BIOINFORMATION
Discovery at the interface of physical and biological sciences

open access www.bioinformation.net Hypothesis

Volume 7(5)

Algorithm for large-scale clustering across multiple genomes

Gangman Yi*, Jaehee Jung

Samsung Electronics Co., Ltd. 416, Maetan 3-dong, Yeongtong-gu, Suwon-si, Gyeonggi-do, 442-742 Korea. Gangman Yi - gangman.yi@gmail.com; *Corresponding author

Received October 16, 2011; Accepted October 24, 2011; Published October 31, 2011

Abstract:
Identifying genomic regions that descended from a common ancestor helps us study the gene function and genome evolution. In distantly related genomes, clusters of homologous gene pairs are evidently used in function prediction, operon detection, etc. Currently, there are many kinds of computational methods that have been proposed defining gene clusters to identify gene families and operons. However, most of those algorithms are only available on a data set of small size. We developed an efficient gene clustering algorithm that can be applied on hundreds of genomes at the same time. This approach allows for large-scale study of evolutionary relationships of gene clusters and study of operon formation and destruction. An analysis of proposed algorithms shows that more biological insight can be obtained by analyzing gene clusters across hundreds of genomes, which can help us understand operon occurrences, gene orientations and gene rearrangements.

Background:
In bacteria, one of the main mechanisms to facilitate control of gene expression is the organization of genes into operons **[1, 2, 3, 4].** The most popular approaches require that the distance between adjacent genes in a cluster to be small **[5, 6, 7, 8],** because homologous regions are straightforward when genomic regions are closely related **[9].** However, it has been shown that there are considerable difficulties to develop efficient algorithms when paralogous genes are allowed **[6, 10].** Several clustering methods have been developed to find gene clusters that have common functions in different genomes. GeneTeams **[5,7]** require that the distance between adjacent genes in a cluster to be small. It allows intervening genes that appear consecutively, possibly in different orders, between genes in a cluster so that various cluster sizes can be adopted. But it includes a restriction that each gene has at most one occurrence in each chromosome. This limits the model as many genomes contain multigene families. To overcome the problem, HomologyTeams **[6]** is a generalized version of GeneTeams that does not require a cluster to appear in every chromosome. It provides statistical significant estimates of identified gene clusters and a fast way to identify sets of gene clusters. Within the gene clusters, neither the order of the genes nor their orientation needs to be conserved, but a fixed threshold of the distance between adjacent genes limits the model **[11].** DomainTeams **[12]** is a modified version of GeneTeams that considers chromosomal regions of conserved protein domains as domain teams rather than each gene as basic unit, while other algorithms allow multiple genomes and use a gene proximity parameter that restricts the number of intervening genes between adjacent genes in a cluster.

While many of these algorithms can be used to perform gene clustering across two or more genomes, very few algorithms are efficient enough to analyze a large number of genomes **[13].** By placing a stricter limit on the maximum cluster size, we observe that efficient algorithms can be developed to perform gene clustering on hundreds of genomes at the same time. This strategy is sufficient for analyzing gene clustering in bacterial genomes, and allows for large-scale study of evolutionary relationships of gene clusters and study of operon formation and destruction. We develop a different formulation based on constraining the overall size of a cluster and develop statistical estimations that allow direct comparisons of clusters of

different sizes. It also allows the study of whether gene clusters in bacteria occur only at the operon level or whether there are higher-level structures, and the functional relationships between them. This algorithm performs seven hundred bacteria data sets which contain a well-characterized list of operons and perform comparative analysis of operon occurrences, gene orientations, and rearrangements both within and across clusters.

Methodology:

We investigate the window-based strategy to analyze hundreds of genomes simultaneously by placing a stricter limit on the maximum cluster size. We allow genes in a same paralogous group to appear in more than one gene on a same chromosome. This makes it possible to avoid the combinatorial explosion of intersecting all combinations of up to k windows at a time with one from each chromosome, and allow the use of a different strategy to find the clusters.

Let $C = \{c_1, \ldots, c_k\}$ be a set of k chromosomes, one from each genome under consideration. We represent each gene by a number, while ignoring its orientation, so that the same number represents a set of orthologous genes in different genomes and each chromosome c_i is represented by a sequence of gene numbers. The same gene number is allowed to appear more than once within each c_i which represents paralogous copies of a gene.

We consider each window of length l within each chromosome and enumerate each of the 2^l subsets of genes within the window. We think of each subset S as a potential gene cluster and identify the subset of chromosomes in which all genes in S appear within a window of length l (See Figure 1 & 2). A subset S that appears in k' chromosomes is represented by k lists $P_1', \ldots, P_k'$, with k' of these lists non-empty and each list P_i' containing a set of positions on c_i, which together specify all the positions of genes that are in S. To ensure that all genes in S are clustered within a region of size at most l on each chromosome c_i for which P_i' is non-empty, we require that each gene number that appears in G' must appear at least once in each of the k' non-empty lists P_i', and within each non-empty P_i', $|r - s| < l$ for any pair of positions $r, s \in P_i'$.

To obtain statistical significance estimates, the p-value of S is estimated by the probability of S appearing in at least this many chromosomes **(equation 1, see supplementary material)**. Let n be the average length of the chromosomes c_i and l' be the average size of the non-empty lists P_i'. The probability of S appearing in a given chromosome can be obtained by assuming a random background distribution based on n, l and l' where n is the average number of genes in a genome, l is the window size, and l' is the number of genes selected by subregion S.

We estimate the p-value of S **(equation 2, see supplementary material)** by the probability of S appearing in at least k' out of k chromosomes spanning at most l in each case using the binomial distribution, and obtain an e-value from this p-value, where l' genes are observed in windows of length at most l in k' out of k chromosomes as shown in **equation 3 (see supplementary material)**.

Since all possible combinations of included genes and intervening genes are included within each window, this algorithm will not lose any clusters that satisfy the definition. One important advantage of the algorithm is that its time complexity grows linearly with the input size and the base of 2 in the exponential part of the time complexity is small, thus a large number of genomes can be considered at the same time.

```
Algorithm largescale_clustering({c1,...,ck},l,n,m,e){
    X ← empty;
    for s ← 1 to k do {
        for each windows w_l of length l on chromosome c_s do {
            G'' ← set of gene numbers in w_l;
            for each combination B of G'', where |B| ≥ 2, do {
                G'' ← empty;
                for i ← to k, where i ≠ s, do {
                    P_i ← set of positions on chromosome c_i, in which gene numbers in B appear;
                    for j ← 1 to |P_i| do {
                        w_j ← window of length l starting from j^th position in P_i on c_i;
                        G''_ij ← set of gene numbers in w_j that must appear at least once in B; }}
            if e-value(B) that appears in G'' with n,l,l' ≤ e then {
                add B to X;
            }}}
```

Figure 1: Algorithm to identify gene clusters that include same gene numbers at least once within a window of length l by the combination of unique gene numbers not to lose potential gene clusters. Paralogous genes are allowed while requiring that each cluster appears in each of the k given chromosomes. X is the set of clusters returned with each cluster B represented by a list of gene numbers. l, n, m, e is the window length, the average length of chromosomes, the minimum number of chromosomes to appear, and e-value cutoff respectively.

BIOINFORMATION
Discovery at the interface of physical and biological sciences

open access www.bioinformation.net Hypothesis

Volume 7(5)

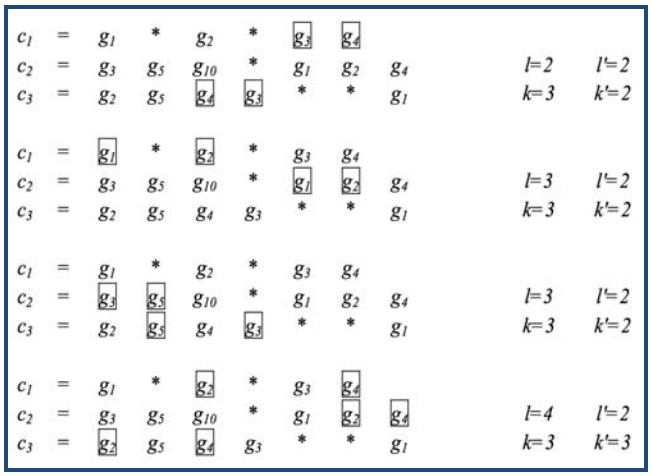

Figure 2: Illustration of sample clusters of size greater than one. l, l', k and k' represent the window size, the number of genes in the window, the total number of chromosomes and the number of chromosomes that appears gene clusters, respectively

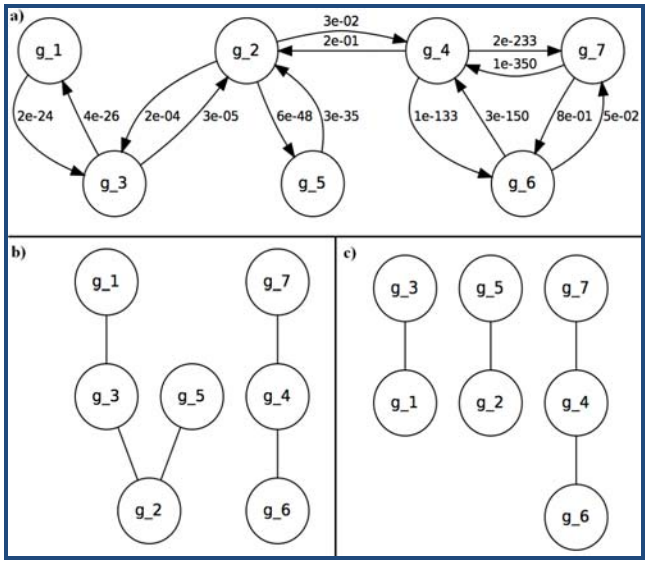

Figure 3: a) The graph by BLAST hits; **b)** *e*-value cut-off: 1e-01; **c)** *e*-value cutoff: 1e-10. Illustration of constructing groups of homologous genes. The first graph is constructed by BLAST *e*-value. Two *e*-value cutoff graphs represent different homologous groups constructed by the different *e*-value cutoff. The stringent *e*-value cutoff makes homologous groups smaller in size, but produces a larger number of homologous groups.

Experimental dataset:

The data set used in the experiment is comprised of 700 bacterial genomes. We got this data set from NCBI (ftp://ftp.ncbi.nlm.nih.gov/genomes/Bacteria). NCBI server supports lists of genes and their sequences in FASTA format. The total number of genes used in 700 bacterial genomes is 2214301 genes, and the average number of genes per a genome s 3163 genes. We apply the algorithm over $1 \leq l \leq 12$ window size on these bacterial genomes, with homolog groups constructed by finding bidirectional best hits using protein-protein BLAST **[14]** with various *e*-value cutoffs, and performing single linkage clustering. The stringent *e*-value cutoff to construct homologous groups generates homologs of the smaller size and the larger number of homologs **(Figure 3).** Since we allow paralogous and orthologous genes, genes that belong to the same homologous group identified by each *e*-value cutoff can appear more than once on the same chromosome and can also appear more than once in the different genomes. We do not assign a gene into new homologous group if only one gene is within the homologous group, since this enables the combination of homolgous group within a window to be reduced. We construct the gene map for each chromosome with the gene location information defined by NCBI.

Results:

To validate our algorithm, we compare the results of gene clusters that include E. coli K12 operons which are experimentally confirmed by the RegulonDB database **[15].** For each predicted gene cluster, we retain gene clusters with an *e*-value below a certain cutoff, while allowing gene clusters that are completely contained within another gene cluster with a better *e*-value so that we do not lose any potential gene clusters that satisfy the definition. We investigate whether these gene clusters correspond mostly to one operon or many operons with different e- value cutoffs, and we partition each gene cluster into maximal subregions so that all genes within each subregion have the same orientation and there are no intervening genes between these genes within the subregion.

For each subregion, we only consider predicted gene clusters that include E. coli. We investigate subregions with different *e*-value cutoffs, and evaluate the agreement between these subregions with experimentally validated E. coli operons from the database. Given a subregion, we compute it with respect to the entire RegulonDB. We choose the *e*-value cutoff 1e-10 which does not drastically effect performance and achieves 96% matches.

We study the distribution of occurrences of operonsin 23 bacterial groups **(Table 1, see supplementary material).** We define the occurrence rate of each bacterial group that appears in gene cluster to be a number of each bacterial group that divides by the total of genomes that appears in gene cluster. We define the overall occurrence rate of a unique bacteria group in all gene clusters to be a number of gene clusters that divides by the total number of gene clusters. Because the algorithm finds

gene clusters by homologs that have a similar gene structure and evolutionary origin to a gene in another species. In gene clusters, we are able to find all 23 bacterial groups.

The gene rearrangement between adjacent genes within bacterial operons is important for function, expression and regulation of these genes **[16, 17].** We study the distribution of gene order with subregions. For a given pair of subregions in a genome g_1 and a set of correspondences with each of them linking a gene in a subregion to a related gene in subregion of another genome g_2, we obtain a subset of one-to-one corresponding pairs of link as follows: if there is more than one link for a gene in subregions$_{s1}$ to another gene in subregions$_{s2}$, we retain the one with the lowest BLAST *e*-value in both s_1 and s_2. In the remaining set of k genes in s_1 and k related genes in s_2, we assign a number from 1 to k to each gene according to the order of genes in subregion, and assign a direction of genes to the number from 1 to k by its gene strand, such as forward and reverse. k genes in subregions$_{s1}$ that correspond to a signed permutation, in which each pair of neighboring genes in s_1 is with number n_1 and n_2, are considered to be a breakpoint if n_1 and n_2 are not consecutive ($|n_1 - n_2| \neq 1$) **[18].** This iterates $|g|(|g|-1)/2$ times, where $|g|$ is the total number of genomes. We define the percentage of conserved neighboring gene pairs to be the total number of neighboring gene pairs that are not breakpoints, which is divided by the total number of neighboring gene pairs by k-1, and use it to evaluate the degree of conservation of gene order. 94.6% had perfectly conserved neighboring gene pairs, which means that gene order within subregions are the same either both the forward and reverse directions. While comparing all subregions in g_1 and g_2, we also evaluate subregion in g_1 that corresponds to one subregion in g_2 with the best percentage of conserved neighboring gene pairs. 96.93% of them are perfectly conserved neighboring gene pairs. We also performed this rearrangement test within gene clusters. 39.74% are perfectly conserved. This is not surprising since it is possible to locate more than one operon within a gene cluster. Although gene order within operons can be unstable [16], our results on gene orientation and gene order indicate that predicted subregions tend to contain only one orientation and the gene order tends to be conserved.

To investigate stronger correlations between the frequency of gene duplication that contains more than two duplicated genes in a subregion and the frequency of gene rearrangement that contains breakpoints between neighboring gene pairs, we compute the Pearson correlation coefficient between them **(Table 2, see supplementary material).** While there were significant positive correlations among subregions, there were no significant correlations within gene clusters with a value of 0.1. Since there can be more than two operons in a gene cluster, the relationship between gene rearrangement and gene duplication is decreased.

Discussion:

In experiments, we validated the results of gene clusters that include Escherichia coli K12 operons, which are experimentally confirmed by RegulonDB. The gene clustering result reveals a significant amount of spatial conservation that is at a higher level than operon and some of these clusters are likely to correspond to uber-operon. The study for the distribution of gene order within a subregion showed the spatial arrangement of genes within bacterial subregion that is important for function, expression and regulation. Although gene order within operons can be unstable **[16],** our results implied that the gene order tends to be conserved and gene orientation appears to face the same direction in the subregion. Because our algorithm does not impose constraints on gene orientations, we can conclude that there is a strong force to preserve gene orientations in bacterial clusters, which is a pre-requisite for functioning as operons. To validate subregions that are predicted from clusters, we investigated a relationship between gene duplication and rearrangement. As a result, a strong positive correlation was predictable between gene duplication and rearrangement considering gene rearrangements include a gene duplication.

Conclusion:

We have developed a gene clustering algorithm that allows the analysis of gene clusters across a large number of genomes and important biological insights to be obtained from this analysis. The proposed algorithm is able to identify gene clusters that are not found by other algorithms, since we developed a different formulation based on window strategy and statistical significance estimates. One of the advantages in the proposed algorithm is that it allows paralogous genes and orthologous genes so that it considers the more biologically accurate model, and the statistical significance estimate while allowing gene cluster that may not appear in every genome that is important for biologists in identifying important gene clusters. We demonstrated proposed algorithms that will be useful for biological insight by analyzing gene clusters across a large number of genomes that can help us understand operon occurrences, gene orientations and distributions of gene rearrangements. We used BLAST and the method in **[19]** to construct homologous gene groups, but it is also possible to use existing database on homology relationships such as COG **[20]** and other ways that establish orthologous and paralogous correspondences **[7].**

References:

[1] Che D *et al. Nucleic Acids Res.* 2006 **34**: 8 [PMID:16682449]

[2] Price M *et al. Nucleic Acids Res.* 2005 **33**: 3 [PMID:15701760]

[3] Salgado H *et al. Proc Natl Acad Sci U S A.* 2000 **97**: 12 [PMID:10823905]

[4] Yang Q & Sze S, *Genome Res.* 2008 **18**: 6 [PMID:18390694]

[5] Bergeron A *et al. Lecture Notes in Computer Science.* 2002 **2452**: 464

[6] He X & Goldwasser MH, *J Comput Biol.* 2005 **12**: 6 [PMID:16108708]

[7] Luc N *et al. Comput Biol Chem.* 2003 **27**: 1 [PMID:12798040]

[8] Parida L, *J Comput Biol.* 2007 **14**: 1 [PMID: 17381345]

[9] Raghupathy N & Durand D, *Mol Biol Evol.* 2009 **26**: 5 [PMID: 19150803]

[10] Hoberman R *et al. Lecture Notes in Computer Science.* 2005 **3388**: 55

[11] Pertea M *et al. Nucleic Acids Res.* 2009 **37:** D479 [PMID: 18948284]

[12] Pasek S *et al. Genome Res.* 2005 **15**: 6 [PMID:15899966]

[13] Kim S *et al. Proc IEEE Comput Syst Bioinform Conf.* 2005 44-55 [PMID:16447961]
[14] Altschul SF *et al. J Mol Biol.* 1990 **215**: 3 [PMID:2231712]
[15] Salgado H *et al. Nucleic Acids Res.* 1999 **27**: 1 [PMID: 9847141]
[16] Itoh T *et al. Mol Biol Evol.* 1999 **16**: 3 [PMID: 10331260]
[17] Tamames, *J Genome Biol.* 2001 **2**: 6 [PMID: 11423009]
[18] Kececioglu J & Sankoff D, *Algorithmica* 1995 **13**: 180
[19] Kellis M *et al. Nature* 2003 **423**: 6937 [PMID: 12748633]
[20] Tatusov R *et al. Science* 1997 **278**: 5338 [PMID: 9381173]

Edited by P Kangueane
Citation: Gangman Yi & Jaehee Jung, Bioinformation 7(5): 251-256 (2011)

BIOINFORMATION

open access

Supplementary material:

Equation 1:

$$p(n,l,l') = \frac{\left((n-l)\left(\frac{l-1}{l'-1}\right) + \binom{l}{l'} \right)}{\binom{n}{l'}}$$

Equation 2:

$$p(k,k',n,l,l') = \sum_{i=k'}^{k} \binom{n}{l'} p(n,l,l')^{i} (1 - p(n,l,l'))^{k-i}$$

Equation 3:

$$e(k,k',n,l,l') = \binom{n}{l'} p(k,k',n,l,l')$$

Table 1: Bacterial group comparison with the original data and gene clusters

Bac Group	%Appear in original data	%Appear in gene cluster
Gammaproteobacteria	23.86	29.15
Firmicutes	20.86	22.73
Alphaproteobacteria	11.86	10.17
Betaproteobacteria	8.29	8.72
Actinobacteria	7.43	6.71
Euryarchaeota	4.71	3.40
Cyanobacteria	4.57	5.38
Epsilonproteobacteria	2.86	2.86
Deltaproteobacteria	2.57	1.32
Bacteroidetes/Chlorobi	2.57	1.50
Crenarchaeota	2.29	1.81
Chlamydiae/Verrucomicrobia	1.86	3.63
Spirochaetes	1.57	2.78
Chloroflexi	1.00	0.99
Other Bacteria	1.00	0.35
Thermotogae	1.00	1.04
Deinococcus-Thermus	0.57	0.47
Aquificae	0.29	0.06
Acidobacteria	0.29	0.09
Other Archaea	0.14	0.01
Planctomycetes	0.14	0.03
Fusobacteria	0.14	0.01
Nanoarchaeota	0.14	0.0

Table 2: Pearson correlation coefficient of subregions and gene clusters.

	subregion	within cluster
Correlation coefficient	0.63	0.10
p-value	1.1e-16	5.6e-16