijcai.2020/372 Main track.

[18] SD Palmer and M Song. 2009. Quantization of multivariate continuous random variables by sequential dynamic programming. In *Proceedings of the CAHSI Annual Meeting*. 43–46.

[19] Karl Pearson. 1900. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 50, 302 (1900), 157–175.

[20] Dominic SC Raj, Vallabh O Shah, Mehdi Rambod, Csaba P Kovesdy, and Kamyar Kalantar-Zadeh. 2009. Association of soluble endotoxin receptor CD14 and mortality among patients undergoing hemodialysis. *American Journal of Kidney Diseases* 54, 6 (2009), 1062–1071.

[21] William M. Rand. 1971. Objective Criteria for the Evaluation of Clustering Methods. *J. Amer. Statist. Assoc.* 66, 336 (1971), 846–850. http://www.jstor.org/stable/2284239

[22] Michael Rebhan, Vered Chalifa-Caspi, Jaime Prilusky, and Doron Lancet. 1998. GeneCards: a novel functional genomics compendium with automated data mining and query reformulation support. *Bioinformatics* 14, 8 (1998), 656–664.

[23] Joe Song, Hua Zhong, and Haizhou Wang. 2020. *Ckmeans.1d.dp: Optimal, Fast, and Reproducible Univariate Clustering*. R package version 4.3.3. https://CRAN.R-project.org/package=Ckmeans.1d.dp.

[24] Mingzhou Song, Robert M Haralick, and Stéphane Boissinot. 2010. Efficient and exact maximum likelihood quantisation of genomic features using dynamic programming. *International Journal of Data Mining and Bioinformatics* 4, 2 (2010), 123–141. https://doi.org/10.1504/ijdmb.2010.032167

[25] Mingzhou Song and Hua Zhong. 2020. Efficient weighted univariate clustering maps outstanding dysregulated genomic zones in human cancers. *Bioinformatics* (Jul 2020). https://doi.org/10.1093/bioinformatics/btaa613 [Published online ahead of print, 2020 Jul 3].

[26] Marlon Stoeckius, Christoph Hafemeister, William Stephenson, Brian Houck-Loomis, Pratip K Chattopadhyay, Harold Swerdlow, Rahul Satija, and Peter Smibert. 2017. Simultaneous epitope and transcriptome measurement in single cells. *Nature Methods* 14, 9 (2017), 865.

[27] Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2010. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *J. Mach. Learn. Res.* 11 (Dec. 2010), 2837–2854.

[28] Haizhou Wang, Ming Leung, Angela Wandinger-Ness, Laurie G Hudson, and Mingzhou Song. 2016. Constrained inference of protein interaction networks for invadopodium formation in cancer. *IET Systems Biology* 10, 2 (04 2016), 76–85. https://doi.org/10.1049/iet-syb.2015.0009

[29] Haizhou Wang and Mingzhou Song. 2011. Ckmeans.1d.dp: Optimal k-means clustering in one dimension by dynamic programming. *The R Journal* 3, 2 (2011), 29–33. https://doi.org/10.32614/RJ-2011-015

[30] Yang Wang, Chong Liu, Ying Fang, Xiaoli Liu, Wentao Li, Shuqing Liu, Yingyu Liu, Yuxi Liu, Catherine Charreyre, Jean-Christophe Audonnet, et al. 2012. Transcription analysis on response of porcine alveolar macrophages to Haemophilus parasuis. *BMC Genomics* 13, 1 (2012), 68.

[31] Ivan Zanoni, Renato Ostuni, Giusy Capuano, Maddalena Collini, Michele Caccia, Antonella Ellena Ronchi, Marcella Rocchetti, Francesca Mingozzi, Maria Foti, Giuseppe Chirico, et al. 2009. CD14 regulates the dendritic cell life cycle after LPS exposure through NFAT activation. *Nature* 460, 7252 (2009), 264–268.

[32] Yang Zhang and Mingzhou Song. 2013. Deciphering interactions in causal networks without parametric assumptions. *arXiv Molecular Networks* (2013), 1311.2707. arXiv:1311.2707 http://arxiv.org/abs/1311.2707

[33] Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. 2017. Massively parallel digital transcriptional profiling of single cells. *Nature Communications* 8, 1 (2017), 1–12.

[34] Hua Zhong and Mingzhou Song. 2019. Directional association test reveals high-quality putative cancer driver biomarkers including noncoding RNAs. *BMC Med Genomics* 12, Suppl 7 (2019), 129. https://doi.org/10.1186/s12920-019-0565-9

[35] Hua Zhong and Mingzhou Song. 2019. A fast exact functional test for directional association and cancer biology applications. *IEEE/ACM Trans Comput Biol Bioinform* 16, 3 (2019), 818–826. https://doi.org/10.1109/TCBB.2018.2809743