Additional File 1: Dataset generation using G-Bic

This file introduces the use of G-Bic's *user interface* by presenting a use case for generating a hetereogeneous dataset step by step. Each of the following sections shows the different stages of the generator. In the end, the output produced is analyzed. A video tutorial is available here.

Dataset Properties

The first step is to define the set of properties that will characterize our dataset S. This can be done through the *"Dataset Properties"* tab on the interface, as exemplified in Figure 1.

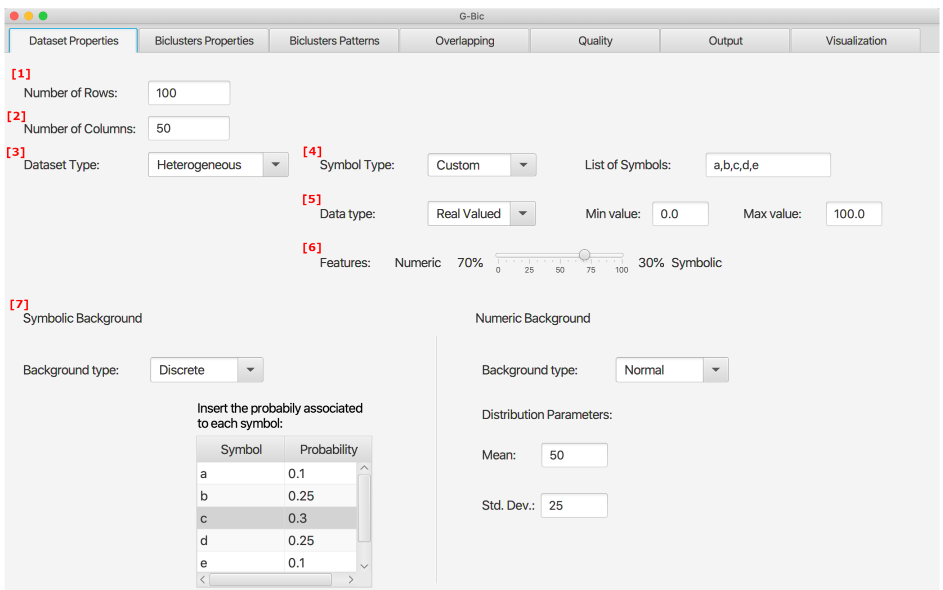


Figure 1 G-Bic: Dataset properties tab.

Dataset S will be composed by 100 Rows and 50 Columns, thus the parameters *Number of Rows* (1) and *Number of Columns* (2). The *Dataset Type* (3) has a hetegeneous nature, there are three features, two to configure the properties of the symbolic (*Symbol Type* (4)) and numeric data (*Data Type* (5)), and another to configure the proportion between symbolic and and numeric data (*Features* (6)).

Regarding the *Symbolic Type* (4), the user must indicate if the alphabet is composed of default symbols, generated automatically, where the user only indicates the alphabet size (Default Symbol Type). Alternatively, if he/she desires a custom alphabet, the list of symbols will be required (Custom Symbol Type). In our simulation, we decided to generate a custom dataset with five custom symbols, a, b, c, d, e.

Considering the *Data Type* (5), the user can choose the numeric properties of the dataset. Currently, both Real Valued and Integer options are available. In both cases, a maximun and minimun range for the number must be selected. In our simulation, we decided to generate real-valued numbers between 0 and 100.

The *Features* slider (6) allows the user to control the proportion between numeric and symbolic features (columns). We generated a dataset consisted of 70% numeric columns.

The last parameter determines how the *background* values (7) of the dataset are distributed: (absence of biclusters): *Uniform*, *Normal*, *Discrete* and *Missing*. If the user chooses *Normal*, or *Discrete*, additional parameters are presented to set the distributions parameters, like the *Mean* and *Standard Deviation* on the *Normal* option, or a table with an editable probability is associated to each symbol, for the *Discrete* one. Since our dataset has a hetegeneous nature, it is needed to define the background for both the symbolic and numeric part. For the discrete, we choose to define the probability for each symbol to be in the symbolic columns. For the numeric background, a normal distribuition was choosen.

Bicluster Properties

The next step defines the amount and the structure of the planted biclusters on the dataset to be generated. The number of biclusters in dataset S can be defined through parameter *Number of biclusters* (1).

The following two sets of parameters define their structure: Row (1)/Column (2) distribution and respective parameters. The user has available two types of distributions: *Normal* and *Uniform*. The interface dynamically adapts the respective parameters to ask for *Mean* and *Standard Deviation* for the first type, and *Min* and *Max* for the second one. For dataset S, its structure follows a uniform distribution, and each bicluster will have a set of rows, columns, and contexts varying between $[6, 8]$ and $[3, 6]$, respectively.

The last parameter, *Contiguity* (4), enables the selection on whether the planted biclusters should be contiguous across the column dimension. In this case, dataset S's biclusters will will not be contiguous.

Figure 2 exemplifies the bicluster's properties tab.

Bicluster Patterns

We now focus the set of patterns that will be expressed by the set of biclusters planted. The number of patterns chosen will be uniformly distributed across the set of bicluster available. For example, if the user sets four patterns, and the dataset has eight biclusters, two biclusters will be assigned to each type.

Dataset S will have every existing pattern following the *Order Preserving*, *Constant*, *Additive* and *Multiplicative* types. As for the *Order Preserving* pattern on contexts, the user is able to select whether the generated temporal pattern can have an arbitrarily number of increases and decreases along time, or follow a monotonically increasing or decreasing pattern. Figure 3 exemplifies the bicluster's pattern tab.

Overlapping

The *Overlapping* tab, shown in Figure 4, allows the user to define the number of biclusters that are allowed to overlap and how their interactions are expressed. This interaction is controled by the first parameter *Plaid Coherency* (1), that makes available the two options: *None* and *No Overlapping*. For dataset S the *None* plaid coherency will be chosen.

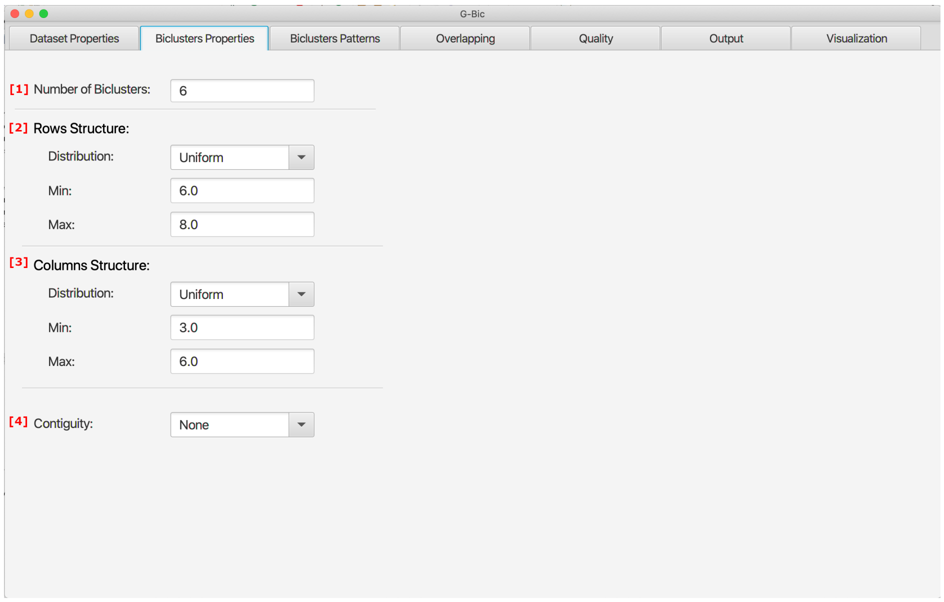


Figure 2 G-Bic: Bicluster properties tab.

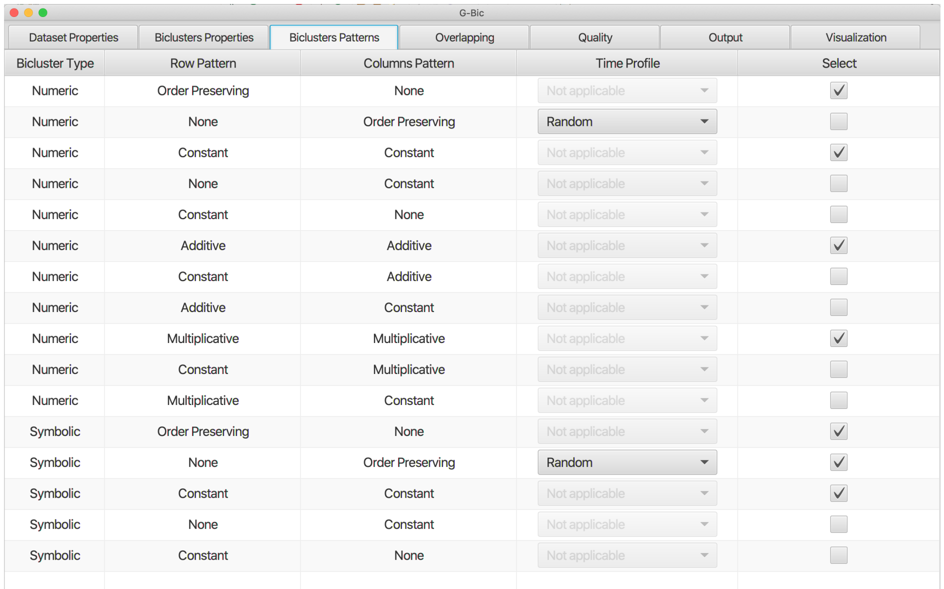
The second step is to set the amount planted biclusters that can overlap. This is done through parameter *% of Overlapping Biclusters* (2). For dataset S, this parameter will be set to 50%.

Then the user has to define how the overlapped biclusters will interact with each other. This is done, first, by defining the maximum number of subspaces that can overlap simultaneously, using the parameter *Maximum Number of Biclustering Interactions* (3). Then the user defines how many elements two overlapped biclusters can share, using parameter *% of Overlapping Elements Between Biclusters* (4). Each bicluster on dataset S can overlap with another one, so the number of simultaneous interactions is 2. A set of biclusters can also share 40% of the smallest bicluster's elements. The last three parameters allow the introduction of restrictions on the number of rows (5) and columns (6) that can be shared by a set of overlapping biclusters. We decided to set the % of overlapping rows to 100% and the % of overlapping columns to 80%.

Quality

The *Quality* tab, illustrated in Figure 5, controls properties from the dataset and the biclusters. Here the user can define the amount of missing values, noise, and errors on both dataset's background and planted biclusters.

For dataset S, the *% of Missing Values on Background* (1) is set to 2% percent, while the *% of Missing Values on Planted Biclusters* (2) is 3%. For noise, the *% of Noise on Background* (3) is 10% and the *% of Noise on Planted Biclusters* (4) is 5%. The *Noise Deviation* (5) is set to 1. This means that the noisy value will be, at maximum, at a distance of 1 from the original value. The last setting defines the proportion of errors on the dataset. The *% of Errors on Background* (6) and the *%*



G-Bic

Dataset Properties | Biclusters Properties | Biclusters Patterns | Overlapping | Quality | Output | Visualization

Bicluster Type	Row Pattern	Columns Pattern	Time Profile	Select
Numeric	Order Preserving	None	Not applicable	✓
Numeric	None	Order Preserving	Random	
Numeric	Constant	Constant	Not applicable	✓
Numeric	None	Constant	Not applicable	
Numeric	Constant	None	Not applicable	
Numeric	Additive	Additive	Not applicable	✓
Numeric	Constant	Additive	Not applicable	
Numeric	Additive	Constant	Not applicable	
Numeric	Multiplicative	Multiplicative	Not applicable	✓
Numeric	Constant	Multiplicative	Not applicable	
Numeric	Multiplicative	Constant	Not applicable	
Symbolic	Order Preserving	None	Not applicable	✓
Symbolic	None	Order Preserving	Random	✓
Symbolic	Constant	Constant	Not applicable	✓
Symbolic	None	Constant	Not applicable	
Symbolic	Constant	None	Not applicable	

Figure 3 G-Bic: Bicluster patterns tab.

of Errors on Planted Biclusters is set to 1%. The error elements will be at a distance from the original values of at least the value of *Noise Deviation* (5). Parameters (1), (3), and (6) control the exact amount of missing values, noise, and errors in the background.

Output

The last stage before generating the new dataset is defining how and where the output will be stored, as resumed in Figure 6. The first parameter, *Save On* (1) allows the user to decide whether the dataset should be stored on a single or onyo multiple files. Multiple files are worth it when the dataset has large dimensions, since it can be divided in small chunks across several files. The second parameter, *File Name* (2), sets the prefix of the name of all three output files. The first file will contain the dataset in a *tsv* format, with the values separated by a tab delimiter, as shown in Figure 7. The remaining two files will contain the information about the biclusters planted on either *txt* format, illustrated in Figure 8, where some statistics and the summary of the first bicluster, as well as the content for the first context is shown; and also by a *JSON* format, as shown in Figure 9. The last parameter, *Save to Directory* (3), specifies where the output will be stored.

Visualization

The last tab of the application allows the user to visualize the output, by showing the biiclusters that resulted from the generation process. Figure 10 shows the visualization options. This tab is composed by two sections: 1) One with the information regarding the bicluster's structure, and 2) one with a graphical representation of the bicluster.

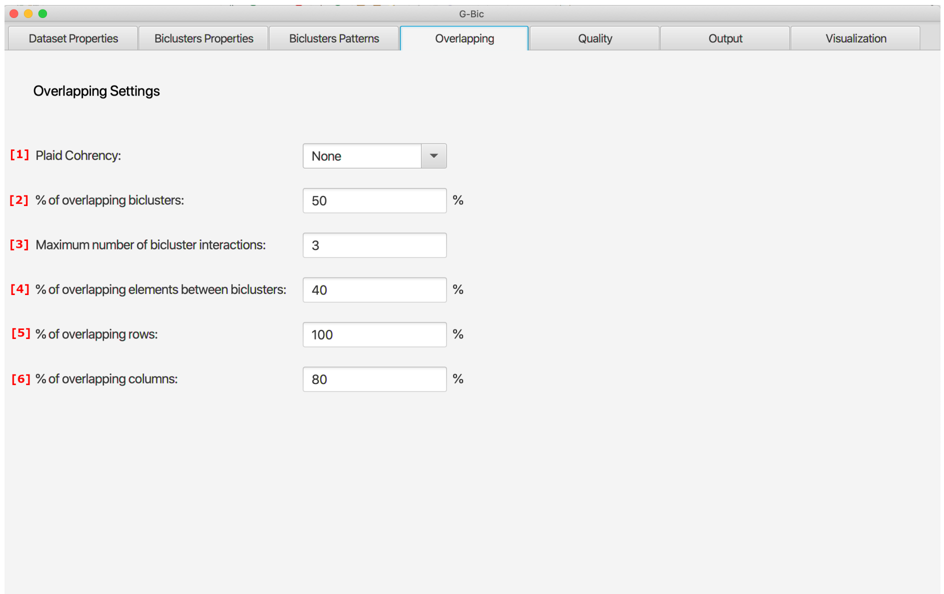


Figure 4 G-Bic: Overlapping settings tab.

As the user chooses one of the available biclusters, the left section of the interface shows information that describes the planted subspace, such as, its dimensions, where it is located (on which rows and columns), which are the patterns followed by each dimension, and respective factors, when available (only in additive or multiplicative patterns), the plaid coherency assumed and the degree of missing values, noise and errors.

In the right section the user can visualize the values of each context through a new windows that displays a graphical representation of the slice using a heatmap, that easily reflects the pattern expressed, as shown in Figure 11.

Simulating real data

In this section, we show how G-Bic can be used to produce reference biclusters. We focus on the most popular areas of application of biclustering: gene expression data, text mining, recommendation systems and biomedical data. Example datasets that we reproduced are in 2. To characterize the biclusters, we complement the definitions by Madeira and Oliveira [1] with empirical conclusions by several authors, [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 7, 8, 13, 14, 15], when applying their biclustering algorithms to the chosen datasets, summarized in Table 3.

Table 2 Example datasets used during the development of some of the existing algorithms

	Dataset Type	Dataset	Description	Dimensions
1	Gene Expression	Arabidopsis [6]	Genes × Conditions	21031 × 351
2	Recomendation Systems	MovieLens-20M [7, 8]	Users × Movies	138000 × 27000
3	Text Mining	Reuters-21578 [12]	Terms × Documents	29930 × 21578
4	Clinical Data	PMSI2013[14]	Patients × Clinical Data	49231 × 7941

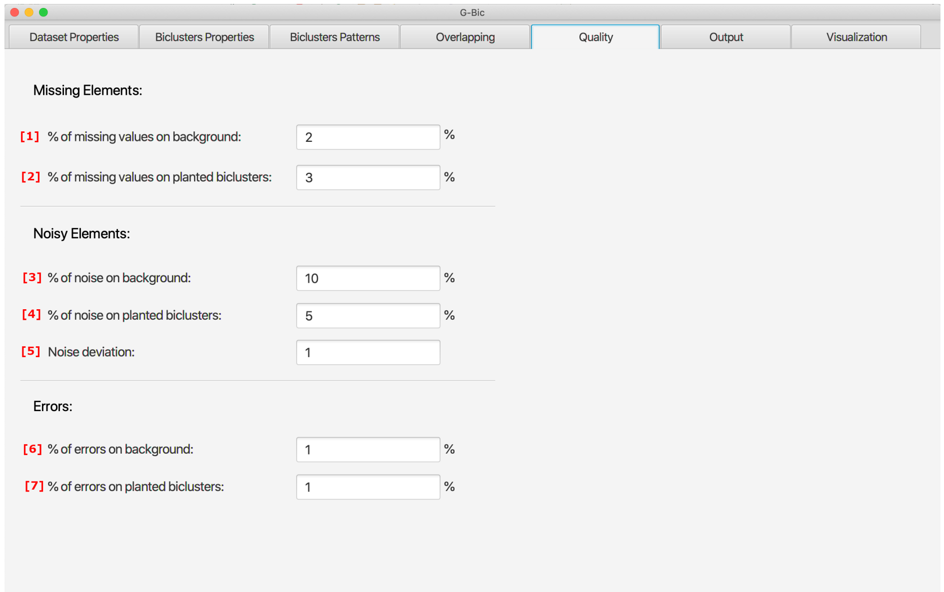


Figure 5 G-Bic: Quality settings tab.

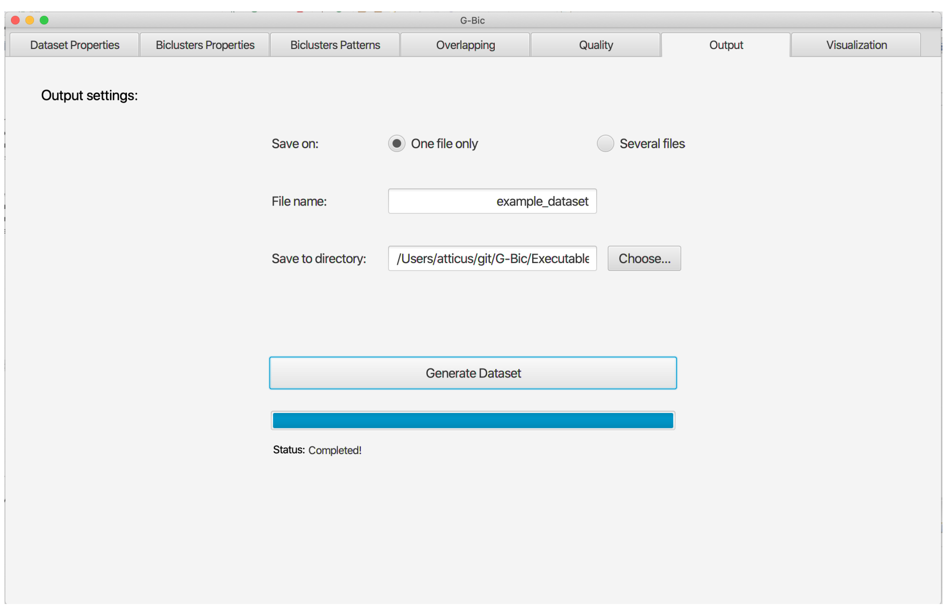


Figure 6 G-Bic: Output tab.

y20	y21	y22	y23	y24	y25	y26	y27	y28	y29
58.53	0	28.32		26.09	b		c	b	c
32.97	96.73	79.96	100	43.95	b	a	d	a	e
0	78.01	88.47	62.61		e	a	b		d
6.13	23.33	39.19	100	89.9	a	d	e	a	c
25.81	76.39	60.56	34.16	82.59	e	c	c	e	d
0	38.62	61.02		89.9	e	a	d	c	b
0	100	47.39	84.22	36.43	c		c	a	d
	45.26	44.66	21.3	25.01	d	a	b	c	e
79.52	14.31		67.58	0	d	c	b	d	a
100	73.54	100	5.32	0	d	e	c	c	b
100	38.74	53.5	43.4	33.49	e	d	b		e
23.52	74.85	0	18.17	36.74	b	a	b	a	e
27.76	20.18	51.29	61.78	33.53	c	b	c	d	e
50.58	58.66	41.1	0	96.01	c	e	c		d
25.16	68.93	53.97	29.85	29.58	c	d	c	c	c
55.8	0	58.03	100	34.25	d	d	b	c	c
51.73	48.9	100	59.82	68.59	e		a	c	d
64.39	67.51	50.46	18.1	70.71	c	d	d	e	d
21.85	57.94	65.73	0	56.7	c	b	b		
91.91	32.79	83.52	47.48	86.54	c	d	d	a	c
86.29	31.69	14.73	52.44	54.48	c	a	d	c	d
42.01	63.45	31.51	46.59	79.38	d		c	c	b
65.12	100	100	25.12	32.91	d	c	d	b	a

Figure 7 G-Bic: Dataset tsv file.

```
Total of planted biclusters: 6
Number of planted numeric biclusters: 2
Number of planted symbolic biclusters: 2
Number of planted mixed biclusters: 2
Bicluster coverage: 4.04%
Missing values on dataset: 9.719999999999999%
Noise values on dataset: 9.68%
Errors on dataset: 9.74%


*** Numeric Biclusters ***

Bicluster #2 (6, 5)
Rows=[2,20,24,39,74,80], Columns=[1,2,6,15,22], RowPattern=Additive, ColumnPattern=Additive, Seed=86.41,  RowFactors=[5.42,-68.32,-34.19,-57.53,12.67,-20.19], ColumnFactors=[-8.2,-1.93,-3.66,-10.08,-3.36], %Missings=3.33, %Noise=0, %Erro

X       y1      y2      y6      y15     y22
x2      83.63   89.9    88.17   81.75   88.47
x20     9.89    16.16   14.43   8.01    14.73
x24     44.01   50.28   48.56   42.13   48.86
x39     20.68   26.95   25.22   18.8    25.52
x74     90.88   97.15   95.42   88.99   95.72
x80     58.01   64.28   62.56   56.13

*** Symbolic Biclusters ***

Bicluster #0 (7, 5)
Rows=[25,47,49,72,81,91,98], Columns=[26,28,33,42,49], RowPattern=None, ColumnPattern=OrderPreserving, TimeProfile=Random, %Missings=2.86, %Noise=0, %Errors=5.71 PlaidCoherency=None

X       y26     y28     y33     y42     y49
x25     c       a       b               e
x47     a       b       a       e       b
x49     c       a       a       e       e
x72     c       b       a       e       e
x81     d       b       b       e       d
x91     b       a       a       e       d
x98     c       c       b       d       c
```

Figure 8 G-Bic: Bicluster's txt file.

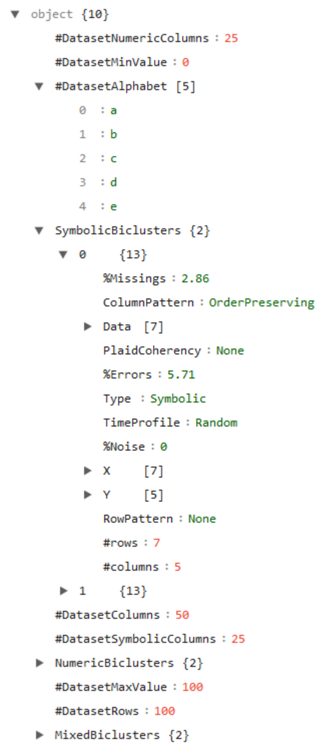


Figure 9 G-Bic: Bicluster's JSON file.

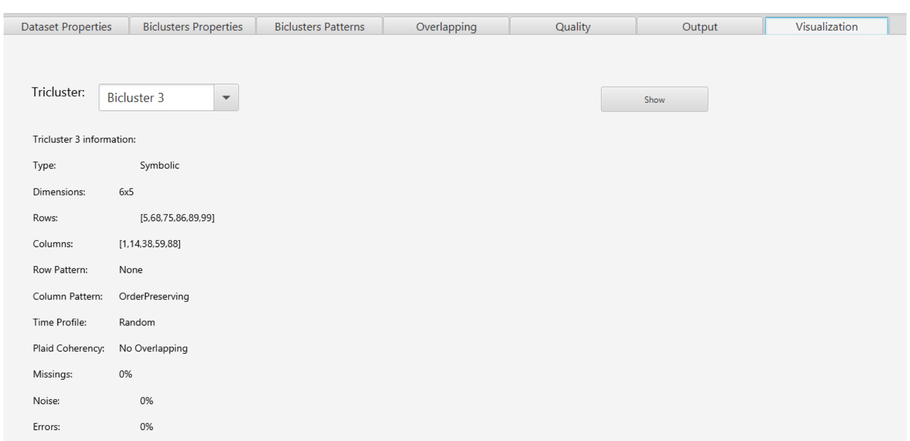


Figure 10 G-Bic: Bicluster's summary

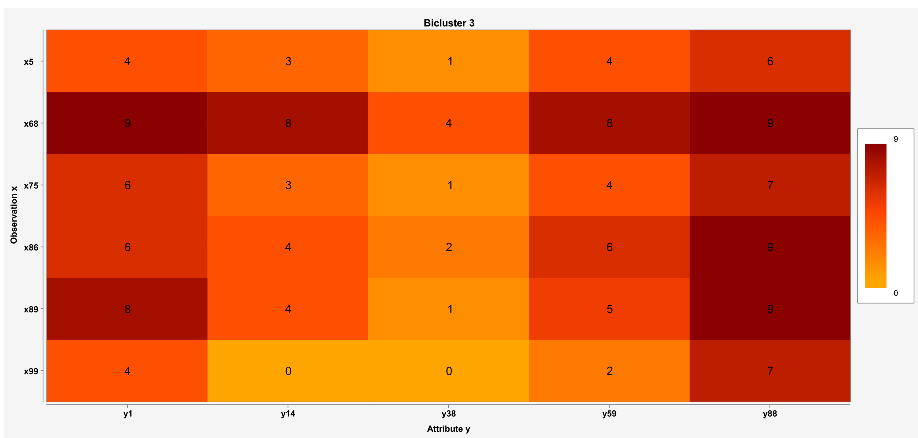


Figure 11 G-Bic: Heatmap to visualize a bicluster's pattern

Table 3 Settings to simulate the real datasets

	Properties	Dataset 1 - Gene Expression	Dataset 2 - Recomendation Systems	Dataset 3 - Text Mining	Dataset 4 - Clinical Data
Dataset	Data type	Numeric	Numeric	Numeric	Mixed
	Dimensions	21031×351	138000×27000	29930×21578	49231×7941
	Background	Uniform	Missing	Missing	Missing
	Missings	0%	$95,5\%$	98%	$99,81\%$
	Noise	10%	0%	0%	0%
	Errors	0%	0%	0%	3%
Biclusters	Number	417	3000	30	70
	Patterns	Additive, Order Preserving	Order Preserving	Constant and Order-Preserving	Mixed
Overlapping	Plaid Coherency	Additive	No Overlapping	No Overlapping	No Overlapping
	% Overlapping Bics	10%	0%	0%	0%

References

1. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics **1**(1), 24–45 (2004). doi:10.1109/TCBB.2004.2. Accessed 2021-05-17
2. Henriques, R., Antunes, C., Madeira, S.C.: A structured view on pattern mining-based biclustering. Pattern Recognition **48**(12), 3941–3958 (2015). doi:10.1016/j.patcog.2015.06.018
3. Eren, K., Deveci, M., Küçüktunç, O., Çatalyürek, V.: A comparative analysis of biclustering algorithms for gene expression data. Briefings in Bioinformatics **14**(3), 279–292 (2012). doi:10.1093/bib/bbs032. http://oup.prod.sis.lan/bib/article-pdf/14/3/279/709239/bbs032.pdf
4. Padilha, V.A., Campello, R.J.G.B.: A systematic comparative evaluation of biclustering techniques. BMC Bioinformatics **18**(1) (2017). doi:10.1186/s12859-017-1487-1
5. Xie, J., Ma, A., Fennell, A., Ma, Q., Zhao, J.: It is time to apply biclustering: a comprehensive review of biclustering applications in biological and biomedical data. Briefings in bioinformatics **20** (2018). doi:10.1093/bib/bby014
6. Wang, S., Yin, Y., Ma, Q., Tang, X., Hao, D., Xu, Y.: Genome-scale identification of cell-wall related genes in Arabidopsis based on co-expression network analysis. BMC Plant Biology **12**(1), 138 (2012). doi:10.1186/1471-2229-12-138. Accessed 2022-01-14
7. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P.: BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 1441–1450. ACM, Beijing China (2019). doi:10.1145/3357384.3357895. https://dl.acm.org/doi/10.1145/3357384.3357895 Accessed 2022-01-14
8. de Castro, P.A.D., de Franca, F.O., Ferreira, H.M., Von Zuben, F.J.: Applying Biclustering to Perform Collaborative Filtering. In: Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007), pp. 421–426. IEEE, Rio de Janeiro, Brazil (2007). doi:10.1109/ISDA.2007.91. https://ieeexplore.ieee.org/document/4389645/ Accessed 2022-01-14
9. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '01, pp. 269–274. ACM Press, San Francisco, California (2001). doi:10.1145/502512.502550. http://portal.acm.org/citation.cfm?doid=502512.502550 Accessed 2021-08-06
10. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '03, p. 89. ACM Press, Washington, D.C. (2003). doi:10.1145/956750.956764. http://portal.acm.org/citation.cfm?doid=956750.956764 Accessed 2022-01-14
11. Busygin, S., Prokopyev, O., Pardalos, P.M.: Biclustering in data mining. Computers Operations Research **35**(9), 2964–2987 (2008). doi:10.1016/j.cor.2007.01.005
12. Mimaroglu, S., Uehara, K.: Bit Sequences and Biclustering of Text Documents. In: Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007), pp. 51–56. IEEE, Omaha, NE, USA (2007). doi:10.1109/ICDMW.2007.38. http://ieeexplore.ieee.org/document/4476646/ Accessed 2022-01-14
13. Vandromme, M., Jacques, J., Taillard, J., Jourdan, L., Dhaenens, C.: A Scalable Biclustering Method for Heterogeneous Medical Data. In: Pardalos, P.M., Conca, P., Giuffrida, G., Nicosia, G. (eds.) Machine Learning, Optimization, and Big Data vol. 10122, pp. 70–81. Springer, Cham (2016). doi:10.1007/978-3-319-51469-7$_6$. Series Title: Lecture Notes in Computer Science. http://link.springer.com/10.1007/978-3-319-51469-7$_6 Accessed 2022-01-02$
14. Vandromme, M., Jacques, J., Taillard, J., Jourdan, L., Dhaenens, C.: A Biclustering Method for Heterogeneous and Temporal Medical Data. IEEE Transactions on Knowledge and Data Engineering, 1–1 (2020). doi:10.1109/TKDE.2020.2983692. Accessed 2021-08-07
15. Hong, M.K.H., Yao, H.H.I., Pedersen, J.S., Peters, J.S., Costello, A.J., Murphy, D.G., Hovens, C.M., Corcoran, N.M.: Error rates in a clinical data repository: lessons from the transition to electronic data transfer—a descriptive study. BMJ Open **3**(5), 002406 (2013). doi:10.1136/bmjopen-2012-002406. Accessed 2022-01-14