Jahn *et al. BMC Bioinformatics* 2013, **14**(Suppl 15):S14
http://www.biomedcentral.com/1471-2105/14/S15/S14

BMC Bioinformatics

PROCEEDINGS Open Access

Statistics for approximate gene clusters

Katharina Jahn[1,2*], Sascha Winter[3], Jens Stoye[1,2], Sebastian Böcker[3]

From Eleventh Annual Research in Computational Molecular Biology (RECOMB) Satellite Workshop on Comparative Genomics
Lyon, France. 17-19 October 2013

Abstract

Background: Genes occurring co-localized in multiple genomes can be strong indicators for either functional constraints on the genome organization or remnant ancestral gene order. The computational detection of these patterns, which are usually referred to as gene clusters, has become increasingly sensitive over the past decade. The most powerful approaches allow for various types of imperfect cluster conservation: Cluster locations may be internally rearranged. The individual cluster locations may contain only a subset of the cluster genes and may be disrupted by uninvolved genes. Moreover cluster locations may not at all occur in some or even most of the studied genomes. The detection of such low quality clusters increases the risk of mistaking faint patterns that occur merely by chance for genuine findings. Therefore, it is crucial to estimate the significance of computational gene cluster predictions and discriminate between true conservation and coincidental clustering.

Results: In this paper, we present an efficient and accurate approach to estimate the significance of gene cluster predictions under the approximate common intervals model. Given a single gene cluster prediction, we calculate the probability to observe it with the same or a higher degree of conservation under the null hypothesis of random gene order, and add a correction factor to account for multiple testing. Our approach considers all parameters that define the quality of gene cluster conservation: the number of genomes in which the cluster occurs, the number of involved genes, the degree of conservation in the different genomes, as well as the frequency of the clustered genes within each genome. We apply our approach to evaluate gene cluster predictions in a large set of well annotated genomes.

Background

Gene order-based analysis of whole genomes has become an important field in comparative genomics. It is well known that genomes evolve, not only at the level of nucleotide sequence, but also by means of large-scale rearrangements, such as inversions and transpositions, as well as changes of the gene content. Focusing on this higher-level structure, genomes are usually modeled as sequences of integers with genes belonging to the same gene family represented by the same integer. If no selective pressure was acting on whole genome evolution, gene order and gene content would randomize over time. In reality, particularly in bacterial genomes, we observe low overall conservation of gene order between species, contrasted by a small number of well-conserved segments. These are often referred to as *gene clusters*. Such local aberrations from genome randomization provide evidence for various biological phenomena and are of high interest in functional and evolutionary genomics [1-9].

When comparing a large number of genomes, the identification of these structures can be a challenging task since conservation patterns may be highly variable across species. Due to micro-rearrangements, gene order can also vary between cluster occurrences. Gene insertions and losses, or mis-annotations may lead to cluster occurrences interrupted by genes that do not belong to the cluster, or containing only a subset of the clustered genes. Moreover, a gene cluster may be present only in a subset of the genomes under study. (A gene cluster displaying all these features, except for mis-annotations, is shown in Additional File 1) Taking these effects into account in different ways, several models of gene clusters

* Correspondence: kjahn@cebitec.uni-bielefeld.de
[1]Genome Informatics, Faculty of Technology, Bielefeld University, 33615 Bielefeld, Germany
Full list of author information is available at the end of the article

and algorithms for their detection have been suggested [10-17].

However, it may as well be the case that seemingly conserved patterns occur merely by chance. To estimate the likeliness of such events, appropriate statistics are needed to quantify the probability of finding gene clusters by chance. Such statistics have been developed for some gene cluster models, in particular *r-windows* [18-20], *segmental homologies* (k-clumps) [21], and max-gap clusters [22-24]. Other methods solve the problem of assigning significance to predicted gene clusters in an ad-hoc manner, including C-Hunter [25], OrthoCluster [26], MCMuSeC [16], CYNTENATOR [27], and i-ADHoRe [28]. In this paper we consider the gene cluster model of approximate common intervals [15]. This can be easily applied to a variety of use cases, as it offers a combination of parameter flexibility and efficiency of computation, even for very large data sets. The variant *reference gene clusters* [17] proves especially useful. We provide a statistical test for evaluating gene cluster predictions against the null hypothesis of random gene order. For our background model, we consider, for each genome G, a random string S of the same length where each character (gene) has a probability proportional to its frequency in G. For multi-chromosomal genomes, or in cases where the (unfinished) genome sequence consists of multiple contigs, we do the same for each chromosome/contig individually. We then estimate p-values, that is, the probabilities of gene clusters of the observed quality or higher being found in the random genomes by chance.

Since the random genomes are drawn independently, we can proceed as follows: For each genome, we compute the likelihood of a gene cluster occurring in the corresponding random genome. These are the individual p-values for each genome. Next, we demonstrate how to combine p-values from individual genomes into one p-value for the gene cluster. The problem of multiple testing is then considered by applying a false discovery rate control to minimize this effect. Finally, we demonstrate that there is excellent concurrence between our calculated p-values and empirically determined p-values and that the method is able to recognize known gene clusters from large genomic data sets.

Methods

Preliminaries

We model a genome as a *string* over a finite alphabet $\Sigma = \{1, \ldots, \sigma\}$ of gene family ids, such that genes belonging to the same homology family are represented by the same integer. In the following, we use the terms "genome" and "string" and, also, "gene" and "character" interchangeably. Given a string S, we denote its *length* by $|S|$ and refer to its ith character as $S[i]$, $1 \leq i \leq |S|$. A character $c \in \Sigma$ is said to *occur* at position i in S if $S[i] = c$. A substring of S from position a to b is denoted $S[a, b]$ for $1 \leq a \leq b \leq |S|$. To capture the character content of $S[a, b]$ regardless of sequential arrangement and multiple character occurrences, we define the *character set* of $S[a, b]$ as

$$\mathcal{CS}\left(S\left[a,\ b\right]\right) := \{S\left[i\right] : a \leq i \leq b\} \subseteq \sum.$$

The corresponding index interval $[a, b]$ is termed a *location* of $C \subseteq \Sigma$ if and only if $C = \mathcal{CS}(S[a, b])$. An interval $[a, b]$ is *left-maximal* if $a = 1$ or $S[a-1] \notin \mathcal{CS}(S[a, b])$; it is *right-maximal* if $b = |S|$ or $S\left[b+1\right] \notin \mathcal{CS}(S[a, b])$; and it is *maximal* if it is both left- and right-maximal.

Given $k \geq 2$ strings $S_1, \ldots S_k$, k maximal intervals written as k-tuple $([a_1, b_1], \ldots, [a_k, b_k])$ are called *common intervals* in $S_1, \ldots S_k$ if and only if

$$\mathcal{CS}(S_1[a_1, b_1]) = \ldots = \mathcal{CS}(S_k[a_k, b_k]) =: C.$$

Given that $S_1, \ldots, S_k$ are genomes, the above character set C corresponds to a gene cluster with perfectly conserved gene content.

In order to model gene clusters with incomplete conservation patterns, we quantify the differences in the gene content of their approximate gene cluster occurrences via their *symmetric set distance*. This measure defines the distance between two finite sets C and C' as the cardinality of their *symmetric difference*:

$$D(C, C') := |C \backslash C'| + |C' \backslash C| = |C \cup C'| - |C \cap C'|.$$

This constitutes a metric and therefore meets all intuitive notions of a distance measure, such as validity of the triangle inequality. In the context of gene clusters, it corresponds to a simple summation over the genes deleted and the genes inserted into a cluster occurrence. Like earlier character set based gene cluster models [29], it disregards recurrences of genes within the same cluster occurrence.

Based on this distance notion, we extend the concept of character set locations towards approximate conservation: Given an integer $\delta \geq 0$, we define an interval $[a, b]$ in a string S as a *δ-location* of character set C, if and only if, $D(C, \mathcal{CS}(S[a, b])) \leq \delta$, and $\mathcal{C} \cap \mathcal{CS}(S[a, b]) \neq \emptyset$.

Let $S_1, \ldots, S_k$ be genomes over a gene alphabet Σ. Let $s \geq 2$ be the minimum cluster size, $\delta \geq 0$ a distance threshold and k' a *quorum* parameter with $2 \leq k' \leq k$. A *reference gene cluster* for parameters s, δ and k' is a set of genes $C \subseteq \Sigma$ with $|C| \geq s$ such that C has an exact occurrence in one of the genomes and δ-locations in at least $k' - 1$ other genomes. In other words, there exist i, a, b such that $C = CS(S_i[a, b])$, and $J \subseteq \{1, \ldots, k\} - \{i\}$ with $|J| \geq k' - 1$ such that each S_j has a δ-location of C for all $j \in J$.

In [17] we studied the following problem: Given genomes $S_1, \ldots, S_k$ and parameters s, δ, k', find all reference gene clusters $C \subseteq \Sigma$ in $S_1, \ldots, S_k$, and for each reference gene cluster, output all its optimal δ-locations. (A δ-location is not optimal if it is a subinterval of a δ-location that has a smaller distance to C.) We introduced an efficient algorithm that runs in $O(k^2n^2\delta^2 + k^2n^2)$ time using $O(kn^2)$ space, where n is the length of the largest genome [17]. The algorithm is exact, meaning that it is guaranteed to find all reference gene clusters and their optimal occurrences as specified by the search parameters.

The above definitions do not take into account multi-chromosomal genomes, or genomes that were not completely assembled and still consist of several contigs. However, it is simple to extend these definitions, as well as the remainder of this paper, to the multi-chromosomal or multi-contig case. For example, we may assume that the different chromosomes/contigs of one genome are concatenated in a single string, separated by symbols $\$ \notin \Sigma$. We can then assume that neither a gene cluster nor an interval is allowed to contain the character \$. Further details will be omitted, aside from saying that some of the complexity bounds mentioned below actually improve for multi-chromosomal and multi-contig genomes.

Significance of a gene cluster for one genome

In this section we estimate the probability of a fixed gene cluster $C \subseteq \Sigma$ having a δ-location in a random genome S of length n, i. e. the p-value of finding an occurrence of C in genome S :

$$\text{p-value} = \mathbb{P}(S\,\text{has a}\ \delta\text{-location of}\ C) \tag{1}$$

In the following we assume that $\delta \leq |C| - 2$ holds. Otherwise the p-value is equal to one whenever $C \cap \mathcal{CS}(S) \neq \emptyset$, and zero otherwise. Let $p(L, d) = p(L, d, C)$ be the probability that a random occurrence of length L has a symmetric set distance *exactly* d to C. Let $q(L, \delta) = q(L, \delta, C)$ be the probability that it has a symmetric set distance of *at most* δ. Note that $p(L, d)$ and $q(L, \delta)$ depend on the cluster C; in the following, our notations omit this dependency for the sake of readability. Then, $q(L, \delta) = \sum_{d=0}^{\delta} p(L, d)$. To simplify our computation, we assume that occurrence probabilities are independent for all intervals $[a, b]$ where $1 \leq a \leq b \leq n$. Clearly, this assumption is not correct: Let A be the event that interval $[i, j]$ forms a δ-location of C, and let B be the event that interval $\left[i, j + 1\right]$ forms a δ-location of C. The fact that the intervals share all positions but one creates a number of non-trivial dependencies. In case $S[j + 1] \in \mathcal{CS}(S[i,\ j])$, the two events are either both true or both false. When the distance between $\mathcal{CS}(S[i,\ j])$ and C is smaller than δ, $p(B|A) = 1$, and vice versa. Such dependencies apply not only to $[i, j]$ and $[i, j + 1]$, but to all intersecting interval pairs. However, we will show later on that the p-values computed under this assumption are very close to the true p-values, for any realistic choices of parameters. We estimate the p-value for a single genome as

$$\begin{aligned}\mathbb{P}(S\,\text{has a}\ \delta\text{-locations of}\ C) &\approx 1 - \prod_{a \leq b} (1 - q(b - a + 1, \delta)) \\ &= 1 - \prod_{L=1,\ldots,n} (1 - q(L, \delta))^{n-L+1}.\end{aligned} \tag{2}$$

To minimize the effects of rounding error accumulation, we instead compute

$$\mathbb{P}(S\,\text{has a}\ \delta\text{-location of}\ C) \approx 1 - \exp\left(\sum_{L=1,\ldots,n} (n - L + 1) \cdot \log\left(1 - q(L,\ \delta)\right)\right) \tag{3}$$

which can be calculated with high accuracy, using mathematical library functions for $f(x) := \exp(x) - 1$ and $g(x) := \log(x + 1)$.

Exact computation using dynamic programming

We need to compute $p(L,\ d)$, the probability that the character set of a random interval of length L has a symmetric set distance d to a given (fixed) gene cluster C, for all L and $d \leq \delta$. Let S_L be a random string of length L, and let $\mathbb{P}(c)$ denote the probability of character $c \in \Sigma$ for any position of the random string. For a sub-alphabet $\Sigma' \subseteq \Sigma$, set $\mathbb{P}(\Sigma') := \Sigma_{c \in \Sigma'} \mathbb{P}(c)$ The distance d between C and $\mathcal{CS}\ (S_L)$ can be partitioned as $d = d_- + d_+$: Here, d_- is the number of characters from C that are *missing* in S_L, and d_+ is the number of *additional* characters in $\mathcal{CS}(S_L)$. Consequently, we can partition the positions in S_L into two types: those positions containing characters from C, and those positions containing characters from $\bar{C} := \sum - C$. Assume that l positions of S_L are occupied by characters from C, $0 \leq l \leq L$, and that $L - \ l$ positions are occupied by characters from $\bar{C}$. We calculate

$$\begin{aligned}p(L,\ d) &= \sum_{d_-=0}^{d} \mathbb{P}(\ |C \backslash \mathcal{CS}(S_L)|\ = d_- \wedge |\mathcal{CS}(S_L) \backslash C|\ = d - d_-) \\ &= \sum_{d_-=0}^{d} \sum_{l=0}^{L} \mathbb{P}(|\ \{i | S_L[i] \in C, 1 \leq i \leq L\}|\ = l \wedge |C \backslash\ \mathcal{CS}(S_L)|\ = d_- \wedge |\mathcal{CS}(S_L) \backslash C| = d - d_-) \\ &= \sum_{l=0}^{L} \sum_{d_-=0}^{d} p_0(l, L).\ p^-(l, d_-).\ p^+(L - l,\ d - d_-)\end{aligned} \tag{4}$$

where $p_0(l, L)$ is the probability of drawing a random string of length L with l characters from C and $L - l$ characters from $\bar{C}$; $p^-(l, d_-)$ is the probability that, for a random string S_l of length l over the alphabet C, $\mathcal{CS}(S_l)$ is missing d_- characters from C; and $p^+(L - l, d - d_-)$ is the probability that, for a random string $S_{l'}$ of length $l' = L - l$ over the alphabet $\bar{C}$, $\mathcal{CS}(S_{l'})$ contains $d_+ = d - d_-$ different characters. If all these values are known, we can compute the desired probability using (4) in time $O(dL)$. In practice, we found that $p^-(l, d_-)$ in (4) decreases rapidly with increasing l. To this end, we can stop the summation, as well as the computation of $p^-(l, d_-)$ and $p^+(L - l,\ d_+)$, as soon as $\sum_{d_-=0}^{d} p_0(l, L).\ p^-(l, d_-).\ p^+(L-l,\ d_+)$ no longer contributes to the sum.

Computing $p_0(l, L)$ is straightforward, using the binomial distribution: One can see that

$$p_0(l, L) = \binom{L}{l} \cdot \mathbb{P}\ (C)^l \cdot \mathbb{P}\ (\bar{C})^{L-l}$$

holds. It remains to be shown how to compute $p^-(l, d_-)$ for missing characters and $p^+(L-l, d_+)$ for additional characters.

Missing characters

We first demonstrate how to compute $p^-(l, d_-)$, the probability that a random string S_l of length l over the alphabet C is missing d_- characters from C. Let $h := |C| - d_-$ be the number of *hits* from $\mathcal{CS}(S_l)$ in C. For readability, we base our computation on the number of hits h. The order of the characters in the random string is not relevant, so we simply check whether a certain character from C has been generated. The statistical equivalent is rolling dice. We assume, for simplicity, that $C := \{1, \ldots, Z\}$. Probabilities of the characters in C are conditional probabilities of the same characters in Σ. We define

$$p[z, l, h] := \mathbb{P}(h\,\text{different outcomes for}\,l\,\text{rolls for}\,\#'s\ 1, ..., z).$$

So, $p[z, l, h]$ is the probability that, by throwing l dice with numbers $1, \ldots, z$, exactly h different numbers have been rolled. In the following, we not only iterate over the number of rolled dice and different outcomes, but also over the numbers that can be rolled. In cases where only numbers $1, \ldots, z$ out of $1, \ldots, Z$ can be rolled, we can calculate the conditional probability of each outcome with the recurrence:

$$\begin{aligned} p[z, l, h] &= \mathbb{P}(\text{no } z \text{ rolled with } l \text{ dice}) \cdot p[z-1, l, h] \\ &+ \sum_{\ell=1}^{l} \mathbb{P}(\ell \text{ times } z \text{ rolled with } l \text{ dice}) \cdot p[z-1, l-\ell, h-1]) \end{aligned} \quad (5)$$

We initialize $p[z, 0, 0] = 1$, $p[z, 0, h] = 0$, $p[z, l, 0] = 0$, and $p[0, l, h] = 0$ for all z and all $l, h > 0$. The two missing probabilities are computed using a binomial distribution. In the end, $p^-(l, d_-) = p[Z, l, h]$ is the probability that in a random string of length l over alphabet C, exactly h different characters from C have been generated. Computation takes $O(|C|hl^2)$ time and $O(hl)$ space, as only the values for $z = Z$ need to be stored.

Additional characters

The recurrence introduced in equation (5), can also be used to compute $p^+(l, d_+)$. We need to set $h := d_+$, and exchange the roles of C and $\bar{C}$. Unfortunately the latter modification has a strong impact on the practical runtime, due to the linear dependence now being on the much larger $|\bar{C}|$ (compared to the rather small $|C|$ for $p^-(l, d_-)$). However, we can mitigate this by pooling genes based on the frequency of their occurrences in the original genome. The genes within each pool have the same occurrence probability in the random genome and need not be distinguished in our calculations. It is sufficient to know for each pool how many of its genes originate from C, and $\bar{C}$ respectively.

Let f be the number of different occurrence frequencies observed in the original genome, and let $F_1, \ldots, F_f$ denote the corresponding gene pools. Given a fixed gene cluster C, we denote by $F_z^{\bar{C}}$ the subset of pool F_z, $1 \le z \le f$, that consists only of genes from $\bar{C}$. We then modify recurrence (5) as follows:

$$\begin{aligned} p[z, l, h] &= \mathbb{P}(\textit{no gene of } F_z^{\bar{C}} \textit{ rolled with } l \textit{ dice})^* p[z-1, l, h] \\ &+ \sum_{\ell=1}^{l} \sum_{h'=1}^{\min(h, |F_z^{\bar{C}}|)} (\mathbb{P}(\ell \textit{ genes of } F_z^{\bar{C}}, h' \textit{differnet ones, rolled with } l \textit{ dice})^* p\,[z-1, l-\ell, h-h']) \end{aligned}$$

The initializations are the same as for the previous recurrence. We can use the binomial distribution to compute the value of $\mathbb{P}$ (no gene of $F_z^{\bar{C}}$ rolled with l dice) and the same type of recurrence as in equation (5) to compute the second probability, $\mathbb{P}(\ell$ genes of $F_z^{\bar{C}}$, h' different ones, rolled with l dice). In the end, $p^+(l, d_+) = p[f, l, h]$ is the probability that in a random string of length l over alphabet $\bar{C}$, exactly h different characters from $\bar{C}$ have been generated.

Due to the second summation, the asymptotic time complexity of the recurrence becomes $O(fh^2l^2)$. We observe that, in practice f, the number of different gene occurrence frequencies is very small compared to $\bar{C}$ and is typically in the size range of large gene clusters. Also, the vast majority of genes occur only once in a genome, with pool sizes dropping quickly for larger occurrence frequencies. Since h' is bounded, not only by h but also by $|F_f^{\bar{C}}|$, the quadratic dependence on h is unlikely to be relevant in practice.

Next, we show how the values of $\mathbb{P}(\ell$ genes of $F_z^{\bar{C}}$, h' different ones, rolled with l dice) can efficiently be computed. Ideally, we would like to precompute these values once for every genome, providing constant-time lookup during computation of p-values for the different gene clusters. At first sight, these probabilities seem to be specific for each cluster, as the gene pools need to be restricted to their complement $\bar{C}$. However, we know that all genes in a pool have the same occurrence probability, therefore it is sufficient to compute the above values for all residual pools, after removing a certain number - not a certain set - of genes. For small pools, which are in the majority, not much extra work is required to do this for all possible residual sizes. For large pools exact computations may, in practice, be too costly. In this case, we suggest working with pools based on the complete alphabet Σ. Due to their size, the removal of a few elements has little influence on the conditional probabilities of the remaining elements.

In practice, we use a faster, but less exact approach: We replace the more accurate estimation of $p^+(l, d_+)$ with a much faster preprocessing that leads to almost identical results. To this end, we compute a global $P^+(l, d_+)$ for the complete alphabet, $\bar{C} := \Sigma$, during preprocessing. This is achieved using recurrence (5). We then assume that, for any cluster C with additional character probabilities $p^+(l, d_+)$, we have $p^+(l, d_+) \approx P^+(l, d_+)$. In doing so, we ignore the fact that the gene cluster C removes some of the genes from the pool of potential additional genes. Clearly, this computation can be carried out very quickly, as we have to compute $P^+(l, d_+)$ only once for each genome. Depending on the distribution of occurrence probabilities, the above approximation can distort the results significantly. We account for this by setting $p^+(l, d_+) \approx (1 - \mathbb{P}(C))^{d^+} \cdot P^+(l, d_+)$, thereby taking into account the occurrence probabilities of the genes in C.

Significance of a gene cluster in multiple genomes

Thus far, we have concentrated on the probability of observing gene cluster occurrences in a single genome. To estimate the significance of observing a gene cluster in multiple genomes, we need to combine the individual probabilities into a single p-value. This gives the probability of observing a gene cluster C, at least as well conserved in the randomized genomes as in the original genome set.

We begin by formalizing the notion of "at least as well conserved". Consider the case where a δ-location of C is detected in all genomes $S_1, \ldots, S_k$. To simplify notation, we assume that $S_1, \ldots, S_k$ are the remaining genomes after removing the reference genome, the one from which we took the interval to generate C. Clearly no p-value estimation is necessary for this genome, and it can be omitted from the following calculations. Let $\mathbf{d} = (d_1, \ldots, d_k)$ be a distance vector, such that d_i is the distance between C and its best approximate occurrence in genome S_i, $1 \leq i \leq k$. We denote by $\mathbf{d}_{\text{obs}}$ the distance vector observed for C and the original genomes. To make different distance configurations comparable, we need to define a linear order of all possible distance vectors. We chose an ordering based on the total distance, $\sum_{i=1}^{k} d_i$. We denote by d_{obs} the sum distance of $\mathbf{d}_{\text{obs}}$. To exclude configurations with δ-locations in fewer genomes than observed in the original data, we further require that each individual distance in the vector is at most δ. To calculate the probability of observing any distance vector that satisfies the above constraints, we define the following recurrence:

$$M[i, d] = \sum_{d'=0}^{\min(d,\delta)} (\mathbb{P}(\text{best } \delta\text{-location in } S_i \text{ has distance } d' \textit{to}\, C) \cdot M[i-1, d-d']). \qquad (7)$$

The base cases are $M[0, 0] = 1$, and $M[0, d] = 0$ for $d > 0$. $\mathbb{P}$(best δ-locations in S_i has distance d' to C) equals $\mathbb{P}$(C has a d'-location in S_i) − $\mathbb{P}$(C has a $(d' - 1)$-location in S_i). These probabilities can be computed with equation (2). Summing over all $M[k, d]$, $0 \leq d \leq d_{\text{obs}}$, gives the desired p-value. i. e. the probability that C is at least as well conserved in the randomized genomes as in the original dataset. This computation takes time $O(k^2\delta^2)$, as i and d are bounded by k and d_{obs}, respectively, and we have $d_{\text{obs}} \leq k \cdot \delta$.

When a gene cluster is observed only in a subset of the studied genomes, it becomes tricky to define the meaning of "at least as well conserved": Is a gene cluster better conserved if it occurs in many genomes at a low quality, or in fewer genomes but at higher quality? We suggest that the latter should be given preference. Otherwise, there is the risk of systematically preferring gene clusters that occur in a large number of genomes, yet only incompletely in the form of one or two genes, over clusters that are very well conserved but only in a small number of genomes. We believe that the latter are the more interesting cases. Therefore we say a gene cluster C with δ-locations in k' out of k genomes and sum distance d_{obs} is conserved at the same or better quality in the randomized genomes, if it has δ-locations in at least k' of them, and the best k' δ-locations (from k different genomes) have a sum distance of at most d_{obs} to C. Unfortunately the recurrence used in equation (7) cannot be extended to compute the corresponding probabilities. We need to track the sum of the k' smallest distances below δ, after processing the first $i \leq k$ genomes. This value cannot be computed in a simple recursive manner, as there is no optimal substructure underlying the problem: The sum, after processing i genomes, depends not only on the sum before the genome was added and the distance for the new genome, but also on the previous number of distances below δ and the values of these distances. Only with all of this information is it possible to decide whether or not the distance encountered for the new genome needs to be added to the sum. In the absence of an efficient dynamic programming approach, we need to sum probabilities over all $k^{\delta+1}$ distance vectors. This becomes infeasible for larger δ.

In order to avoid exponential running time, we studied a simpler approach where we use a fixed distance threshold δ' for all genomes. This can be either the original threshold δ, or the largest entry in $\mathbf{d}_{\text{obs}}$, i. e. the largest of the distances between C and its best δ-location in each genome.

As a consequence, we do not need to sum over pairwise distances in the recurrence, but over the number of genomes that contain a δ'-location. Let p_j be the likelihood of a δ'-location in genome S_j, computed using equation (2). The likelihood of having δ'-locations in exactly i out of j genomes $S_1, \ldots, S_j$, $1 \leq i \leq j \leq k$, can be computed using the recurrence

$$Q[i,j] = (1-p_j)\cdot Q[i,j-1] + p_j \cdot Q[i-1,j-1]. \quad (8)$$

The base cases are $Q[0, 1] = 1 - p_1$, $Q[1, 1] = p_1$, and $Q[i,j] = 0$ for all $i > j$. The likelihood of finding at least k' δ'-locations in k genomes is then the sum over all $Q[i, k]$ with $k' \leq i \leq k$. Computation requires $O(k^2)$ time. This method will be referred to as "global distance bound".

Unfortunately, the above approach has the following problem: Consider two (otherwise identical) gene clusters, both found in three genomes. The first cluster is found with distances 0, 1 and 4 in the three genomes; the second cluster with distances 3, 4 and 4. Common sense tells us that the first gene cluster is more significant, because it is less likely to occur by chance. However, the approach described above will come up with identical likelihoods, as, in both cases we have $\delta' = 4$. In fact, for the first cluster it may be beneficial to exclude the last occurrence, as this may lead to a smaller p-value; we omit the details.

To ensure that gene clusters of this type are evaluated differently while keeping the computational complexity reasonable, we resort to the following simplification: Recall that $\delta' \leq \delta$ is the maximum distance of the cluster to any occurrence. For genomes where we do not detect a δ-location, we use the single-genome p-value with distance threshold δ'; for genomes other than the reference genome that contain a δ-location (and, hence, a δ'-location), we use the single-genome p-value with distance threshold given by the distance of the detected interval in this genome. In the above example, for the first cluster, we use distance threshold 0 for the first genome, 1 for the second genome, and 4 for the third genome; for the second cluster we use distance threshold 3 for the first genome and 4 for the remaining two. This method will be referred to as "individual distance bounds".

False discovery rate control

Since we are testing significance not only for a single gene cluster, but for the complete set of gene cluster predictions reported by a search extending over all possible reference intervals, we need to adjust our p-values accordingly. We use false discovery rate (FDR) control [30] to counteract the problem of multiple testing. In detail, we sort all the clusters by their p-value and multiply the p-value p of any gene cluster with the number of possible reference intervals in all k genomes divided by the index i in the sorted cluster list:

$$p_i^{FDR} = p_i . \frac{\sum_{j=1}^{k} \frac{n_j . (n_j + 1)}{2}}{i}, \quad (9)$$

where $n_1, \ldots, n_k$ are the genome lengths. This is a conservative estimation and comes at the cost of increasing the probability of producing false negatives. That is, gene clusters that should be regarded as significant may be declared non-significant after the FDR correction. In addition, equation (9) actually overestimates the number of gene clusters tested as we do not take into consideration certain gene clusters that appear multiple times in a genome (and should be tested only once). We do not use the more powerful Šidák correction [31], as independence of the different tests cannot be guaranteed. FDR-corrected p-values can be larger than one, which solely indicates that this correction is conservative.

Evaluation of p-value accuracy for a single genome

We now evaluate the accuracy of the p-value estimation we introduced above for a single genome. Recall that we use two simplifications to keep this task computationally feasible. First, we assume that the intervals within a genome do not overlap, in order to gain statistical independence of the probabilities $q(L, \delta)$ in equation (2). Second, we employ a heuristic approach to deal with additional characters in cluster occurrences. To show that our calculations are still sufficiently accurate, we compare our estimated p-values with p-values derived empirically from large-scale simulations of random genomes.

Two simulation studies were performed, one with biological data and one with simulated data. To reduce simulation time, we performed the first study on four, somewhat small, bacterial genomes with just 600 to 850 genes each, namely *Buchnera aphidicola APS*, *Ureaplasma urealyticum*, *Mycoplasma pneumoniae*, and *Borrelia burgdorferi*. We downloaded these genomes from the NCBI database [32] and assigned homology families, using GHOSTFAM [33] with standard parameters. Ten billion random sequences were then generated, with the same length and character frequencies as the original genomes.

Our reference gene cluster detection algorithm [17] was applied to the four original genomes, with three different parameter settings $(\delta = 1, s = 4)$, $(\delta = 3, s = 6)$, and $(\delta = 5, s = 9)$. The quorum parameter was set to $k' = 2$ in each case. This search returned 459 gene clusters. We computed for each gene cluster C, the p-value of each occurrence, excluding the reference interval. This was done with equation (2), by setting the threshold to δ', the symmetric set distance between C and the character set of the occurrence. Next, the p-value for a genome with randomized gene order containing a δ'-location of C was empirically determined. To this end, all random genomes corresponding to the genome where the occurrence was observed were searched for intervals with a symmetric set distance of, at most, δ' to C. The number of genomes with at least one such occurrence was divided by the total number of tested genomes. This gives an empirical estimate of the true p-value. When no occurrences were found, we omitted the pair from further analysis, as this shows the empirical p-value to be out of the scope of the current evaluation. This resulted in the omission of 277 gene clusters. The maximum p-value calculated for any such omitted pair was $9.23 \cdot 10^{-11}$. Therefore all of the omitted values fall into the 95% Agresti-Coull confidence interval [34] of finding no occurrences, that has an upper bound of $4.64 \cdot 10^{-10}$. (Note that when searching for gene clusters, what matters in p-value estimation is the *order of magnitude*. For example, p-values $2.7 \cdot 10^{-380}$ and $5.4 \cdot 10^{-380}$ differ by a factor of two; still, we would regard a gene cluster with either of these p-values as *highly* significant).

For the remaining 182 gene clusters, we have plotted the calculated p-values against the empirical p-values in Figure 1. Unless our independence assumption is severely violated, this should result in points close to the main diagonal. Note that both the calculated p-values and the empirical p-values can deviate from the true p-value; this estimate becomes highly inaccurate, particularly for very small empirical p-values, as only few of the billions of genomes contain an occurrence of the gene cluster.

Nevertheless, we find an excellent agreement between the calculated p-values and the empirical p-values. For numerical comparison of these values, we transform both into the log scale: Otherwise, almost all p-values are far too small to contribute to Pearson correlation or coefficient of determination. The Pearson correlation of the log-log pairs is $r = +0.9976775$ ($r^2 = 0.9953604$). As we want to use the calculated p-values as a predictor of the empirical p-values, we also computed the coefficient of determination

$$R^2 = 1 - \frac{\sum_i (y_i - x_i)^2}{\sum_i (y_i - \bar{y})^2}$$

where the y_i are the observations (log empirical p-values), the x_i are the predictions (log calculated p-values), and $\bar{y}$ is the mean of the observations. For the bacterial genomes, we achieve $R^2 = 0.9951661$. The observed deviations for small probabilities must be attributed to the fact that, for these reference gene clusters, ten billion random genomes are not enough to give a good estimate of the true value. It appears very likely that the empirical p-value is inaccurate, rather than the calculated p-value.

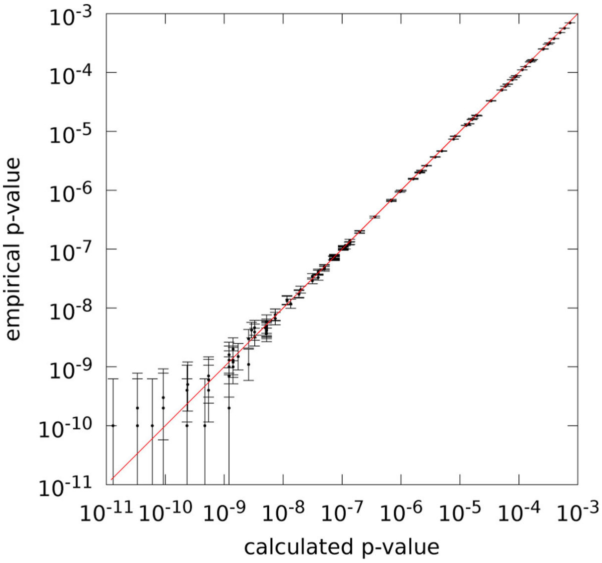

Figure 1 Comparison of empirical and calculated p-values for bacterial genomes. Comparison of empirical and calculated p-values for gene clusters from four bacterial genomes (log-log plot, $N = 182$). Pearson correlation of log-log-pairs is $r = +0.9976775$ ($r^2 = 0.9953604$), coefficient of determination of log-log-pairs is $R^2 = 0.9951661$ (identity in red). Error bars visualize the 95% Agresti-Coull confidence interval.

We argue that these results strongly indicate that our calculated p-values are very close to the true values; hence, although equation (2) does not take into account statistical dependencies, our calculations are still highly accurate.

The number of p-value pairs in the above study is relatively small and not sufficient to firmly conclude that our calculations are accurate. To obtain a greater degree of certainty, we performed a second study using random genomes. Here, random genomes of different sizes and with different character distributions were generated. In order to create random genomes with similar characteristics to true biological data, we studied the gene family size distribution in real genomes. (A complete list of the genomes can be found in Additional file 1). Gene family size appears to roughly follow a heavy-tailed distribution (Additional file 1). The Pareto distribution was therefore selected for simulating genomes. The probability density function is:

$$f(x) = \frac{\alpha x_m^{\alpha}}{x^{\alpha+1}}, \quad \alpha, k > 0, \quad x \geq x_m. \tag{10}$$

In the following calculations, we use $x_m = 1$, so that each gene appears at least once. The bacterial genomes we use later on for our evaluation approximately follow a Pareto distribution with $\alpha = 2.8$ (Additional file 1).

For each random genome we uniformly draw its length $n \in [1250,\ 1750]$. To select the character content of the genome, we repeat the following: We choose the next character and draw the number of occurrences of this character in the random genome using the above Pareto distribution. We repeat this until all n positions of the random genome are filled, discarding surplus copies of the last added character. In this manner, we generated ten genome contents.

To generate a random gene order, we could concatenate the genes and shuffle the resulting string. To speed up computations, we proceeded in a slightly different way: Instead of generating a random genome and then searching for reference gene clusters, we simply assume a gene cluster C to be present. To obtain useful p-values we combined different strategies to construct C: (1) nine clusters are chosen with two to ten genes using the most commonly occurring genes; (2) nine clusters are chosen with two to ten genes using the rarest genes, usually occurring only once; (3) 82 clusters are chosen with two to ten genes, randomly selected. For each of these 100 gene clusters we randomly choose a maximum allowed distance $\delta \in [0, |C| - 2]$. As mentioned earlier, for higher values of δ exact p-values can be easily computed, simply by testing whether a single gene from C is present in the genome.

We then proceeded as described above, comparing the calculated p-value to an empirical p-value obtained from one billion random draws. From the $10 \cdot 100 = 1000$ gene clusters tested, we omitted 249 p-value pairs where no occurrences were empirically observed. The maximum calculated p-value for any such omitted pair is $2.45 \cdot 10^{-9}$, while the upper bound of the 95% Agresti-Coull confidence interval for observing no occurrence is $4.64 \cdot 10^{-9}$. To further increase the accuracy of the empirical estimation of the 49 clusters that were found between one and ten times only, we did an additional nine billion random draws. For five clusters, we still observed less than ten occurrences; the largest calculated p-value among these clusters is $5.40 \cdot 10^{-10}$, the upper bound of the 95% Agresti-Coull interval for observing nine occurrences is $1.74 \cdot 10^{-9}$. As for these cases the empirically determined p-values are presumably inaccurate, we excluded them from our analysis.

For the remaining 746 gene clusters, the calculated p-values vs. empirical p-values are plotted in Figure 2. Again, we see an excellent agreement between calculated and empirical p-values: Pearson correlation of the log-log pairs is $r = +0.99989$ ($r^2 = 0.999783$), and the coefficient of determination of the log-log pairs is $R^2 = 0.99975$.

Results

To evaluate our method, we used a dataset of 119 bacterial genomes from the RefSeq database [32]. A complete list of the genomes can be found in Additional File 1. This dataset has previously been used by Ling *et al.* [16] for the evaluation of the MCMuSeC software; we removed 14 plasmids. A partitioning of the genes of these genomes into homology families is available based on COG (Clusters of Orthologous Groups) numbers [35]. We deliberately ran our analysis on this set of well-described bacteria to facilitate evaluation of our cluster predictions based on the annotations provided in the database.

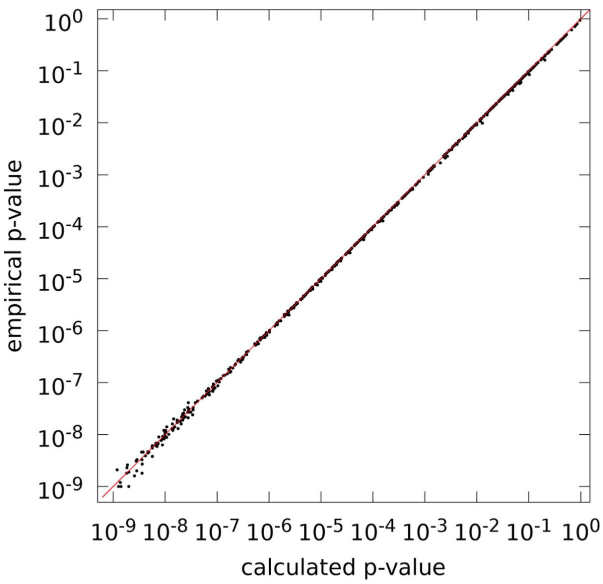

Figure 2 Comparison of empirical and calculated p-values for simulated data. Comparison of empirical and calculated p-values for random gene clusters in random genomes (log-log plot, $N = 746$). Pearson correlation of log-log-pairs is $r = +0.99989$ ($r^2 = 0.999783$), coefficient of determination of log-log-pairs is $R^2 = 0.99975$ (identity in red).

We searched for reference clusters with *Mycobacterium tuberculosis* CDC1551 (refSeq NC_002755) as the reference genome, which contains 4189 genes. We used five different combinations of δ and s, namely ($\delta = 0$, $s = 4$), ($\delta = 1$, $s = 5$), ($\delta = 2$, $s = 6$), ($\delta = 3$, $s = 7$) and ($\delta = 5$, $s = 8$). The quorum parameter was set to $k' = 10$ in each case. Finding the gene clusters took about 9.3 minutes on a laptop computer (run as a single thread on an Intel i5 M520 processor, 2.40 GHz, 8 GB RAM). For multiple genome p-value calculation, we applied the "individual distance bounds" method. Computing p-values for the resulting 582 gene clusters (including duplicates) required 1.2 minutes. The p-values were FDR-corrected for the 8, 775, 955 intervals in the *M. tuberculosis* genome. Gene cluster lists were merged and duplicate occurrences removed. For gene clusters where at least one occurrence, in one of the genomes, intersected, we report only the one with the smaller p-value. This resulted in 63 gene clusters. The best 20 are shown in Table 1; the complete list can be found in Additional File 1. The gene cluster with the best p-value ($3.24 \cdot 10^{-1308}$ after FDR correction) is found in 108 out of the 119 genomes; it contains nine genes with functional annotations linked to the 50S and 30S ribosomal subunits. By contrast, the second most significant gene cluster appears in 114 genomes and shows a much higher degree of conservation with

Table 1 Top 20 gene cluster predictions

			distance to ref.					
ID	**G**	**GN**	**min**	**max**	**avg**	**p-score**	**corr. p-score**	**description**
1	9	108	2	5	2.8	1314.43	1307.49	30S/50S ribosomal subunit
2	7	114	0	3	1.6	1258.12	1251.18	30S/50S, rpoA, infA
3	6	91	0	2	0.7	1031.47	1024.83	ATP synthase
4	9	57	0	5	1.4	896.31	890.57	NADH dehydrogenase
5	8	108	3	5	4.1	716.68	711.29	30S/50S ribosomal subunit
6	8	88	0	5	4.2	569.88	564.63	phosphate ABC transporter
7	8	93	0	5	4.1	486.80	481.67	infB, rfbA, nusA, hypothetical protein
8	8	79	3	5	4.6	367.33	362.27	putative/peptide ABC transporter
9	8	62	3	5	4.4	294.41	289.40	sugar ABC transporter
10	8	65	2	5	4.1	290.24	285.24	N-acetylmuramoyl, cell division
11	4	33	0	0	0.0	272.99	267.55	succinate dehydrogenase
12	8	51	3	5	4.9	221.73	216.79	pdhA/B/C
13	8	48	2	5	4.9	216.54	211.62	ATP-dependent (Clp) protease, trigger factor
14	8	58	0	5	4.2	216.12	211.20	50S L31, prfA, thrA/B/C, rho, hemK
15	8	50	4	5	4.9	213.61	208.70	hisA/C/F/H
16	6	32	0	2	1.7	200.11	194.80	dnaA/N, gyrA/B, recF
17	6	27	1	2	1.7	194.39	189.10	carA/B, pyrC/B/R
18	8	67	4	5	5.0	192.56	187.69	elongation factor Tu, G; 30S S7
19	8	29	4	5	4.5	190.62	185.75	sulfate ABC transporter
20	8	44	2	5	4.3	181.13	176.28	argB/C/D/G/H/F/R

The first 20 gene clusters found when searching *Mycobacterium tuberculosis* CDC1551 against 118 bacterial genomes. Clusters are sorted by p-values, computed using the "individual distance bounds" method. "G" is the number of different genes in the reference gene cluster; "GN" is the number of genomes where the reference gene cluster is found; "distance to ref." indicates the observed distances between the reference gene cluster and its occurrences. The "p-score" is the negative $\log_{10}$ of the p-value, before and after FDR correction.

regards to inserted and deleted genes. However, it contains just seven genes (functional annotation is also 50S/30S ribosomal subunits); its p-value is $6.61 \cdot 10^{-1252}$. The cluster in position four of the list (NADH dehydrogenase) is only contained in 57 genomes. The p-value, of $2.68 \cdot 10^{-891}$, is still very low as it contains nine genes and a low average distance of 1.4.

We have also computed p-values using the "maximum distance bound" method, see Additional file 1. The best scoring cluster, in both cases, is part of the 30S/50S ribosomal subunit, with nine conserved genes. However, using the "maximum distance bound" method, the best scoring occurrence of this cluster is only found in 66 genomes, with a p-value of $1.19 \cdot 10^{-946}$, while the occurrence in 108 genomes only receives a p-value of $1.43 \cdot 10^{-714}$ due to its maximum distance of only five. The p-value for the 66 genomes occurrence using the "individual distance bounds" method is $2.37 \cdot 10^{-947}$, while the 108 genomes occurrence receives a p-value of $3.24 \cdot 10^{-1308}$.

None of the clusters in the above experiments have a corrected p-value anywhere close to 0.05, the typical threshold used to discriminate significant from non-significant observations. To get some insight into this "grey area" where no confident predictions can be made, we studied gene cluster predictions with a corrected p-value close to 0.05. We obtained these in three runs of our program, ($s = 3,\ \delta = 1,\ k' = 9$), ($s = 4,\ \delta = 4,\ k' = 8$) and ($s = 6,\ \delta = 7,\ k' = 7$). For each setting we collected all predictions with a p-value > 0.05 (corrected p-score < -1.3), and also the same number of predictions with the biggest p-values < 0.05. The complete list of these 84 predictions can be found in Additional file 1. We compared the predictions with known E. coli operons that we obtained from the RegulonDB database [36]. As can be seen in Additional file 1 most of the 84 predictions above and below the threshold contain at least one operon. A more formal analysis of these findings is hard to obtain with the limited data available. What appears to be a false positive prediction based on RegulonDB, may in fact be an unknown gene cluster, or one that is not well enough confirmed to appear in the database. Also a non-significant prediction, that is in fact a confirmed operon, does not mean that our p-values are too strict. Statistical significance by itself is simply not a necessary condition for a biological gene cluster.

Conclusion and outlook

In this paper, we presented the first statistical model to estimate the significance of gene cluster predictions under an approximate common interval-based gene cluster model. The underlying p-value calculations integrate all parameters that influence the quality of gene cluster conservation. Namely, the number of genomes in which the gene cluster is detected, the size of the gene cluster, the degree of conservation within the different occurrences, as well as the genome-wide frequencies of the genes involved. To keep the computation time feasible, we had to make some simplifying assumptions in the p-value calculation, but we have experimentally shown that our estimations are remarkably close to empirically derived p-values. An analysis of a set of well annotated genomes has proven that our method is able to rediscover known, highly conserved gene clusters with p-values clearly showing that such conservation did not occur by chance. The gene cluster at position 20 in our output list (functional annotation: arginine biosynthesis) still receives a highly significant p-value of $5.20 \cdot 10^{-177}$. We also studied clusters with low significance and observed known operons with p-values below the significance threshold of 0.05, as well as unknown clusters with significant p-values. However, due to the limited data available it was not possible to distinguish between false and true positives.

Although the simplifying assumptions seem to have little effect in practice, it would be an interesting next step to study more accurate models in the future. In particular, the assumption that the probabilities of observing a cluster are statistically independent in overlapping intervals could be omitted. This could be achieved by accounting for such dependencies in our calculations, for example by employing the inclusion/exclusion principle. Alternatively our present p-value approximation might be amenable to control by the Chen-Stein method [37].

Another strong assumption in our model is that any two genomes show fully randomized gene order, unless evolutionary pressure prohibits it. This assumption may be violated if we analyze genomes of closely related species or strains. In this case, significances will be overrated by our p-value estimation. For the dataset analyzed in this study at least, this problem is less severe than one might expect. Even strains of the same species show a relatively high amount of random shuffling (Additional file 1). Nevertheless, by lowering the quorum parameter to $k' = 5$, we observe that many conserved regions are detected, where all or most species are *Mycobacteria*. In cases where more closely related strains are analyzed, we suggest several workarounds in Additional file 1. In the future, it will be an interesting problem to include incomplete randomization into our statistical model.

Additional material

Additional file 1: (PDF).

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

K.J. and S.W. were supported by DFG grants ST 431/5-1 and BO 1910/8-1, respectively.
Declarations
We acknowledge support for the Article Processing Charge by the Deutsche Forschungsgemeinschaft and the Open Access Publication Funds of Bielefeld University Library.
This article has been published as part of *BMC Bioinformatics* Volume 14 Supplement 15, 2013: Proceedings from the Eleventh Annual Research in Computational Molecular Biology (RECOMB) Satellite Workshop on Comparative Genomics. The full contents of the supplement are available online at http://www.biomedcentral.com/bmcbioinformatics/supplements/14/S15.

Authors' details

[1]Genome Informatics, Faculty of Technology, Bielefeld University, 33615 Bielefeld, Germany. [2]Institute for Bioinformatics, Center for Biotechnology (CeBiTec), Bielefeld University, 33615 Bielefeld, Germany. [3]Chair for Bioinformatics, Friedrich-Schiller-University Jena, Germany.

Published: 13 December 2013

References

1. Tamames J, Casari G, Ouzounis C, Valencia A: **Conserved Clusters of Functionally Related Genes in Two Bacterial Genomes.** *J Mol Evol* 1997, **44**:66-73.
2. Dandekar T, Snel B, Huynen M, Bork P: **Conservation of gene order: a fingerprint of proteins that physically interact.** *Trends Biochem Sci* 1998, **23(9)**:324-8.
3. Overbeek R, Fonstein M, D'Souza M, Pusch GD, Maltsev N: **The Use of Gene Clusters to Infer Functional Coupling.** *Proc Natl Acad Sci USA* 1999, **96(6)**:2896-2901.
4. Huynen M, Snel B: **Gene and context: integrative approaches to genome analysis.** *Adv Protein Chem* 2000, **54**:345-379.
5. Wolf YI, Rogozin IB, Kondraskov AS, Koonin EV: **Genome Alignment, Evolution of Procaryotic Genome Organization, and Prediction of Gene Function Using Genomic Context.** *Genome Res* 2001, **11**:356-372.
6. Yanai I, Mellor J, DeLisi C: **Identifying functional links between genes using conserved chromosomal proximity.** *Trends Genet* 2002, **18(4)**:176-179.
7. Rogozin IB, Makarova KS, Wolf YI, Koonin EV: **Computational Approaches for the Analysis of Gene Neighborhoods in Prokaryotic Genomes.** *Brief Bioinform* 2004, **5(2)**:131-149.
8. Price M, Huang K, Alm E, Arkin A: **A novel method for accurate operon predictions in all sequenced prokaryotes.** *Nucleic Acids Res* 2005, **33(3)**:880.
9. Homma K, Fukuchi S, Nakamura Y, Gojobori T, Nishikawa K: **Gene Cluster Analysis Method Identifies Horizontally Transferred Genes with High Reliability and Indicates that They Provide the Main Mechanism of Operon Gain in 8 Species of γ-Proteobacteria.** *Mol Biol Evol* 2007, **24(3)**:805-813.
10. Heber S, Stoye J: **Algorithms for Finding Gene Clusters.** In *Proc. of Workshop on Algorithms in BioInformatics, WABI 2001 Lect. Notes Comput. Sci. Volume 2149.* Springer, Berlin; 2001:254-265.
11. Béal MP, Bergeron A, Corteel S, Raffinot M: **An Algorithmic View of Gene Teams.** *Theor Comput Sci* 2004, **320(2-3)**:395-418.
12. He X, Goldwasser MH: **Identifying Conserved Gene Clusters in the Presence of Homology Families.** *J Comp Biol* 2005, **12**:638-656.
13. Rahmann S, Klau GW: **Integer Linear Programs for Discovering Approximate Gene Clusters.** In *Proc. of Workshop on Algorithms in BioInformatics, WABI 2006, Lect. Notes Comput. Sci. Volume 4175.* Springer, Berlin; 2006:298-309.
14. Ling X, He X, Xin D, Han J, Han J: **Efficiently identifying max-gap clusters in pairwise genome comparison.** *J Comput Biol* 2008, **15(6)**:593-609.
15. Böcker S, Jahn K, Mixtacki J, Stoye J: **Computation of median gene clusters.** *J Comp Biol* 2009, **16(8)**:1085-1099.
16. Ling X, He X, Xin D: **Detecting gene clusters under evolutionary constraint in a large number of genomes.** *Bioinformatics* 2009, **25(5)**:571.
17. Jahn K: **Efficient computation of approximate gene clusters based on reference occurrences.** *J Comput Biol* 2011, **18(9)**:1255-1274.

18. Durand D, Sankoff D: **Tests for Gene Clustering.** *J Comp Biol* 2003, **10**:453-482.
19. Raghupathy N, Hoberman R, Durand D: **Two plus two does not equal three: statistical tests for multiple genome comparison.** *J Bioinform Comput Biol* 2008, **6**:1-22.
20. Raghupathy N, Durand D: **Gene Cluster Statistics with Gene Families.** *Mol Biol Evol* 2009, **26(5)**:957-968.
21. Calabrese PP, Chakravarty S, Vision TJ: **Fast identification and statistical evaluation of segmental homologies in comparative maps.** *Bioinformatics* 2003, **19(Suppl 1)**:i74-i80.
22. Hoberman R, Sankoff D, Durand D: **The Statistical Analysis of Spatially Clustered Genes under the Maximum Gap Criterion.** *J Comp Biol* 2005, **12(8)**:1083-1102.
23. Wang X, Shi X, Li Z, Zhu Q, Kong L, Tang W, Ge S, Luo J: **Statistical inference of chromosomal homology based on gene colinearity and applications to Arabidopsis and rice.** *BMC Bioinformatics* 2006, **7**:447.
24. Parida L: **Gapped Permutation Pattern Discovery for Gene Order Comparisons.** *J Comp Biol* 2007, **14**:45-55.
25. Yi G, Sze SH, Thomas MR: **Identifying clusters of functionally related genes in genomes.** *Bioinformatics* 2007, **23(9)**:1053-1060.
26. Zeng X, Pei J, Vergara IA, Nesbitt MJ, Wang K, Chen N: **OrthoCluster: A New Tool for Mining Synteny Blocks and Applications in Comparative Genomics.** *Proc of Extending Database Technology, EDBT 2008* 2008, 656-667.
27. Rödelsperger C, Dieterich C: **CYNTENATOR: progressive gene order alignment of 17 vertebrate genomes.** *PLoS One* 2010, **5**:e8861.
28. Proost S, Fostier J, Witte DD, Dhoedt B, Demeester P, de Peer YV, Vandepoele K: **i-ADHoRe 3.0-fast and sensitive detection of genomic homology in extremely large data sets.** *Nucleic Acids Res* 2012, **40(2)**:e11.
29. Didier G, Schmidt T, Stoye J, Tsur D: **Character Sets of Strings.** *J Discr Alg* 2007, **5**:330-340.
30. Benjamini Y, Hochberg Y: **Controlling the false discovery rate: a practical and powerful approach to multiple testing.** *Journal of the Royal Statistical Society. Series B (Methodological)* 1995, 289-300.
31. Šidák Z: **Rectangular Confidence Regions for the Means of Multivariate Normal Distributions.** *J Am Stat Assoc* 1967, **62**:626-633.
32. Pruitt K, Tatusova T, Browen GR, Maglott D: **NCBI Reference Sequences (RefSeq): current status, new features and genome annotation policy.** *Nucleic Acids Res* 2012, **40**:D130-135.
33. Schmidt T, Stoye J: **Gecko and GhostFam - Rigorous and Efficient Gene Cluster Detection in Prokaryotic Genomes.** In *Comparative Genomics, Methods in Molecular Biology. Volume 2.* Humana Press;Bergman N 2007:165-182.
34. Agresti A, Coull BA: **Approximate Is Better than "Exact" for Interval Estimation of Binomial Proportions.** *The American Statistician* 1998, **52(2)**:119-126 [http://www.jstor.org/stable/2685469].
35. Tatusov R, Fedorova N, Jackson J, Jacobs A, Kiryutin B, Koonin E, Krylov D, Mazumder R, Mekhedov S, Nikolskaya A, *et al*: **The COG database: an updated version includes eukaryotes.** *BMC Bioinformatics* 2003, **4**:41.
36. Salgado H, Peralta-Gil M, Gama-Castro S, Santos-Zavaleta A, Muñiz-Rascado L, García-Sotelo JS, Weiss V, Solano-Lira H, Martínez-Flores I, Medina-Rivera A, Salgado-Osorio G, Alquicira-Hernández S, Alquicira-Hernández K, López-Fuentes A, Porrón-Sotelo L, Huerta AM, Bonavides-Martínez C, Balderas-Martínez YI, Pannier L, Olvera M, Labastida A, Jiménez-Jacinto V, Vega-Alvarado L, del Moral-Chávez V, Hernández-Alvarez A, Morett E, Collado-Vides J: **RegulonDB v8.0: omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more.** *Nucleic Acids Research* 2013, **41(Database)**:203-213.
37. Arratia R, Goldstein L, Gordon L: **Two Moments Suffice for Poisson Approximations: The Chen-Stein Method.** *The Annals of Probability* 1989, **17**:18.

doi:10.1186/1471-2105-14-S15-S14
Cite this article as: Jahn *et al.*: Statistics for approximate gene clusters. *BMC Bioinformatics* 2013 **14**(Suppl 15):S14.