Target Gene Mining Algorithm Based on gSpan

Liangfu Lu[1], Xiaoxu Ren[1], Lianyong Qi[2]*, Chenming Cui[3], Yichen Jiao[3]

[1] School of Mathematics, Tianjin University, Tianjin, China;
[2] School of Information Science and Engineering, Qufu Normal University, Rizhao, China;
[3] School of Software, Tianjin University, Tianjin, China;
*Corresponding Author: lianyongqi@gmail.com

Abstract. In recent years, the focus of bioinformatics research has turned to biological data processing and information extraction. New mining algorithm was designed to mine target gene fragment efficiently from a huge amount of gene data and to study specific gene expression in this paper. The extracted gene data was filtered in order to remove redundant gene data. Then the binary tree was constructed according to the Pearson correlation coefficient between gene data and processed by gSpan frequent subgraph mining algorithm. Finally, the results were visually analyzed in grayscale image way which helped us to find out the target gene. Compared with the existing target gene mining algorithms, such as integrated decision feature gene selection algorithm, our approach enjoys the advantages of higher accuracy and processing high-dimensional data. The proposed algorithm has sufficient theoretical basis, not only makes the results more efficient, but also makes the possibility of error results less. Moreover, the dimension of the data is much higher than the dimension of the data set used by the existing algorithm, so the algorithm is more practical.

Keywords: GSpan gene mining algorithm, Gene expression data, Data mining, Visual analysis.

1 Introduction

1.1 A Subsection Sample

With the rapid development of high-throughput technology, various types of biology research mass data have been produced. Bioinformatics and computational biology have been developing corresponding theories and technologies to analyze the information. Moreover, as the focus of human genome research shifts to functional genome, the emphasis of bioinformatics research has quietly turned from the accumulation of biological data to the processing and information extraction of biological data. And it has become an urgent problem to be solved in [1]. Similarly, it is still a difficult problem to understand and explain complex life phenomena. The process of life activity and the factors involved in it are a complex network system. The study of biological networks is a key to understanding complex life activities [2]. Among the various types of networks, of special relevance are collaborative networks. Collaborative networks have

been used in many cases to develop and implement software in different domains that must jointly identify problems and provide solutions [3]. And collaborative networks have also been widely used in manufacturing successfully [4]. In this paper, we use the cooperative mutual information between two sets to calculate the correlation between gene data. And we designed a target gene mining algorithm based on gSpan to solve above problems. The data used in the experiment was divided into two parts, the experiment group and the test group. Data cleaning algorithm was applied to the all the data for primary dimensionality reduction. Then, the binary tree was processed by gSpan frequent subgraph mining algorithm after calculating the Pearson correlation coefficient between different data samples, which was constructed by the filtered data set. Finally, target gene was analyzed with the support of visual methods, which were included line chart and grayscale image and other algorithms.

2 Target gene mining algorithm based on gSpan

2.1 Gene data sources and features

When extracting the gene data used in the paper, we compare the gene probe on a gene probe chip with a corresponding gene fragment of the sample, and we can obtain a value representing the difference between the gene probe and the gene fragment. And taking a logarithm of 2 as the base, a gene data can be received.

Table 1 is a sample of a gene dataset. Where the numbers in the first row represent the number of that data. In the following, each row represents the gene data measured with a specific gene probe chip for each gene sample fragment. The first column in the table shows the names of the gene probes, such as “TC01003440.hg.1” and “TC01003573.hg.1”.

Table 1. An example of gene data sets.

Number of Samples	1	2	3	4	5	6
TC01003440.hg.1	3.773364	4.266670	4.701382	4.891576	4.373340	3.786559
TC01003573.hg.1	1.979416	2.225050	2.472456	1.904527	2.572532	2.624649
TC01003581.hg.1	2.411289	2.630495	3.164683	2.468128	1.821613	0.894983
TC01003634.hg.1	1.023536	4.366745	1.049543	1.032603	0.694951	0.805963
TC01003635.hg.1	3.908017	4.654112	4.519364	3.662154	3.416480	3.878956
TC01003707.hg.1	4.680764	4.819676	5.480835	5.273236	4.406693	4.900424
TC01003855.hg.1	2.900651	4.433520	4.448808	3.992489	4.582560	3.164090
TC01003992.hg.1	3.537443	4.958187	4.264466	3.472608	3.246787	3.903415
TC01005205.hg.1	3.519913	3.387307	4.837730	4.734122	4.124148	3.688145
TC01005809.hg.1	3.091624	3.081775	4.138366	4.133202	3.486166	3.385050
TC02000261.hg.1	4.983322	4.684757	5.213353	5.376272	4.915276	5.093583

According to gene data, they were divided into experimental group and control group. The experimental group were similar in character, while the control group had the opposite character in group.

2.2 Data cleaning based on the overall characteristics of gene samples

Considering that not all the genes in the data sample are target genes, we need to clean the genes to reduce the number of redundant genes [5-7].

Data cleaning between different groups:

Different samples from different groups are thought to have great diversity in gene expressions, which can be measured by variance [8]. For a gene fragment, the variance among all the samples is:

$$\sigma^2 = \frac{\sum(X-\mu)^2}{N} \tag{1}$$

Where σ is the variance and X represents the number of the amount of all these gene fragments in a sample. μ denotes the mean and N is the total sample number.

Data cleaning in every group:

In terms of gene expression for studying traits, there is a small difference between the same set of samples. Similar to group cleaning, in this part, variance can also be used to measure differences in groups. Variance can also be used to measure differences within a group. In this paper, we set a threshold, if the variance is greater than the threshold gene fragment, which indicates that the gene fragment is not similar to the same group of samples, and it does not conform to the principle of small differences between the target gene fragment within the group sample. So these gene fragments are cleaned from all the gene fragments.

Gene filtering:

The threshold was set to the result of above two data cleaning. And those gene fragments of low variance were filtered, which means these gene fragments were similar for all the samples and didn't show different gene expressions between different groups. On the contrary, those gene fragments of high variance were filtered.

2.3 Data cleaning based on the characteristics of the specific sample

After preliminary cleaning of genetic data using variance, according to the authenticity of data samples among individuals, a more accurate and detailed method is adopted to sort out the selected data samples.

First, consider that for the same trait, there is a high probability that two of the samples exhibiting the same or similar trait will have the same gene expression pattern when the sample size is large [9]. Thus, Pearson correlation coefficient was calculated in all the couples after coupling the samples in the same group [10-12].

For two random scalars X and Y, the Pearson correlation coefficient between them is:

$$\rho = \frac{cov(X,Y)}{\sqrt{DX}\sqrt{DY}}, \tag{2}$$

$$cov(X,Y) = E\big(\big(X - E(X)\big) \cdot (Y - E(Y))\big). \tag{3}$$

In this case, genes with high similarity in low correlation combinations are more likely to contain target genes in [13]. And the union of similar gene fragments in every

couple was included in the set of gene fragments to construct the graph. Similarly, considering the nature between two different groups of sample individuals, paired combinations of data samples from all different groups with high correlation can highlight the target genes with low similarity. And we select the more dissimilar gene fragments from these combinations and merge them as another part of the final composition of the gene fragment set [14].

For two random variables, mutual information can be seen as reducing the uncertainty of one variable by knowing the value of another variable. The average mutual information is calculated by

$$I(\mathrm{X};\mathrm{Y}) = H(X) + H(Y) - H(XY) \tag{4}$$

$$H(X) = -\sum_{p(x)} p(x) \log p(x) \tag{5}$$

$$H(XY) = -\sum_{x}\sum_{y} p(xy) \log p(xy) \tag{6}$$

where $I(X;Y)$ represents the average mutual information of random variables X and Y, which are gene data of two samples. The $H(X)$ and $H(Y)$ represent the entropies of X and Y. The $H(XY)$ is the joint entropy of X and Y. The $p(x)$ represents the probability distribution of X. The $p(xy)$ is the joint probability distribution of X and Y.

3 The construction of the graph

3.1 Basis of construction

For a dataset that has n samples and the selected sample is P, GA and GB are the two gene fragments needed to be calculated. The mutual information $MI(n)$ of GA and GB and their corresponding gene data $GA(n)$ and $GB(n)$, was calculated firstly. Then, the gene data of sample P on gene fragments GA and GB were removed from $GA(n)$ and $GB(n)$, getting gene data $GA(n-1)$ and $GB(n-1)$ and calculate the mutual information $MI\ (n-1)$ between them. When the mutual information $MI(n)$ is higher than $MI(n-1)$, which means that the gene data added to the sample P lead to the increase of the amount of mutual information between $GA(n)$ and $GB(n)$ and also means the gene data GA of samples has close relationship with the gene data GB of sample P.

Based on this, MIP, which is calculated by the difference between *MI(n-1)* and $MI(n)$, is used as a criterion to measure the correlation between gene data of sample P on GA and GB. The larger the MIP, the greater the correlation between the two gene fragments. And the smaller the MIP, the smaller the correlation between the two gene fragments.

3.2 Construction procedure

To simplify the calculation, gene data was constructed to be a binary tree. The fragment that has the smallest variance in the group after data cleaning was placed as the root [15]. The root of a node's left subtree was the most correlative gene fragment to the node, and the root of the node's right subtree was the second most correlative gene fragment to the node [16]. All the gene fragments were inserted in the binary tree based on breadth-first principle.

4 gSpan algorithm

The gSpan algorithm is the most widely used by the subgraph mining algorithm in the world, which was proposed by Yan and Han in [17].

4.1 Definition

Frequent subgraph: given a graph set $D = [\mathrm{G}_1, \mathrm{G}_2, \cdots, \mathrm{G}_\mathrm{n}]$ and a graph g, The number of the graph g included in set D is called G's support and recorded as support(g). For a given minimum threshold minSup, if $support(g) \geq minSup$, g is called a frequent subgraph of D. DFS Lexicographic Order: let $Z = \{code(G,T)|T\ is\ a\ DFS\ tree\ of\ graph\ G\}$. Assuming there is a linear sequence Q in a label set, then the lexicographic combination of T and Q is a linear sequence in set $ET \times L \times L \times L$.

The definition of DFS lexicographic order is that for $a = code(G_a, T_a) = (a_0, a_1, \ldots, a_n)$, $b = code(G_b, T_b) = (b_0, b_1, \ldots, b_n)$ and a, b belong to Z, we conclude $a \leq b$ if and only if the following condition is true:

(1) there exists a t, $0 \leq t \leq min(m,n)$, $a_k = b_k$, $k < t$, the lexicography combination of a_t and b_t is the mentioned above linear sequence.

(2) $a_k = b_k$, $0 \leq k \leq m$ and $m \leq n$.

DFS code: gSpan algorithm uses five parameters to code the edge in the graph by the way like $(i, j, l_i, l_{(i,j)}, l_j)$, in which l_i and l_j are the vertexes of edge $l_{(i,j)}$.

Smallest DFS code: for a given graph G, $Z = \{code(G,T)|T\ is\ a\ DFS\ tree\ of\ graph\ G\}$, $min(Z(G))$, according to DFS lexicographic order, which is called the smallest DFS code of G.

DFS code tree: in DFS, each node represents a DFS code, the relationship between the parent node and child node obey the abovementioned parent-child relationship. Relationships between brothers and sisters are consistent with DFS lexicographic order, which means the preorder traversal of DFS code tree obeys DFS lexicographic order.

4.2 The gSpan algorithm

The thinking in the gSpan algorithm is shown in table 2 and table 3 as fake codes.

Table 2. gSpan main program.

Algorithm 1. GraphSet_Projection (D, S).
1: Sort the labels of vertexes and edges in D by frequency;
2: Delete infrequent vertexes and edges;
3: Remark the remaining vertexes and edges
4: Save the frequent edges of D into set S^1;
5: Sort S^1 by DFS lexicographic order；
6: S ← S^1;
7: **for** each edge e that belongs to S^1 **do**;
8. Use e to initialize s, putting all the graphs that include e into set D;

9.Subgraph_Mining (D, S, s);
10.$D \leftarrow D - e$;
11. **if** $|D|$ < minSup;
12.**break;**

Table 3. gSpan subprogram.

Subprogram 1 Subgraph_Mining(D, S, s).
1: **if** $s \neq min(s)$
2: **return;**
3: $S \leftarrow S \cup \{s\}$;
4: Enumerate s in all the graphs that belongs to set D and count the amount of its subgraph;
5: **for each** e which is the subgraph of s **do**
6: **if** $support(e) \geq minSup$;
7: $s \leftarrow e$;
8: Subgraph_Mining(D_s, S, s);

4.3 Description of target gene mining algorithm based on gSpan

The gSpan frequent subgraph mining algorithm was applied to the graph sets which is constructed by experiment group and test group to get their frequent subgraphs. Then the set of target gene fragment was got by analyzing two groups' frequent subgraphs.

5 Visual analysis of gene data

5.1 The decision with visual analysis

An appropriate threshold was needed while doing gene cleaning in every group and between groups. Line chart was used to help us to make the decision quickly and properly by analyzing the result of data cleaning when threshold changed. The inflection points in the threshold's line chart were considered to be possible choices of threshold values based on the characteristic that the inflection point in a line chart always represented the critical state.

5.2 Grayscale image of gene data

Grayscale image was applied to the gene data to make the visual analysis process more convenient and intuitive, which started with mapping all the raw gene data to a grayscale interval that was from 0 to 1. The mapping procedures are as follows:

（1） Selecting the biggest gene data M from all the raw gene data;

（2） Rounding up M to get an integer N;

（3） Dividing all the original gene data by integer N, and get the gray value after mapping.

Table 4 shows some raw gene data and their grayscale values after mapping.

Table 4. Examples of mapping raw gene data to grayscale value.

Raw gene data	Grayscale values
3.773364	0.7546728
1.979416	0.3958832
2.411289	0.4822578
1.023536	0.2047072
3.908017	0.7816034
4.680764	0.9361528
2.900651	0.5801302
3.537443	0.7074886
3.519913	0.7039826
3.091624	0.6183248
4.983322	0.9966644

Then a single column vector that contained all the grayscale values was inserted into an $N * N$ matrix and N was an integer which was got by rounding up the square root of the data number. The blank areas in the matrix were filled with 0. Table 5 was the matrix filled with the grayscale values in table 4.

Table 5. The example of grayscale value filling.

0.7546728	0.7816034	0.6183248	0
0.3958832	0.9361528	0.9966644	0
0.4822578	0.5801302	0	0
0.2047072	0.7074886	0	0

The grayscale matrix was used as an input value to get a grayscale image. And Fig. 1 was an example of grayscale images.

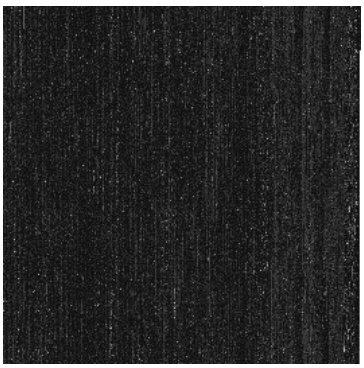

Fig. 1. An example of grayscale images

6 Experiment results and analysis

6.1 Experiment results

Dental caries is one of the most common chronic diseases, and it is easy for individuals to suffer from this disease throughout their lives. In recent years, many researchers have studied the prevention of dental caries and some oral health mechanisms [18-19]. And the gene data used in the experiment was selected from the University of California, Los Angeles, which was extracted to study the pathogenic and resistance gene about dental caries in the human body. And the data consist of 21 samples, which are divided into two groups. One group is of high S.mutans and high caries, abbreviated as HSHC in the following, which has 10 samples. Another group is of high S.mutans and low caries, abbreviated as HSLC in the following, which has 11 samples. Data which were selected from 70523 gene probe chips were included in the raw gene data and each gene probe chip corresponded to a gene fragment.

Some frequent subgraphs are obtained in this experiment by using the gSpan algorithm. Considering that some frequent subgraphs are meaningless due to the default value of the algorithm in the process of constructing gene fragment graphs. After removing these meaningless subgraphs, we can obtain two frequent subgraphs, which are shown as Fig. 2 and Fig. 3. The relationships between nodes in the frequent subgraphs and serial numbers of gene probe chips were listed in table 6. The gene fragments corresponding to the nodes were the result of this experiment.

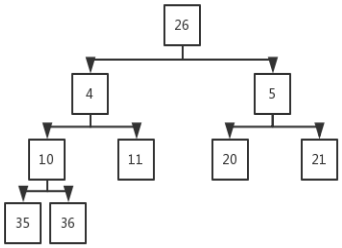

Fig. 2. Frequent subgraph of group HSLC

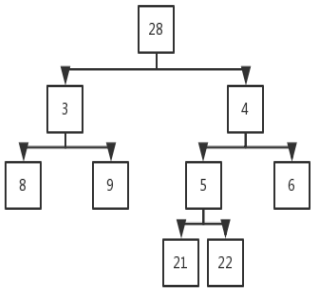

Fig. 3. Frequent subgraph of group HSHC

Table 6. Relationships between nodes in the frequent subgraphs and serial numbers of gene probe chips

Group HSLC		Group HSHC	
Nodes	Serial numbers	Nodes	Serial numbers
4	TC01000938.hg.1	3	TC01000711.hg.1
5	TC01001103.hg.1	4	TC01000938.hg.1
10	TC01001464.hg.1	5	TC01001103.hg.1
11	TC01001809.hg.1	6	TC01001120.hg.1
20	TC01003427.hg.1	8	TC01001143.hg.1
21	TC01003428.hg.1	9	TC01001291.hg.1
26	TC01003573.hg.1	21	TC01003428.hg.1
35	TC02000261.hg.1	22	TC01003431.hg.1
36	TC02000500.hg.1	28	TC01003634.hg.1

After verifying, nodes 4, 5, 21 in group HSLC (e.g. the gene fragments corresponding to gene probe chips number "TC01000938.hg.1" "TC01001103.hg.1" "TC01003428.hg.1"), were the target gene fragments to determine the gene fragment of dental caries in this experiment. Thus, the coverage rate to dental caries was 100% and dental caries gene occupied 33.3% of the results, which meant the experiment finished with good consequence.

6.2 Experiment result analysis

Comparing with existing target gene mining algorithms, there were two advantages of gSpan target gene mining algorithm:

(1) Higher accuracy. Although some existing algorithms also have 100% coverage rate to dental caries, the dental caries gene can only occupy for 15%-20% of the results. By constructing the gene data into graph during data process creatively and using subgraph mining algorithm, which makes the result can be more accurate and target gene can occupy a higher percentage of the result.

(2) Higher data dimensions. Existing algorithms always have dimension under 10,000 and extract some uncommon situations. The dataset used in this experiment came from raw gene data with a dimension of 70523, which is much higher than the dimension used by existing algorithms and is also much more practical.

7 Conclusion

The gSpan target gene mining algorithm was proposed in this paper in order to help mine target gene from a huge amount of gene data. And this paper focused on the following aspects:

(1) We mainly introduced the data cleaning algorithm, graph construction algorithm and gSpan frequent subgraph mining algorithm. And a target gene mining algorithm based on gSpan was proposed on the basis of the algorithms mentioned above.

(2) To verify the effectiveness of algorithm, the gene data about human dental caries extracted by the University of California, Los Angeles was used as data set. According to the experiment, the results' coverage rate to dental caries was 100%, which was the same as existing algorithms. And dental caries gene occupied 33.3% of the results, which was higher than existing algorithms which showed that the new algorithm had the better effect.

(3) Grayscale image and line chart were introduced to help visually analyze the result of the algorithm and make the decision.

And the edges of the graph only had two statuses, existing or not. Thus, the data can be constructed into weighted undirected graphs, in which the weight of each edge represents the correlation between two node genes. And the gSpan algorithm can be replaced by algorithms that can process weighted undirected graphs. Moreover, network models can also be introduced to help us find new ways to do the filtering, construction and target gene finding work [20]. Thus, a lot of work can be done to optimize the target gene mining algorithm in the future.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under No.51877144.

References

1. Michihiro K, George K. Gene classification using expression profiles: a feasibility study[J]. International Journal on Artificial Intelligence Tools, 2001, 14(04):641-660.
2. Lee I, Blom U M, Wang P I, et al. Prioritizing candidate disease genes by network-based boosting of genome-wide association data[J]. Genome Research, 2011, 21(7):1109.
3. Sabau G, Bologa R, Bologa R, et al. Collaborative Network for the Development of an Informational System in the SOA Context for the University Management[C]// International Conference on Computer Technology and Development. IEEE, 2009:307-311.
4. J. Shuman and J. Twombly, "CollaborativeBusiness," in Collaborative Networks Are The Organization : An Innovation in Organization Design and Management, vol. Volume, 8 vols., Newton, USA: The Rhythm of Business, Inc., 2009.
5. Alon U,Barkai N,Nootterman D A,et al.Broad patterns of gene expression revealed by clustering analysis of tum orand normalcolon tisues probed by oligonucleotide arrays[J].Science,1999,96(12):6745-6750.
6. jie Z, Cheng-quan G, Jun-rong C, Li-xin G. Tumor identification based on gene expression profiles and the search about extraction of the feature genes[J]. Mathematics in Practice and Theory. 2011,41(14):67-79.
7. Ya-ning Z, Yan-hui Z. Extraction of Tumor Gene and Its Classification based on SNR[J]. Journa of Xiangfan University. 2011,32(8):13-6.

8. Quan-jin L, Ying-xin L, Xiao-gang R. Cancer information gene identification based on statistical method[J]. Journal of Beijing University of Technology. 2005,31(2):122-5.
9. Yongxiu C. Understanding of correlation coefficient[J]. 2011,(7):15-9.
10. Hong-bin L, Guang-zhong H, Qiu-ting G. Similarity Retrieval Method of Organic Mass Spectrometry Based on the Pearson Correlation Coefficient[J]. Chemical Analysis and Meterage. 2015,24(3):33-7.
11. Niyogi X. Locality preserving projections[C]//Neural information processing systems. 2004,16:153.
12. Yong-chao W. A Novel D-S Combination Method of Conflicting Evidences Based on Pearson Correlation Coefficient[J]. Telecommunication on Engineering. 2012,52(4):466-71.
13. Jie L, Li-jun D, Sheng-nan T. Refinement Procedure for Eigen genes of colon Carcinoma based on BB-SIR[J]. World SCI-Tech R&D. 2011,33(4):588-91.
14. Shoujue W, Lingfei Z. Gene selection for gene expression data analysis[J]. Micro Computer Information. 2008,24(3-3):193-4.
15. Jing-jing S, Li-bo W, Wei L. Gene selection for cancer diagnosis[J]. Computer Engineering adn Applications. 2010:218-20.
16. Jun W. Method of effective DNA microarray data feature extraction[J]. Modern Eclectronics Technique. 2014,37(13):95-8.
17. X Y, J H. gSpan: Graph-based substructure pattern mining[J]. ICDM IEEE. 2002.
18. Lin T H, Lin C H, Pan T M. The implication of probiotics in the prevention of dental caries[J]. Applied Microbiology & Biotechnology, 2018, 102(2):577-586.
19. Philip N, Suneja B, Walsh L J. Ecological Approaches to Dental Caries Prevention: Paradigm Shift or Shibboleth?[J]. Caries Research, 2018, 52(1-2):153-165.
20. Liu H, Bebu I, Li X. Microarray probes and probe sets[J]. Frontiers in Bioscience, 2010, 2(1):325.