A Statistical Framework
for Spatial Comparative Genomics

Rose Hoberman

May 2007

CMU-CS-07-130

School of Computer Science
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Dannie Durand, Chair
Andrew Moore
Russell Schwartz
Jeffrey Lawrence (University of Pittsburgh)
David Sankoff (University of Ottawa)

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

This research was sponsored by NIH grant 1 K22 HG 02451-01, a David and Lucille Packard Foundation fellowship, an AT&T Bell Labs Graduate Research Program grant, and by a Barbara Lazarus Women@IT Fellowship, funded in part by the Alfred P. Sloan Foundation. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, or the U.S. Government.

Keywords: spatial comparative genomics, comparative genomics, gene clusters, max-gap clusters, gene teams, whole genome duplication, paralogons, synteny, ortholog detection

For Derek, who supported me through the many ups and downs of grad school
with unflagging confidence, continual encouragement, and lots of laughter.

Abstract

Comparison of the spatial organization of related genomes reveals a wealth of information about how complex biological systems evolve and function. A fundamental task in spatial comparative genomics is identification of homologous genomic regions, regions that have descended from a common region in an ancestral genome. While closely related regions are characterized by conserved gene content and order, in more distantly related genomes homologous regions will be apparent only as gene clusters, pairs of regions with similar, but not identical, gene content and scrambled gene order. As gene content and order diverge, statistical tests to reject the null hypothesis that these regions share genes by chance become essential.

In this thesis, I provide statistical tests to assess the significance of gene clusters for a variety of biological questions and search scenarios. I present the first formal statistical framework for the max-gap cluster, the most widely used cluster definition in genomic analyses. This framework provides statistical tests for two common search scenarios and facilitates principled selection of parameter values prior to conducting a search for gene clusters.

Second, I propose novel statistical tests for clusters spanning three genomic regions, for two comparative genomics applications: analysis of conserved linkage within multiple species and identification of large-scale duplications. Multi-genome clusters are of increasing importance, yet existing tests focus almost exclusively on pairwise comparisons. My results demonstrate that simultaneously considering information from more than two regions dramatically improves sensitivity over pairwise methods.

Third, I demonstrate the importance of incorporating cluster statistics in algorithms for spatial comparative genomics. Orthologs, genes that descended from a common ancestor through speciation, are the fundamental unit of comparison in many comparative genomics applications. Using my statistical framework for evaluating max-gap clusters, I develop a new method for ortholog prediction based on conserved spatial organization. Using statistical significance to rank conserved patterns makes it possible to accommodate a variety of spatial features in a single framework, yielding a method that can be applied to a broad range of genomic data sets. This flexible framework outperforms current spatial ortholog prediction methods, especially on highly diverged genomes.

Acknowledgments

I would like to express my warmest thanks to my advisor, Dannie Durand, for her encouragement, support, and guidance, and for the countless hours she spent providing detailed feedback on papers, talks, and thesis drafts. I also want to thank David Sankoff for his confidence in me, for supporting me in my career, and for encouraging me to "hurry up and finish." I also wish to acknowledge the other members of my committee: Jeffrey Lawrence, Andrew Moore, and Russell Schwartz, for the valuable feedback they have provided.

Many people have contributed to making my time in graduate school productive and rewarding. I would like to thank everyone in the Durand lab for creating a friendly and stimulating working environment. I am grateful to Sharon Burks, for being always available for any question, no matter how trivial or momentous. I am also indebted to the Toyota Technological Institute at Chicago, which graciously provided me with a desk and computer for writing the final chapters of my thesis. I would like to thank my friends for making my time at CMU so enjoyable: Anna, Chuck, Kathy, Katrina, Leaf, Heather, Martha, Sonya, Spoons, and TianKai. Finally, I would like to thank Derek Raymond Dreyer for all the love, companionship, and culinary kudos he has given me over the past six years.

Contents

Chapter 1

Introduction

Comparative genomics, the analysis and comparison of genomes from related species, is a powerful technique for understanding how complex biological systems evolve and function. Genomes can be compared on a range of scales to ask a variety of questions. Features that have been compared include gene complement, gene order, sequence similarity of both coding and non-coding DNA, and the intron and exon structure of related genes. In this work, I focus on the spatial arrangement of genes within a genome, and use the term spatial comparative genomics to refer to this particular aspect of the field. The analysis of conserved spatial organization can further our understanding of protein function and regulation, functional constraints on genome organization, the rates and patterns of chromosomal evolution, phylogenetic relationships, and how evolutionary processes lead to functional innovation.

Spatial comparative genomics is used to identify homologous[1] features in related genomes, facilitating the transfer of knowledge between organisms [113, 119]. Although increasing numbers of genome sequences are becoming available, most experimental studies are still carried out on a small set of model organisms. By determining how genes and genomic regions of poorly-studied organisms correspond to those of well-studied organisms, knowledge about one species can improve understanding of others. In particular, although humans are among the most well-studied organisms, many types of experimentation cannot be carried out on humans. Thus, transfer of knowledge from model organisms is essential for understanding human biological processes, and developing new disease treatments.

Conserved patterns in spatial organization can also help elucidate protein function and regulation. In bacteria, functionally related genes tend to be spatially clustered on the chromosome. Comparisons of gene order can identify sets of genes whose spatial arrangement is conserved, and that are likely to be functionally related. Unlike sequence or structural homology methods, which primarily provide insight on the biochemical function of a protein, spatial clustering offers evidence of associations between proteins, such as physical interactions, or participation in the same pathway. These types of associations help identify the physiological or cellular role of a protein, complementing information derived from sequence comparisons. In bacteria, conserved gene order and content have been used for prediction of operons [37, 57, 130, 135, 177, 179, 181], horizontal transfers [97], and more generally to investigate the relationship between spatial organization and functional selection [86, 87, 95, 124, 159, 162, 163].

Finally, analyses of spatial organization serve an invaluable role in evolutionary biology. A great deal of spatial comparative genomics methodology has been developed for the study of ancient large-scale or whole

[1] Homology and other biological terms are defined in Appendix A

genome duplication events [4, 53, 54, 110, 148, 150, 174, 175, 180]. Conserved segments between different genomes have been used extensively to reconstruct the history of chromosomal rearrangements and infer an ancestral genetic map for a diverse group of species [18, 42, 51, 115, 116, 129, 142, 149], as well as to provide novel features for new phylogenetic approaches [12, 44, 74, 140, 141, 164].

All of the evolutionary and functional questions described above require the identification of homologous chromosomal segments, chromosomal regions that have descended from the same chromosomal region in an ancestral genome. When comparing two genomes, researchers are generally interested in finding *orthologous* segments, regions that have descended from the same chromosomal region in the genome of the most recent common ancestor (MRCA) of the two species. In other cases, a genome self-comparison is conducted to identify evidence of whole genome or large scale duplication. In this case, chromosomes within a single genome are compared in order to find duplicated, or *paralogous*, segments that derive from the same region in the pre-duplication genome.

Immediately following speciation, offspring genomes have very similar gene content and order. Similarly, a whole genome duplication yields two very similar copies of the ancestral genome, both embedded within a single genome. In both cases, the two genome copies will diverge over time due to a wide range of evolutionary processes acting on the genome at different scales. These processes can radically alter genomic sequence, gene complement, and gene order. On a local scale, genomic sequence evolves through point mutations and small insertions and deletions. Larger scale *genome rearrangements*, such as translocations, transpositions, and successive inversions of large regions of a chromosome, shuffle genes within and between chromosomes, and scramble gene order with respect to the ancestral genome. In addition, the *gene complement*, the set of genes that appear in the genome, will be altered by domain shuffling, horizontal transfer, gene loss, and gene duplication.

As gene content and order diverge, homology can be significantly obscured. It is essential to not only design sensitive search algorithms to identify homologous regions, but to apply statistical tests to show that local similarities in gene content could not have occurred by chance. Although there is a long history of searching for conserved chromosomal regions, there has been very little work on formal statistical models for assessing their significance. This is the problem I address in this thesis.

1.1 Background

In closely related genomes, homologous segments will be characterized by conserved gene order and content, as well as similarity in non-coding regions, allowing them to be identified through direct sequence comparison. However, for more diverged genomes, sequence similarity will only be detectable in regions under selection, such as protein coding regions. Furthermore, over time, successive rearrangements will cause the scrambling of gene order. For comparisons of such diverged genomes, genes are frequently treated as markers, and homologous chromosomal regions are detected by searching for *gene clusters*, pairs of regions with similar but not identical gene content, and possibly scrambled gene order.

To detect distantly related homologous chromosomal segments, it is common to use a map-based approach, in which clusters are detected based on the locations of genomic *markers*, rather than direct comparison of the primary sequence. A marker-based approach to the identification of homologous segments typically involves the following steps:

1. Markers must be mapped to their location in the genome. When the markers are genes and the data

are genomic sequences, this reduces to the problem of gene finding.

2. Homology between markers must be established.
3. A precise cluster definition must be selected, to specify the types of clusters sought, and an algorithm must be developed, to identify such clusters via genome comparisons.
4. Statistical tests must be applied, to ensure that the clusters obtained are not due to chance similarities.

The focus of this thesis is the last step, the development of statistical tests to assess the significance of gene clusters. The design of statistical tests will depend on decisions made in the previous three steps. In the next three sections, I give a brief introduction to existing approaches for each of the first three tasks. Then, at the end of this chapter, I will discuss the implications of all of these choices for the development of formal tests to assess cluster significance.

1.1.1 Marker Identification

Map-based approaches to genome comparison require as input a set of markers, sequences with unique locations in the genome. Frequently, genes are used as markers since their sequences tend to be conserved over long periods of evolutionary time. Also, in many genomic studies, it is genes that are the unit of interest. More recently, other types of markers have also been considered [126, 128, 145]. In this thesis I assume that genes are used as markers, but all of the methods discussed here are general enough to be applied to other types of markers as well.

Maps derived from whole genome data provide a close to complete listing of the location of all genes, although errors in gene finding may occasionally result in markers that do not correspond to protein coding regions. Sequence data also allows the precise order and physical distances between genes to be determined, as well as gene orientation.

Until recently, maps were constructed from genetic linkage data, derived from the statistical analysis of co-occurrence of traits. Unlike markers identified from genomic sequence data, markers in linkage maps represent well-studied genes, for which the existence of corresponding transcripts has been verified. However, linkage maps can be quite sparse, with markers representing only a subset of all genes. Also, linkage maps have low resolution: distances are approximate, gene orientation is unknown, and the respective ordering of nearby genes can not always be determined with certainty. Our current views of comparative spatial genomics, as well as much of the existing models and methodology, are informed by this history. There are many organisms where linkage maps are currently the only type of spatial data available. Thus, the basic genome model used in this thesis is general enough to be applied to both linkage maps and modern genomic datasets.

We assume a genome consists of a single linear unbroken chromosome, represented as a sequence of n genes: G = (g_1,. . .,g_n). The orientation of each gene is ignored. This model assumes that genes do not overlap, and disregards the physical distance between genes. The distance between genes is defined to be equal to the number of genes between them. This model can be advantageous for genomic comparisons because physical distances often differ substantially between organisms. In addition, it eliminates the need to model the variation in gene density that can lead to gene-rich and gene-poor regions of chromosomes. A model based on physical distances would have to take into account the fact that a cluster that is unlikely to appear in a gene-poor region might easily occur by chance in a gene-rich region.

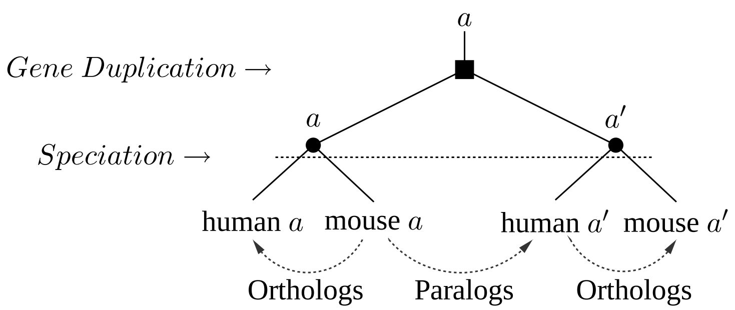


Figure 1.1: A hypothetical gene tree showing the evolution of the a gene family. An ancestral gene a undergoes a gene duplication, giving rise to gene a'. A speciation event occurs, giving rise to human and mouse. Each lineage contains a copy of the a and a' genes. The a gene in human is orthologous to the a in mouse. The a' genes are orthologs as well. All pairs consisting of one a gene and one a' gene are paralogous. All four genes are homologous, as they arose from a single ancestral gene. Together, they form a gene family.

1.1.2 Homology Detection

For genome comparison, once a set of markers is determined for each genome, their homologous counterparts in the other genome must be located. Two genes are homologous if they arose from a single gene in an ancestral genome. Homologous genes are either orthologs or paralogs. Two genes in different species are orthologous if they arose from a single gene in the MRCA of the two species, and paralogous if they arose through a duplication event that preceded the divergence of the species [59, 61]. These relationships are illustrated in Figure 1.1.

In general, common ancestry is inferred from sequence similarity. However, homology identification based on sequence comparison of genes is still an imprecise science. This problem is especially difficult for distantly related proteins, since distinguishing significant sequence similarity in the twilight zone is particularly problematic [86]. Other factors that complicate homology identification include the presence of large families of multi-domain proteins [156], and the difficulty of distinguishing orthologs from paralogs [134, 165, 59, 61]. As a consequence, gene homology identification is an area of active research [21, 26, 33, 40, 92, 147, 176].

For the purposes of this thesis, I assume that homology relationships have already been established, and treat the set of homologous gene pairs as fixed input data. The real-valued similarity scores are discarded, and matches are considered binary, *i.e.* each pair of genes is either considered homologous or non-homologous. The biological definition of homology implies that it should be an equivalence relation. In practice, however, although *detectable* homology relations are generally symmetric, they are rarely transitive. That is, although gene x maybe be similar to gene y, and y is similar to gene z, there may be no detectable similarity between x and z. In general, homology is a many-to-many relationship, but often the data we are given is one-to-one. This is a computational not a biological requirement—many algorithms for finding gene clusters assume a one-to-one mapping between genes. In addition, this restriction is often enforced when the goal is to identify orthologous segments, as allowing only a one-to-one mapping significantly reduces noise in the comparative map [183]. Thus, I assume a model in which a gene has at most one homolog, except in Chapter 4, in which I present a new method for generating a one-to-one mapping from a many-to-many dataset.

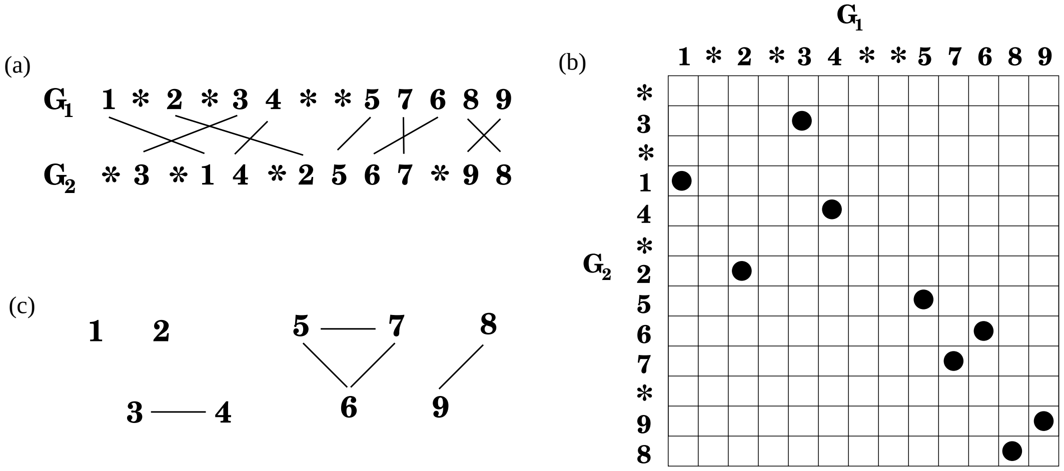


Figure 1.2: Three ways to visualize the comparison of marker order in two related genomes. Integers and stars denote genes, with stars denoting singletons. (a) A comparative map. Lines show the mapping between homologous genes. (b) A dot plot showing the same information in a matrix format. Columns represent genes in G_1 and rows represent genes in G_2. A matrix element is 1 (black circles) if the genes are homologous, and 0 (empty) otherwise. (c) A graph in which vertices represent homologous gene pairs, and edges connect vertices if the corresponding genes are close together in both genomes. In this example, edges connect genes if the sum of the distances between the genes in both genomes is no greater than two.

1.1.3 Cluster Detection

Given the set of genes, their locations, and the homology mapping, the next step is to formally characterize homologous segments. We have an informal notion of the signature of gene clusters: pairs of regions with similar but not identical gene content, and scrambled gene order. In order to construct algorithms to find such clusters, this informal notion of a gene cluster must be defined more rigorously.

The formal characterization of a gene cluster is critical to sensitive detection of ancient homology without inclusion of false positives. Cluster definitions are based on simplified models of real biological processes. In order to be useful, these models must abstract away much of the underlying biology and focus on only a few features of interest. Researchers represent a genome comparison in a number of ways. For example, consider two genomes G_1 = `1*2*34**56789` and G_2 = `*3*14*2567*98`, where the integers correspond to homologous gene pairs, and the stars indicate *singletons*, genes with no homolog in the other genome. Three ways of visualizing the ordering of genes in the two genomes are shown in Figure 1.2. Figure 1.2(a) shows a *comparative map* representation, in which homologous pairs are connected by a line. Alternatively, in a *dot plot* (shown in Figure 1.2(b)), the horizontal axis represents G_1, the vertical axis represents G_2, and homologous pairs are represented as dots in the matrix. Finally, this data can be converted into an *undirected graph* (shown in Figure 1.2(c)), where each vertex v corresponds to a homologous gene pair. Two vertices are connected by an edge if the corresponding genes are close together in both genomes, where "close" is determined based on a user-defined distance function and threshold.

Cluster Definitions and Algorithms

Deciding how exactly to define the structures of interest is one of the most challenging tasks in cluster identification. Cluster definitions can be declarative, specifying precise conditions that allow one to identify a cluster, or they can be constructive, in which an algorithm to find clusters is given, but explicit cluster criteria are not specified. Although a constructive definition makes it clear how to find clusters, it does not necessarily provide information about what the resulting clusters will look like. Unless a formal definition can be abstracted from the algorithm, it can be difficult to reason about these types of models, or to develop formal statistical test for them. A declarative definition, on the other hand, is often easier to reason about, but it requires an additional search procedure to find clusters that satisfy the formal definition. Whether a declarative or constructive definition is used, in both cases, it is necessary to verify that the constructive and formal definitions are equivalent. Recently there has been a movement to formalize cluster definitions, and to develop precisely formulated search algorithms, so that correctness and efficiency of these algorithms can both be analyzed.

The most conservative approach defines conserved segments as *common substrings*, contiguous regions that contain the same genes in the same order, and sometimes orientation [161, 14, 114, 120]. For example, two common substrings can be found in the example genome in Figure 1.2(b): $\{\texttt{6,7}\}$ and $\{\texttt{8,9}\}$. However, such a stringent definition will invariably lead to the exclusion of many regions that did indeed descend from a single ancestral region but have since undergone small rearrangements.

A slightly more liberal approach defines a conserved segment as a *common interval*, a set of genes occurring contiguously in each genome. The order of genes within the cluster may differ from genome to genome. For example, two common intervals can be found in the example genomes in Figure 1.2: $\{\texttt{5,6,7}\}$ and $\{\texttt{8,9}\}$. A number of researchers have developed search algorithms to efficiently find common intervals in genomic data [48, 77, 169]. However, this definition is still generally too strict, since gene duplication and loss are common when comparing distantly related genomes, and a single gene insertion or deletion in one genome will destroy a common interval.

The r-window definition generalizes a common interval, allowing rearrangements, as well as a limited number of insertions and deletions. An r-*window* cluster is defined as a pair of windows, each containing r genes, in which at least k genes are shared [34, 50, 62]. Note that if $k = r$, then an r-window reduces to a common interval of size k. An r-window corresponds to a square in the dot plot with sides of length r, which contains at least k homologs. For example, when $r=5$ and $k=4$, two clusters can be found in the example genome in Figure 1.2(b): $\{\texttt{5,6,7,9}\}$ and $\{\texttt{6,7,8,9}\}$. We distinguish between the genes that appear in both regions that make up the cluster (the “marked” genes) and the intervening “unmarked” genes that occur in only one of the two regions, although they may have a homolog elsewhere in the genome. One limitation of the r-window definition is that it is unclear how to best choose the window size. If the window size is too small, then a cluster may be missed, since it does not fit within the window. If the window size is too large, however, then even if it contains a cluster the window may not be densely populated with homologs, and the cluster may not appear significant. Rather than fixing the window size in advance, we would prefer to allow the window to grow to its “natural size.” In other words, we would like to keep extending the window as long as we continue to find homologs nearby in both genomes.

To gain extensibility, the more general *max-gap* cluster definition has been proposed [10]. It also ignores gene order and allows insertions and deletions, but does not constrain the maximum length of the cluster to r genes. Instead, a max-gap cluster is described by a single parameter g, and is defined as a set of marked genes where the distance (or *gap*) between adjacent marked genes in each genome is never larger than a given distance threshold, g. Note that when $g = 0$, max-gap clusters reduce to *common intervals*. When the maximum gap allowed is $g = 1$, two maximal max-gap clusters are found in the example genome in Figure 1.2(b): $\{$1,2,3,4$\}$ and $\{$5,6,7,8,9$\}$. A max-gap cluster is *maximal* if it is not contained within any larger max-gap cluster. Correct search algorithms for this definition require some sophistication. Many groups design heuristics to find max-gap clusters, but such methods are not guaranteed to find all maximal max-gap clusters. The implications of using these search methods is discussed further in Section 2.3.3. Bergeron *et al.* originally developed a divide-and-conquer algorithm (called GeneTeams) to conduct a whole genome comparison, and efficiently detect all maximal max-gap clusters [10]. This algorithm was later extended by He and Goldwasser [75]. Their HomologyTeams algorithm handles paralogs, and is one of the few algorithms for finding gene clusters in which it is not assumed that the homology mapping is one-to-one.

Other cluster definitions include that of Calabrese *et al.* [30], in which the distance between each pair of homologs is evaluated as a function of the gap size in *both* genomes. Unlike the max-gap definition, which only requires the distance in each genome to *some* other marked gene in the cluster be small, this method requires that all marked genes that are adjacent in genome G_1 *also* be close in genome G_2, but not vice versa. A very different approach by Sankoff *et al.* [143] explicitly evaluates a cluster (or segment) by a weighted measure of three properties: compactness, density, and integrity. They seek a global partition of the genome into segments such that the sum of segment scores is minimized. Clusters have also been defined in terms of graph-theoretic structures (*e.g.* Figure 1.2(c)), such as connected components [128] or high-scoring paths [71, 175]. Finally, a variety of heuristics have been proposed to search for gene clusters [6, 30, 32, 72, 73, 171, 175, 179], the majority of which are specifically designed to find sets of genes in approximately collinear order (*i.e.* forming a rough diagonal on the dot plot). Many constructive definitions give only a vague description of the clustering procedure. Even those that are more precisely specified cannot be easily summarized without describing the full heuristic of each procedure.

Search Strategies

The significance of a gene cluster depends not only on the characteristics of the cluster, but also on how the cluster was found. The larger the search space, the less significant the cluster. Unfortunately, however, most statistical tests do not consider the size of the search space, and most experimental studies present clusters without providing the details of the search procedure that are needed to correctly assess significance. Durand and Sankoff [50] characterized the following three most common search strategies:

1. **Reference set:** Given a set of genes of interest, the goal is to identify subsets of these genes that are located in close proximity in the genome. In this case, the search space is the entire genome. For example, the genes of interest may be located in a particular genomic region (the ”reference region”), and homologous regions, which will presumably contain many of the same genes, are sought. In other cases, the genes of interest share a particular functional or regulatory property, and the goal is to find evidence of functional constraints on spatial organization.

2. **Window sampling:** Given two chromosomal regions, the goal is to determine whether the regions share a significant number of homologs, in order to obtain evidence that they descended from a single region in an ancestral genome. In many cases, these windows are selected because they contain a pair of known homologs of particular interest. This search scenario may be used, for example, to determine whether a particular set of paralogs were duplicated through a large scale event, or to assess whether the gene order around a pair of orthologs has been conserved. In window sampling, the search space

is confined to the two regions of interest.

3. **Whole genome comparison:** Given two genomes, the goal is to identify all clusters of genes that appear in proximity in both genomes. When assessing the significance of individual gene clusters found through whole genome scans, the much larger search space must be taken into account to avoid overestimating cluster significance.

Note that in the reference set search scenario, only a single genome is analyzed. Although the set of genes may have been selected based on their location in a second genome, the search problem is defined with respect to a single genome. In the window sampling and whole genome comparison scenarios, on the other hand, the search problem is defined with respect to two genomes.

1.1.4 Statistical Tests for Gene Clusters

The previous section introduced the basic steps involved in identifying gene clusters, from determining the position of markers in the genome to designing cluster definitions and algorithms. A final critical step in the identification of ancient segmental homologies is significance testing. Over time, processes of genome mutation and rearrangement cause the properties of homologous segments to become more and more similar to the statistical background. Thus, to evaluate putative homologous segments, it is imperative to test and reject the hypothesis that the observed similarities could have occurred by chance.

In general, it is not possible to estimate a clustering algorithm's accuracy, sensitivity or specificity, since in the vast majority of cases the true evolutionary relationships are not known. Synthetic data can be used to evaluate cluster-finding algorithms, but the rates of mutation and rearrangement events are also unknown, and so evaluations based on simulated data are only informative to a limited degree. Thus, statistical tests of cluster significance are critical for accurate identification of ancient segmental homologies.

Statistical models also enable the principled selection of search parameters. Many cluster definitions are based on user-defined parameters. For example, the r-window cluster definition requires the user to specify the window size r. If parameters are selected too conservatively, many significant clusters will not be detected. On the other hand, very liberal parameter values may lead to biologically meaningful clusters being detected but discarded as not statistically significant. A statistical model can be used to determine the range of parameter values within which a cluster will still be significant.

Lastly, formal statistical models allow us to investigate statistical trends for particular cluster models, and ensure that the statistical behavior meets our expectations. For example, a rigorous statistical analysis can show that a cluster definition is inappropriate for certain types of data, or for certain regions of the parameter space. A statistical model is also useful for comparing the power of alternative clustering models under different models of genome evolution.

Related Work

The development of statistical models for gene clusters is largely an uncharted area. The significance of a cluster depends on a broad range of factors, including characteristics of the data and model, the cluster definition, the search procedure, and the biological question of interest. At present, the significance of putative clusters is often not evaluated at all, or only informally. There are three basic approaches to testing the significance of gene clusters: combinatorial analysis, statistical analysis, and analysis by randomization.

These approaches are often complementary, such that improvements in accuracy and efficiency may be obtained when they are used in combination. For the most part, however, existing significance tests are based on data randomization, or on very simple combinatorial models that are applicable to only a limited set of conditions and cluster definitions.

The most common approach is to assess cluster significance with randomization tests. Observed clusters with properties that are rare in randomized data are assumed to correspond to homologous segments. Tests based on randomization are simple to implement for null models of random gene order, since sampling permutations uniformly is quite straightforward. For more complicated null hypotheses, however, randomization tests may be more difficult to design. Furthermore, randomization tests can be computationally expensive. Combinatorial approaches for calculating cluster statistics may help to reduce this running time by specifying a biased distribution for importance sampling [25]. Although the sample space (all possible gene permutations) is very large, only a small fraction of random samples will contain any clusters at all. Combinatorial analysis can be used to devise a sampling strategy that selects samples only from the small fraction of permutations for which the probability of a cluster is high. Finally, randomization studies require complete knowledge of all markers and homologs in the data. When only partial data is available, randomization tests are not feasible.

The only purely statistical approach to assessing cluster significance, of which I am aware, is that of Calabrese *et al.* [30]. The authors present a search algorithm and a statistical model to test putative clusters detected by their algorithm. They define a random variable X_{ij} for each pair of genes (i, j), where X_{ij} is one when the genes are homologous, and zero when the pair is unrelated. Their statistical tests are based on an assumption that the X_{ij}'s are independently distributed, Bernoulli random variables. Under this model, the number of homologs for any given gene i can be described by a random variable $Y_i = \sum_j X_{ij}$. This model implicitly assumes that gene family sizes are binomially distributed (since Y_i is the sum of independent Bernoulli random variables). However, this assumption is not supported by the data. Rather, gene family sizes typically follow a power law: small gene families are most common and large gene families are rare. Thus, it is unclear to what extent this approach allows accurate estimation of cluster significance.

A number of significance tests based on simple combinatorial arguments have been introduced within the methods sections in various papers focusing on the analysis of particular genomic datasets [46, 51, 123, 167, 174, 177]. These tests provide a good starting point, but make so many simplifying assumptions that their descriptive power is limited.

A few more rigorous combinatorial analyses have been made in conjunction with the development of algorithms for cluster identification. In this work the mathematical quantity that is to be estimated is carefully defined, but the connection to the biological question of interest is not addressed, and overly strict simplifying assumptions are made, *e.g.* that the two genomes have identical gene complements [43, 169]. Furthermore, these attempts are generally based on very conservative cluster definitions, such as common intervals and max-gap clusters in which the maximum gap is is at most one [43, 169]. The one combinatorial approach that provides significance tests for a broad range of different biological scenarios was introduced by Durand and Sankoff [50], and later extended by Raghupathy and Durand [132]. While this was the first statistical work in this area to clearly describe both the biological and mathematical problem of interest, a number of open problems remain. In particular, this work is not applicable to the most commonly used cluster definition, the max-gap cluster. Furthermore, the tests are designed almost exclusively for comparisons of two genomic genomes. Designing general statistical tests for clusters spanning multiple regions remains an unsolved problem.

The Design of Statistical Tests for Gene Clusters

Formal statistical models are needed to test the significance of gene clusters for a range of different biological questions of interest. However, translating from a biological question to a formal mathematical statement of the problem is not trivial. In particular, selecting appropriate null and alternate hypotheses, as well as a test statistic, is challenging.

For many of the problems in spatial comparative genomics, how to specify appropriate null and alternate hypotheses is not always obvious. Given a biological question, such as whether local conservation of spatial organization in the genome provides evidence either of shared ancestry or functional selection on gene order, the goal is to show that a cluster with particular characteristics is unlikely to be observed by chance. However, the meaning of “by chance” depends on the particular biological question.

For example, to show evidence that a particular set of genes were duplicated in one large-scale duplication event, it is necessary to demonstrate that the observed cluster is unlikely to be the result of multiple, independent, gene duplications. In this case, if we assume that a duplicated gene is located anywhere in the genome with equal probability, then an obvious null hypothesis is that of random gene order. Although there is some empirical evidence that the destination of single gene duplications tends to be closer to the source than would be expected by chance [175], this process is poorly understood. Furthermore, subsequent rearrangements complicate the picture. Thus, tests of large-scale duplications are generally conducted against the simple null hypothesis of random gene order. If the null hypothesis of random gene order cannot be rejected, no more complex, biologically motivated null hypothesis need be considered. Similarly, when testing for segmental orthology, tests are typically conducted against a null hypothesis of random gene order.

A test statistic should summarize all the properties of the sample that are relevant to the hypothesis being tested; the value of the statistic is then used to decide whether or not the null hypothesis can be rejected. It is difficult to devise a test statistic for gene clusters that captures all properties of interest. Ideally, the number of shared genes, the number of insertions and deletions, and the degree of disorder would all be captured by the test statistic. With clusters that span multiple regions, it is important to consider the number of genes shared by all the regions, as well as the number of genes shared by various subsets of the regions. Given this complexity, selecting an appropriate test statistic is not always straightforward.

Finally, tests of cluster significance need to consider not only the characteristics of the cluster being evaluated, but also the characteristics of the marker and homology data, and the size of the search space. The specific properties of the genomic data, such as the number of genes and the number of homologous gene pairs, must be factored into any model of cluster significance. In particular, as the number of matches between genes increases, so do chance occurrences of gene clusters. The significance of a cluster also depends on the number of possibilities considered during the search. The search space is $O(n)$ for a reference set scenario, but a whole genome comparison, on the other hand, is equivalent to comparing $O(n^2)$ pairs of regions.

Section	Cluster Definition	Sampling strategy	Number of regions	Null Hypothesis
2.2	Max-gap	Reference set	One	Random Gene Order
2.3	Max-gap	Whole genome comparison	Two	Random Gene Order
3	r-window	Window sampling	Three	Random Gene Order

Table 1.1: Overview of statistical tests presented in this thesis.

1.2 Thesis Overview

In this thesis I present statistical tests for two commonly used cluster definitions: max-gap clusters and r-windows. These tests take the number of shared genes into account, but allow insertions, deletions and rearrangements of genes within a cluster. They include models for comparison of two regions and comparison of three regions, and consider several different sampling strategies. These tests are summarized in Table 1.1.

In Chapter 2, I present the first formal statistical model for max-gap clusters. Tests for a reference set scenario are presented in Section 2.2, and probabilities for clusters found through whole genome comparison are derived in Section 2.3. I use the probability expressions derived in Section 2.3 to analyze the significance of clusters found in a whole genome comparison of *E. coli* and *B. subtilis*. I also investigate the impact of the search procedure on the set of max-gap clusters identified on three datasets: *E. coli* compared with *B. subtilis*, human compared with mouse and human compared with chicken.

In Chapter 3, I propose novel statistical tests for r-windows sampled from three distinct genomic regions, including comparisons of three regions selected from three distinct genomes, and comparison of a pair of regions duplicated by whole genome duplication with a reference region selected from a pre-duplication genome. I use the analytical expressions I derive to investigate the impact of the fraction of singletons genes on cluster significance, and to evaluate alternative test statistics. I also compare the sensitivity of these tests with that of existing approaches.

In Chapter 4, I develop a novel method for ortholog prediction based on max-gap statistics. In Sections 4.2 and 4.3 I review existing methods for identifying orthologs, based on sequence and/or spatial information. In Section 4.4, I present my method for ortholog prediction, which includes a new algorithm for finding a particular sub-type of max-gap gene clusters, and a method for statistically validating gene clusters when the homology mapping is many-to-many. In Section 4.6, I present empirical results on a set of γ-proteobacteria, and compare the performance of my method with previous results on this dataset.

Finally, in Chapter 5, I discuss a number of insights that have developed over the course of work, on how to improve upon existing cluster definitions and statistical tests. A number of important open problems raised by this thesis are also described.

Appendix A contains a glossary of technical and biological terms. Appendix B provides a detailed catalog of cluster properties upon which many existing gene cluster definitions, algorithms, and statistical tests are explicitly or implicitly based. Appendix C gives detailed derivations of many of the combinatorial expressions used in Chapter 2.

Chapter 2

Max-Gap Cluster Statistics

In this chapter, I present the first formal, rigorous mathematical model of max-gap gene cluster probabilities [81, 80]. The max-gap definition has been proposed independently several times, and has also been referred to as *gene teams* [10], δ*-teams* [76], and γ*-intervals* [43]. Although the max-gap definition has emerged as perhaps the most popular in empirical studies [11, 20, 37, 62, 102, 110, 124, 151, 162, 171, 175], no formal statistical tests have been developed for max-gap clusters. Studies based on max-gap criteria currently use randomization to estimate the significance of clusters [11, 110, 124, 151, 171, 175]. Analytical statistical models in the literature are designed for other definitions of gene clusters [30, 46, 50, 51, 167, 174]. It is not obvious how to extend them to apply to this commonly used cluster model.

Before presenting the main results of this chapter, I first present some necessary technical preliminaries. After stating formal definitions for a max-gap chain in one genome, and a max-gap cluster in two genomes, I present some general combinatorial expressions that are useful for deriving the results in the following sections. Next I present the first statistical tests for max-gap clusters for two of the basic search scenarios presented in Section 1.1.3.

In the first scenario, we wish to find clusters of a subset of genes that are pre-specified, or *marked*. In the second scenario, we are given two genomes, and a mapping between their homologs, and we wish to identify all sets of genes that are found in spatial proximity in both genomes. In this whole genome comparison problem, the set of genes in a cluster emerges from the comparison of two whole genomes. The window sampling search scenario is not addressed in this section, as it is not compatible with the max-gap cluster definition, which allows the length of a cluster to be arbitrarily large.

For all tests, the null hypothesis is that the genes are randomly distributed in the genome, *i.e.* that each permutation of the n genes is equally likely to occur.

2.1 Technical Preliminaries

As we stated in Chapter 1.1, we model a genome as a sequence of n genes. It is assumed that genes do not overlap, and gene orientation and physical distance between genes is disregarded. This model assumes that the genome consists of a single linear, unbroken chromosome. If a genome contains multiple chromosomes, then we assume they have been concatenated in an arbitrary order to create one long sequence of genes. In this case, our model may slightly overestimate the probability of a cluster since it would erroneously enumerate clusters that span chromosome boundaries. This effect should be small however, as the number of chromosome boundaries is very small compared to the number of genes. If there are circular chromosomes, they can be broken at their origin or terminus. In this case, our model may slightly underestimate the probability of a cluster since it would fail to enumerate clusters that span the origin or terminus. Again, this effect should be small. Our model also assumes that each gene has at most one homolog in the other genome, as discussed in Section 1.1.2.

2.1.1 Max-Gap Terminology

Definition 2.1.1. *A genome $G = \{1, ..., n\}$ is a sequence of genes, ordered by their position in the genome. We define $\Delta(i, j)$, the* ***gap*** *between the i^{th} and j^{th} genes, as the number of genes between them,* i.e. $\Delta(g_i, g_j) = |i - j| - 1$, *if the genes are on the same chromosome, and $\Delta(i, j) = \infty$ if the genes are on different chromosomes.*

We are interested in identifying sets of genes that appear in proximity in the genome, such that each gene in the set is close to at least one other gene in the set, *i.e.* the maximum gap, or *max-gap*, between the genes is small.

Definition 2.1.2. *Let $X = \{x_1, ..., x_m\}$ be a set of m genes in genome G, such that gene x_i precedes x_j in the genome iff $i < j$. Note that X is not required to be a contiguous set of genes, so genes that are adjacent in X are not necessarily adjacent in the genome. We define $\Delta(X)$, the* ***max-gap*** *of a set of genes X, as the maximum gap between adjacent genes in X,* i.e. $\Delta(X) = \max\limits_{1 \leq i < m} \Delta(x_i, x_{i+1})$.

In the reference set scenario, we are given a set of genes of interest (the marked genes), and we wish to determine whether any subset appears in proximity in the genome. We use the term *chain* to describe a subset of genes that are located close together in one genome.

Definition 2.1.3. *We say that X forms a g-**chain** of G if $\Delta(X) \leq g$. A g-chain X is* ***maximal*** *if it is not contained within a larger chain,* i.e. *there is no g-chain $X' \supset X$.*

For example, consider the genome G = `abc*d***ef*`, where stars indicate unmarked genes. If $g = 2$, then $\{a, b, d\}$ forms a g-chain, since neither (a, b) nor (b, d) is separated by more than two genes. However, $\{a, b, d\}$ is not a maximal 2-chain since it is contained within the larger 2-chain $\{a, b, c, d\}$. The set $\{e, f\}$, on the other hand, is a maximal 2-chain.

Definition 2.1.4. *The* ***size*** *of a chain $X = \{x_1, ..., x_m\}$ is the total number of genes it contains: $|X| = m$. The* ***length*** *of X is the total number of genes spanned by the chain: $\Delta(x_1, x_m) + 2$.*

In the example above, the chain $\{a, b, d\}$ is of size three, and length five, whereas the chain $\{a, c\}$ is of size two and length three.

Now that we have introduced the term *chain* to describe a set of genes that are located close together in one genome, we introduce a formal definition of a *cluster*, a set of genes that are located close together in *two* genomes.

Definition 2.1.5. *A set of genes, X, forms a g-***cluster*** in genomes G_1 and G_2 if X forms a g-chain in G_1, each gene in X has a homolog in G_2, and X's homologs form a g-chain in G_2. A g-cluster X is* ***maximal*** *if it is not contained within a larger cluster,* i.e. *there is no g-cluster $X' \supset X$.*

g	maximal g-clusters
0	$\{a, b\}, \{c\}, \{d\}, \{e\}, \{f\}$
1	$\{a, b\}, \{c\}, \{d\}, \{e\}, \{f\}$
2	$\{a, b, d\}, \{c\}, \{e\}, \{f\}$
3	$\{a, b, c, d, e, f\}$

Table 2.1: The max-gap clusters of G_1 = `abc*d***ef*` and G_2 = `abfd***c*de`, for values of g from 0 to 3.

For example, consider the genomes G_1 = `abc*d***ef*` and G_2 = `abfd***c*de`, where each letter corresponds to a homologous gene pair, and the stars indicate singletons. If $g = 2$, then $\{a, b\}$ forms a g-cluster, since a and b are within two genes of each other in both genomes. The set $\{a, b\}$ is not a maximal g-cluster, however, since it is contained within the g-cluster $\{a, b, d\}$. The set $\{b, c\}$ does not form a g-cluster since it does not form a g-chain in G_2. The complete list of maximal g-clusters of G_1 and G_2 is given in Table 2.1, for all values of g.

In this chapter we often assume that g is given, and fixed. In this case, we use the term *max-gap chain*, or even just *chain*, as shorthand for a maximal g-chain, and *max-gap cluster* as shorthand for a maximal g-cluster.

2.1.2 Generalized Dice Equation

Here I introduce several related combinatorial expressions that are used repeatedly in subsequent sections to compute cluster probabilities under a number of different search scenarios. Assume we are given m marked genes, and that these genes are all located within a window of l genes. In the following sections we are interested in three related quantities:

1. the probability of finding all m genes in a max-gap chain of length exactly l,

2. the probability of finding all m genes in a max-gap chain of length no greater than l, given that the first gene in the chain is the first gene in the window, and

3. the probability of finding all m genes in a max-gap chain anywhere within a window of length l.

The number of ways to place the m genes so they form a max-gap chain of length exactly l is equivalent to the number of ways to place m genes in a window of size l, such that they form a max-gap chain, and both the first and last positions contain a marked gene (exemplified in Figure 2.1(a)). In this case, all $m - 1$ gap sizes are constrained to be no more than g, and the gaps must sum to $l - m$. The second problem is similar, except that only the first position must contain a marked gene (Figure 2.1(b)). In this case, in addition to the $m - 1$ constrained gaps, there is also one unconstrained gap after the last marked gene, which can be larger than g. In the third problem, neither endpoint is required to contain a marked gene (Figure 2.1(c)). In this case, there are two unconstrained gaps, one at each end of the chain. The only difference between these three problems is the number of unconstrained gaps.

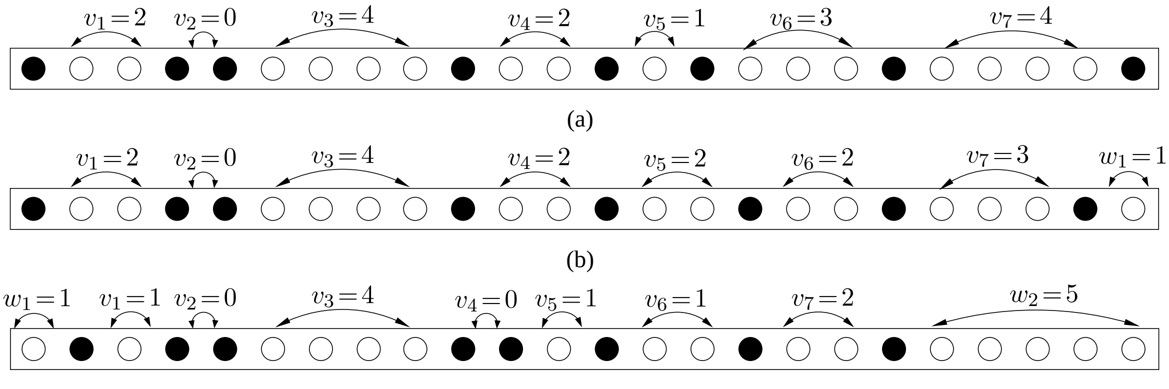


Figure 2.1: Three max-gap 4-chains of size $m = 8$, located in a window of $l = 24$ genes. The window is shown as a rectangle. Genes in the chain (the *marked* genes) are shown as black circles, and all other genes are shown as unfilled circles. The size of each constrained gap v_i in the chain is labeled. The size of each unconstrained gap w_i is also labeled. (a) A max-gap 4-chain of length exactly 24, in which both endpoints of the window contain a marked gene. All gaps are constrained. (b) A max-gap 4-chain of length 23, in which only the leftmost endpoint of the window contains a marked gene. There is one unconstrained gap, after the rightmost marked gene. (c) A max-gap 4-chain of length 18, in which neither endpoint contains a marked gene. There are two unconstrained gaps, one at each end of the chain.

We can formulate all three problems as instances of a more general problem: for a given, non-zero integer s, find the number of solutions to the following equation

$$\sum_{i=1}^{c} v_i + \sum_{j=1}^{u} w_j = s, \text{ such that } 0 \leq v_i \leq g, \forall i \in 1..c \text{ and } w_j \geq 0, \forall j \in 1..u, \tag{2.1}$$

where c is the number of constrained gaps, and u is the number of unconstrained gaps. This problem is a more general version of a well-known problem [170]: determining the number of ways of rolling c dice, each with faces numbered 0 to g, such that the sum of their faces is equal to s. In this generalized version, in addition to having c dice with faces from 0 to g, we also have u “infinite” sided dice, and we wish to know the number of ways of rolling the dice to get a sum of s.

Let $d_g(d, u, s)$ be the number of solutions to Equation 2.1. An expression for $d_g(c, u, s)$ can be derived using recurrence equations (see Appendix C.1):

$$d_g(c, u, s) = \sum_{i=0}^{\lfloor s/(g+1) \rfloor} (-1)^i \binom{c}{i} \binom{s - i(g+1) + c + u - 1}{c + u - 1}. \tag{2.2}$$

Using this equation, we can now give an expression for each of the three probabilities described above. The probability of m marked genes forming a chain of exactly length l is $d_g(m-1, 0, l-m)/\binom{l}{m}$, since the number of constrained gaps is $m - 1$, the number of unconstrained gaps is 0, the gaps must sum to $l - m$, and $\binom{l}{m}$ is the number of ways of placing m genes anywhere withing a window of l genes. The probability that all m marked genes will form a chain of length no greater than l, given that the first gene in the chain is the first gene in the window, is $d_g(m-1, 1, l-m)/\binom{l}{m}$. The probability that the genes will form a chain anywhere within a window of size l is $d_g(m-1, 2, l-m)/\binom{l}{m}$.

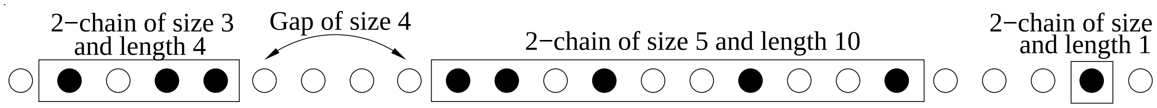


Figure 2.2: A sample genome ($n = 24$), with $m = 9$ marked genes shown in black. Three maximal max-gap chains are found when the maximum gap allowed is $g = 2$. The first has size three and length four, and the second has size five and length ten. The rightmost marked gene forms a trivial chain of size one.

Note that $d_g(m-1, 1, l-m)$ should equal $\sum_{r=m}^{l} d_g(m-1, 0, r-m)$, since the number of ways of getting a chain of no greater than length l is simply the number of ways of getting a chain of length exactly r, summed over all possible values of r from m to l. It is easy to verify that this equivalence holds (see Appendix C.2 for a proof). Similarly, we can show that $d_g(m-1, 2, l-m) = \sum_{r=m}^{l} d_g(m-1, 1, r-m)$.

For certain chain lengths, $d_g(m-1, u, l-m)$ can be reduced to a simpler expression. The maximum possible length of a max-gap g-chain of size m is $L_m = m + g(m-1)$, which occurs when all $m-1$ gaps are of size g. In the case where there are no unconstrained gaps, if $l > L_m$ then $d_g(m-1, 0, l-m)$ is zero, since there is no way to get a chain of length greater than L_m. In the case when there is one unconstrained gap, there is a special case when $l \geq L_m$. In this case, the constraint on the length of the chain is irrelevant, and the problem is much simpler. The number of ways of getting a chain of length no greater than $l \geq L_m$ is $d_g(m-1, 1, L_m - m) = d_g(m-1, 1, (m-1)g)$, which can be shown to be equal to $(g+1)^{m-1}$. This is simply the number of ways of choosing $m-1$ gaps so that the length of each gap is between 0 and g.

It is also useful to observe that $d_g(m-1, 0, l-m)$ is symmetric around $l = m + (L_m - m)/2$, in other words $d_g(m-1, 0, i) = d_g(m-1, 0, L_m - m - i), \forall i \in \{m..L_m - m)\}$ (see Appendix C.3 for a proof). This symmetry can be exploited to compute $d_g(m-1, 0, l-m)$ more efficiently when l is large. A similar symmetry can be exploited to reduce the time required to compute $d_g(m-1, 1, l-m)$ by half, for large values of l. $d_g(m-1, 1, L_m - i - m)$ is the number of ways to generate a max-gap chain of size m and length no greater than $L_m - i$. It can be shown that this is equivalent to $(g+1)^{m-1} - d_g(m-1, 1, i)$. This is the number of ways to generate a chain of size m with any length, minus the number of ways to generate a chain of size m and length no greater than $m + i$.

These expressions will be used in the subsequent sections in various situations in which the length of a chain is constrained. In addition, we will use the generalized dice equation to enumerate arrangements in which there are more than two constrained gaps. When g is fixed, I will use $d(c, u, s)$ as shorthand for $d_g(c, u, s)$.

2.2 Reference Set

In the reference set scenario, the task is to assess whether it is significant to find a particular set of genes clustered together in the genome. We wish to find clusters of a subset of m genes that are pre-specified, or *marked*. These genes may be of interest, for example, because their homologs are contiguous in another region or genome (a "reference region") or because they share some functional properties. We are interested in the probability that all m marked genes, or a sizable subset, appear in close proximity within the genome of interest.

There are many possible tests that could be considered for this problem. Indeed, this problem is very similar to a standard one-dimensional, discrete scan statistic problem, for which many tests have been devised [66, 67]. Since the focus of this chapter is statistical tests for max-gap gene clusters, our tests are based on the maximum gap observed between marked genes. The expressions derived in this Section will also be useful for computing cluster probabilities for the whole genome comparison problem presented in Section 2.3.

We provide two tests of spatial clustering of the reference set of genes. In the first test, the test statistic is the largest gap observed between the marked genes, *i.e.* the smallest value of g for which all m genes form a single g-chain. For example, in Figure 2.2, all $m = 9$ genes form a 4-chain. If the probability of observing a complete 4-chain is small, we will be able to reject the null hypothesis of random gene order. Even if the probability is large, however, there still may be a high degree of clustering of a sub-set of the genes. Thus, we propose a second test in which g is not an observed property of the data, but a parameter selected by the user. With this approach, all maximal g-chains are identified, and the size of the largest maximal g-chain is the test statistic. For example, in Figure 2.2, with a max-gap of $g = 2$, the largest maximal g-chain is of size five. This test may give different results depending on what value of g is selected by the user. If tests are conducted with multiple values of g, then a correction must be applied to the p-values to account for the potential increase in Type I errors.

2.2.1 Exact Probabilities for Complete Chains

In this section, I consider the significance of a *complete* chain, containing all m genes of interest. The test statistic Y in this scenario is the maximum gap between the marked genes. The p-value is the probability of observing a gap between marked genes of no more than g in a random genome: $P_M = P_0(Y \leq g)$. If $P_M < \alpha$, the null hypothesis of random gene order can be rejected at a significance level of α.

Given a random permutation of n genes, we wish to determine the probability of observing all m marked genes (in any order) in a g-chain. The probability is

$$P_M(m, g, n) = N_M(m, g, n) / \binom{n}{m}, \tag{2.3}$$

where $N_M(m, g, n)$ denotes the number of ways to place m marked genes in a genome of size n so that they form a g-chain. Notice that $N_M(m, g, n)$ is precisely the quantity $d_g(m-1, 2, n-m)$ derived in the previous section.

When m and g are not too large (*i.e.* $(m-1)g + m \leq n + 1$), we can express $N_M(m, g, n)$ in closed form. Our approach is to enumerate all possible chains by the position of the leftmost marked gene in the chain. Given the position of the first marked gene, there are $(g+1)^{m-1}$ ways to place the remaining marked genes so that they form a max-gap chain of any length. There are n possible starting positions for the chain. However, $m-1$ of these starting positions are so close to the end of the genome that there will be no room for the remaining $m-1$ marked genes. In addition, $(m-1)g$ of these positions are close enough to the end of the genome so that they can fit only a subset of all $(g+1)^{m-1}$ possible chains. Cumulatively, half of the chains starting at these $(m-1)g$ positions will extend beyond the end of the genome (a proof of this claim is given in Appendix C.3). Combining these terms, the total number of chains is

$$N_M(m, g, n) = \begin{cases} \left[n - (m-1) - \frac{(m-1)g}{2} \right] \cdot (g+1)^{m-1}, & \text{if } L_m \leq n+1, \\ d_g(m-1, 2, n-m), & \text{otherwise.} \end{cases} \tag{2.4}$$

For typical reference set problems, values of g and m are small compared to n, and L_m will be much smaller than the size of the genome, so the closed form expression can be used.

In some cases we may wish to constrain the total length of the chain, by adding the restriction that all m genes must appear in a window of size at most r. The limit on window size ensures a minimum cluster density, while the max-gap property prevents the gaps between marked genes from becoming too large. More formally, given a genome of size n, the probability of finding all m marked genes (in any order) in a window of size at most r, such that the gap between adjacent marked genes is never more than g, is

$$\begin{aligned} P_R(m, g, r, n) &= \frac{1}{\binom{n}{m}} \left[(n - r + 1) \cdot d_g(m - 1, 1, r - m) + \sum_{i=m}^{r-1} d_g(m - 1, 1, i - m) \right] \\ &= \frac{1}{\binom{n}{m}} \left[(n - r + 1) \cdot d_g(m - 1, 1, r - m) + d_g(m - 1, 2, r - 1 - m) \right]. \end{aligned} \tag{2.5}$$

There are $n - r + 1$ positions starting a window of at least r, and one window at the end of the genome of each size from m to $r - 1$.

2.2.2 Exact Probabilities for Incomplete Chains

Requiring all m genes of interest to appear in a single chain is often too strict a requirement. Frequently, only a subset of the m genes of interest are found in close proximity in the genome [2, 45, 55, 65, 83, 90, 91, 101, 103, 127, 136, 154, 157, 167]. For example, in Figure 2.2, when $g = 2$, the marked genes form three maximal g-chains: the first of size $h = 3$, the second of size $h = 5$, and the last of size $h = 1$.

Thus, in this section I provide a statistical test for *incomplete* max-gap chains: maximal g-chains of size $h < m$. In this case, the maximum gap value g is fixed in advance. We search the genome for all maximal chains of marked genes. The test statistic H_{max} represents the size of the largest chain, where the largest chain is the one that contains the most marked genes. The p-value is the probability under the null hypothesis that the largest chain will be of size h or greater: $P_H = P_0(H_{\text{max}} \geq h)$. This is simply the probability of observing *at least* one chain of size h or greater in a random genome.

Dynamic program to compute exact probabilities for incomplete chains when $\mathbf{h \leq \frac{m}{2}}$ Unlike complete chains, there can be more than one incomplete chain of size h or greater in the same genome. A simple extension of Equation 2.4 to incomplete chains would therefore over-count permutations containing more than one chain. Instead, I present a simple dynamic programming algorithm to count those permutations which *do not* contain a chain of size h or greater, and subtract to obtain the probability of observing at least one incomplete chain. The algorithm moves along the genome, adding a marked or unmarked gene at each step. It keeps track of runs of marked genes that satisfy the max-gap chain criterion and avoids creating a chain of size h or greater by judicious placement of unmarked genes.

The quantity $N_{\bar{H}}[n, m, j, q]$ represents the number of ways to place m marked genes in n slots without creating a max-gap chain of size h or greater, where j is the distance to the previous marked gene and q is the size of any chain created so far. It is defined recursively as follows:

$$N_{\bar{H}}[n, m, j, q] = \begin{cases} 0, & \text{if } q = h \text{ or } n < m \\ 1, & \text{else if } m = 0 \\ N_{\bar{H}}[n{-}1, m, j{+}1, q] + N_{\bar{H}}[n{-}1, m{-}1, 0, q{+}1], & \text{else if } j \leq g \\ N_{\bar{H}}[n{-}1, m, j{+}1, q] + N_{\bar{H}}[n{-}1, m{-}1, 0, 1], & \text{otherwise.} \end{cases}$$

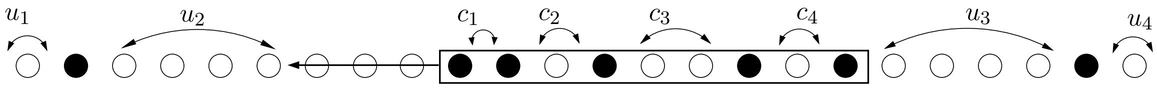


Figure 2.3: An incomplete 2-chain of size $h = 5$ (in a rectangle), located in a genome of $n = 24$ genes, with $m = 7$ marked genes. There are $h - 1 = 4$ constrained gaps, $m - h + 2 = 7 - 5 + 2 = 4$ unconstrained gaps, and one gap of size exactly $g + 1$ (shown by the straight arrow).

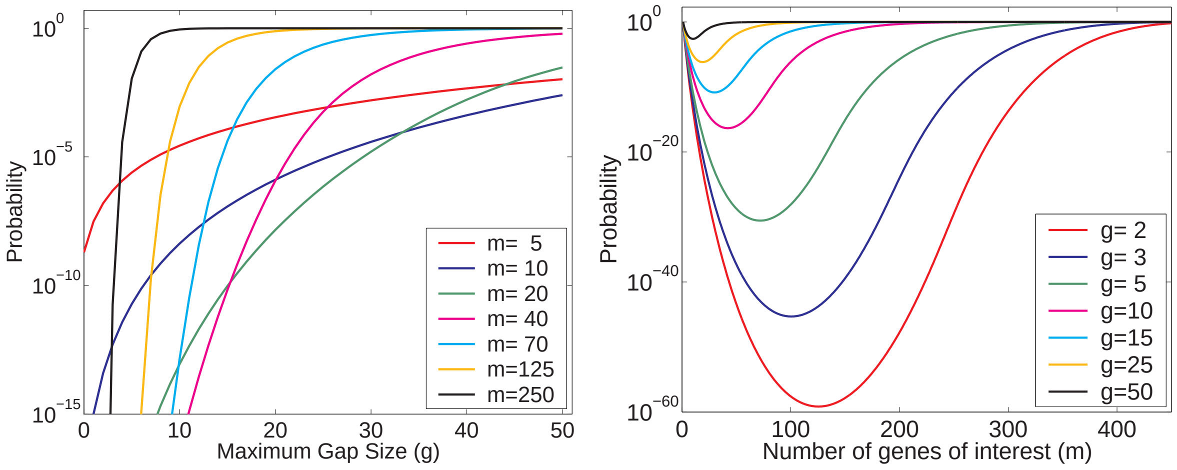
The probability of observing at least one incomplete chain of size at least h is then just one minus the probability that the genome contains no incomplete chains:

$$P_H(n, m, h, g) = 1 - \frac{N_{\bar{H}}[n, m, g + 1, 0]}{\binom{n}{m}}. \tag{2.6}$$

The complexity of computing P_H is $O(nmgh)$. Since $h < m$, this is bounded above by $O(nm^2g)$. However, in practice m will be significantly smaller than n. For example, the size of typical bacterial genomes ranges from 500 to 5000 [153], whereas the average number of genes in an operon is predicted to be between two and four, and the large majority of operons contain fewer than fifteen genes [186]. Vertebrate genomes can be much larger. For example, the estimated size of the human genome is around $25,000$ genes [88], but duplicated or conserved regions reported in the literature tend to include only five to thirty genes in a window containing a hundred genes at most [2, 45, 55, 65, 83, 90, 91, 101, 103, 127, 136, 154, 157, 167]. If we make the conservative assumption that $m \leq \sqrt{n}$ and that g is a small constant, then the running time will be bounded above by $O(n^2)$.

Exact probabilities for incomplete chains when $\mathbf{h > \frac{m}{2}}$ When $m > h > \frac{m}{2}$, the probability can be computed directly because there can be at most one chain of size h or greater, so we do not have to worry about over-counting permutations containing more than one chain. There are $m-h$ marked genes that are not in the chain. These genes can appear to the left or to the right of the chain. We enumerate permutations based on the number of marked genes that appear to the left of the chain. To do this we divide the permutations that contain a chain of size h or greater into $m - h$ disjoint sets. Let E_i represent the permutations containing a chain of size h or greater, such that exactly i marked genes are to the left of the chain, where $0 \leq i \leq m-h$.

The cardinality $|E_i|$ can be computed easily using the generalized dice equation presented in Section 2.1.2. There are $h - 1$ gaps in the chain, each constrained to be no more than g, so $c = h - 1$. The total number of gaps is $m + 1$ ($m - 1$ between the marked genes, one left of the leftmost marked gene, and one right of the rightmost marked gene). Thus, there are $u = m + 1 - (h - 1) = m - h + 2$ unconstrained gaps. When $i = 0$, the constrained and unconstrained gaps together must sum to $n - m$, so $|E_0| = d(h-1, m-h+2, n-m)$. When $i > 0$ we have to ensure that there is a gap of at least $g + 1$ between the chain and the marked gene immediately left of it, as shown in Figure 2.3. Our goal is to enumerate the permutations with i genes to the left of the chain. If there was a marked gene within g, to the left of the chain, then that gene would be part of the chain, and there would only be $i - 1$ genes to the left of the chain. Thus, when $i > 0$ it is necessary to include a gap of size *at least* $g + 1$ immediately left of the chain. The generalized dice equation was only designed to handle gaps with a maximum size, not a minimum size. A gap with a minimum size of $g + 1$ can just be represented as two gaps—one of size exactly $g + 1$, and one unconstrained. Thus, the unconstrained and constrained gaps in this case must sum to $n - m - (g + 1)$, and when $i > 0$, $|E_i| = d(h - 1, m - h + 2, n - m - g - 1)$.



(a) (b)

Figure 2.4: Probability of a complete max-gap chain of m marked genes in a genome of size $n = 500$ (a) as a function of g and (b) as a function of m.

The probability of observing at least one maximal chain of size h or larger is:

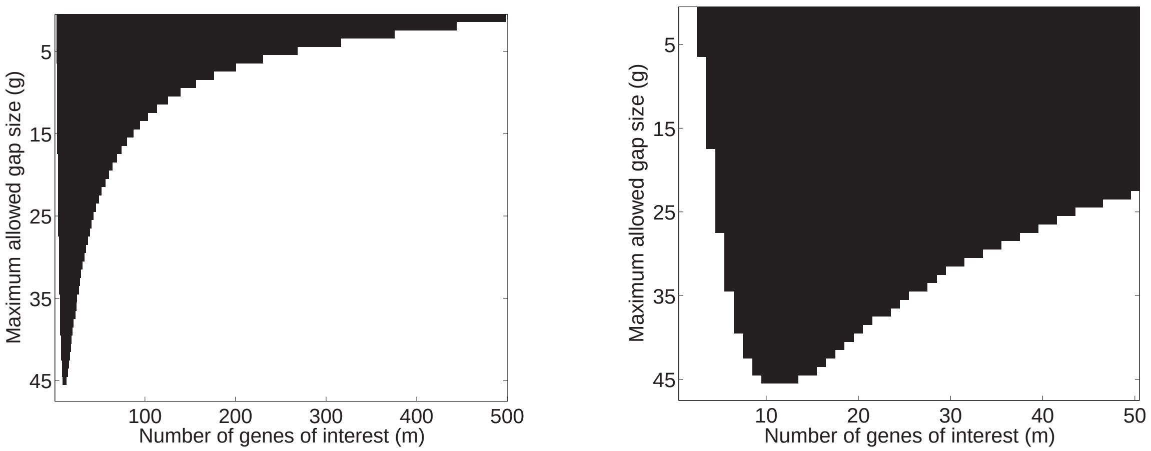
$$\begin{aligned} P_H(n, m, h, g) &= \frac{\sum_{i=0}^{m-h} |E_i|}{\binom{n}{m}} \\ &= \frac{1}{\binom{n}{m}} \left[d(h-1, m-h+2, n-m) \;+\; (m-h) d(h-1, m-h+2, n-m-g-1) \right] \end{aligned} \tag{2.7}$$

This test is based only on the size of the largest chain, and thus may sometimes result in an error of the second kind, *i.e.* it may not reject the null hypothesis of random gene order even though there is significant clustering of the marked genes. For instance, in some cases the probability of observing at least one chain of size h may be too large to reject the null hypothesis, yet the total number of chains will be much higher than expected by chance. It is possible that an alternative test statistic, such as the number of g-chains of size at least h, or the number of marked genes in chains of at least size h, may provide a test of higher power. This is left for future work.

2.2.3 Experiments

The behavior of max-gap cluster statistics for a marked gene scenario was investigated by plotting the probabilities computed by Equations 2.4, 2.6, and 2.7 graphically. I selected parameter values corresponding to the range of values seen in real analyses. For example, I selected values of g ranging from 0 to 50, since typical values of this parameter used in genomic analyses range from three in bacteria [162] to about thirty in human [110]. I calculated probabilities for genomes sizes of 0.5K, 1K, 5K, 20K, and 25K, corresponding to typical gene sets for bacteria, yeast, worm, and higher eukaryotes like human and *Arabidopsis*.

Complete chains The probability of finding a complete chain for varying values of n, m, and g was calculated from Equation 2.4. For complete chains I computed cluster probabilities for all values of m ranging from two to the genome size n.



(a)

(b)

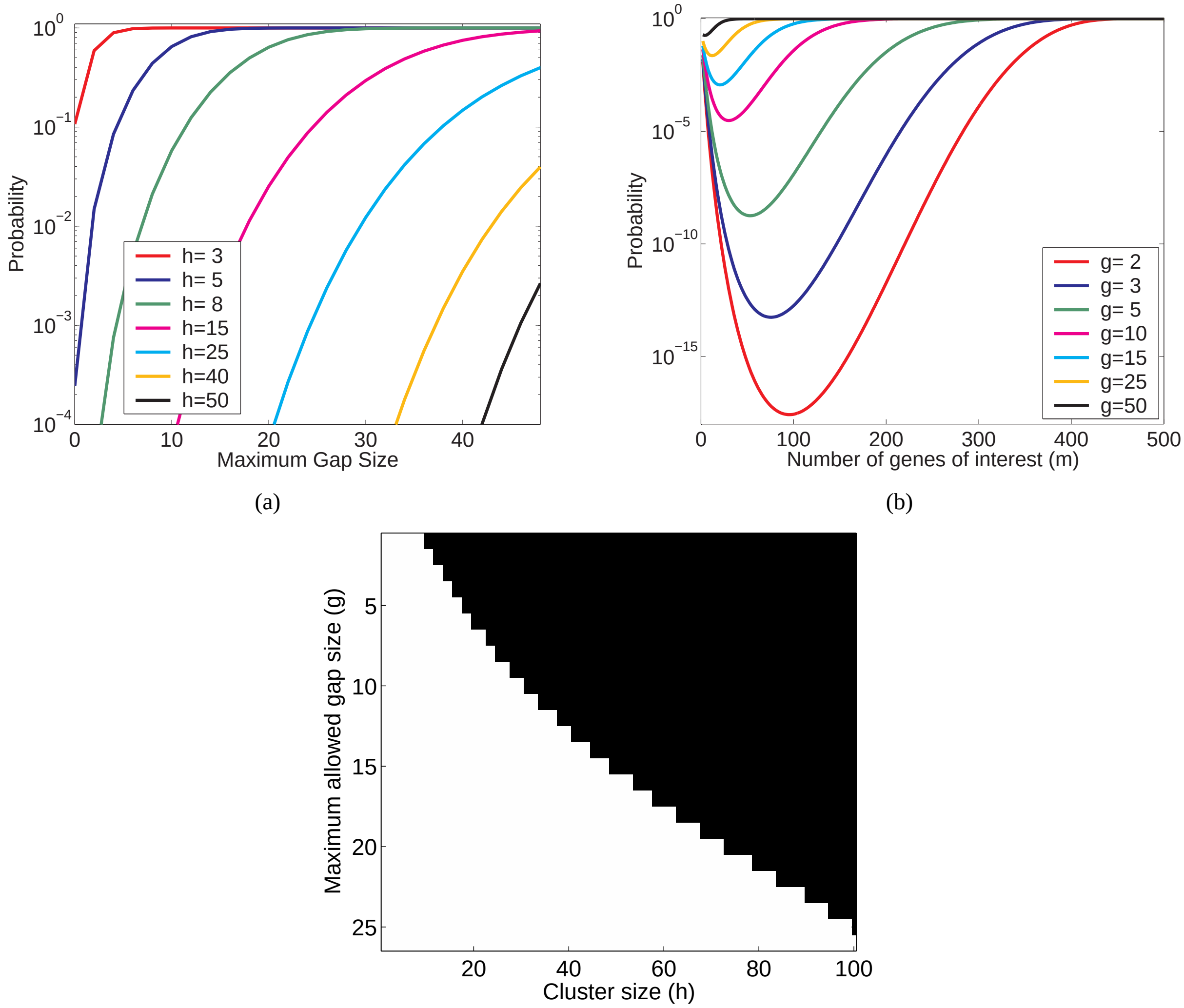
Figure 2.5: Region of the parameter space that is statistically significant (shown in black) at the $\alpha = 0.0001$ level for a complete chain in a genome of size $n = 500$. (a) Complete parameter space where m ranges from 1 to 500. (b) Detail for $m \leq 50$.

Figure 2.4 shows the probability of observing a complete chain containing all m marked genes in a genome of size $n = 500$, as m ranges from 5 to 250 and g increases from 0 to 50. The probability of finding a complete chain increases monotonically with g. We might also expect that this probability will increase monotonically with m, or equivalently, that larger chains will always be more significant, but this is not the case. As Figure 2.4(b) shows, as m increases, the probabilities first decrease and then increase. This makes sense intuitively if ones considers the extreme cases: when $m = 1$ or $m = n$ the probability of finding a complete chain will clearly be one, and the values of m in between these two extremes will have probabilities of less than one. Calculations with larger genome sizes show that as n increases the probabilities decrease but the general trends seen in Figure 2.4(b) remain the same. not shown).

Another question of interest is the range of values of m and g for which it is possible to obtain a significant chain. Figure 2.5 shows the parameter values for which the probability of observing a complete chain in a genome of size 500 is no more than 0.0001. The significant region of the parameter space is shown in black, indicating that as gap size increases, the range of values of m for which it is possible to obtain a significant chain becomes more and more restricted.

Incomplete chains I calculated the probability of finding an incomplete chain from Equations 2.6 and 2.7 for the values of n and g as stated above. I chose to examine values of m ranging from 3 to 250, which covers the range of gene numbers found in typical reference regions of interest (cited above), and values of h ranging from 3 to $m/2$. Figure 2.6(a) shows that as the maximum gap size allowed increases, so does the probability of finding an incomplete chain. Increasing the required size (h) of the chain, on the other hand, decreases its probability of occurring by chance. Figure 2.6(b) shows the probability of max-gap chains for varying values of m, where $h = \frac{m}{2}$. As in the case of complete chains, the probabilities first decrease then increase with m. Probabilities were also calculated for larger genome sizes. Again, as n increases chain probabilities decrease but the general trends are similar (data not shown).

Finally, Figure 2.6(c) shows the parameter values for which the probability of observing an incomplete



(a)

(b)

(c)

Figure 2.6: (a) Probability of an incomplete chain of size at least h when $n = 1000$ and $m = 50$. (b) Probability of an incomplete chain that contains at least half of all m marked genomes when $n = 500$. (c) Region of the parameter space that is statistically significant (shown in black) at the $\alpha = 0.0001$ level for an incomplete chain of $m = 100$ marked genes in a genome of size $n = 1000$.

chain in a genome of size $n = 1000$ with $m = 100$ marked gene is no more than 0.0001. For example, with a maximum gap size of $g = 5$, a chain is not significant until it contains at least 20 marked genes.

2.3 Whole Genome Comparison

In a *whole genome comparison* we are given two genomes, G_1 and G_2, of length n_1 and n_2 respectively, and a mapping between the m homologs shared between G_1 and G_2. We are interested in assessing the significance of a cluster composed of a set of homologs found in proximity in two different genomes, under the assumption that both the homologs and the singletons are randomly distributed throughout the genome.

Recall that we say a set of genes forms a max-gap cluster only if they form a max-gap chain in both genomes of interest, and the cluster is *maximal*, in other words the set of genes is not included within any larger max-gap cluster. The span of two max-gap clusters can overlap, but their gene content will always be disjoint, *i.e.* a gene can be contained in only one maximal cluster.

When $h = m$, the probability of finding a complete max-gap cluster when comparing two genomes of size n_1 and n_2 is $P(n_1, m, g) \cdot P(n_2, m, g)$ where $P(n, m, g)$, defined in Equation 2.4, is the probability of observing a complete chain of m marked genes in a single genome. For whole genome comparison, m is the number of shared homologs. Figure 2.4(b) shows how $P(n, m, g)$ varies as m ranges from 2 to n. Recall that for whole genome comparison the percentage of homologous genes shared between two closely related genomes may be quite high. Thus, squaring the probabilities in Figure 2.4(b) would result in many parameter values for which the probability of a complete cluster will approach one.

To understand this, first consider the simpler case in which the gene sets are identical; e.g.,$m = n$. In this case $P(n, m, g)$ equals one; under a simple model of identical gene content, there will always be a max-gap cluster of size n, since a window that spans the entire genome will contain n genes with no gaps, and the n genes will be identical in both genomes. Even without assuming identical gene content, when mg is large with respect to n we will still be likely to observe extremely large clusters. Indeed, a complete cluster can be found whenever g is greater than the longest contiguous run of singletons. This observation has implications for the design of statistical tests.

Recall that for testing the significance of incomplete chains of marked genes, the p-value is equal to the probability of observing a chain of size h or greater. This conforms to the traditional approach in hypothesis testing of determining the probability under the null hypothesis of obtaining a value of the test statistic that is *more extreme* (e.g. less likely) than the observed value. However, the probability of finding a cluster by whole genome comparison may actually increase with the size of the cluster. For example, as Figure 2.4 shows, the probability of a complete cluster is often greater than 0.5. Whenever this is the case, the probability of observing a cluster of size $m-1$ must be *less than* 0.5. Thus, there is no guarantee that a larger cluster will be less likely to occur by chance, and so a larger cluster is not more “extreme” from a statistical viewpoint. Thus, for whole genome comparison, rather than calculate the probability of finding a cluster of size greater than or equal to h, I determine the probability of a *maximal* cluster of *exactly* size h. I calculate this probability by counting the number of permutations of the n_1 and n_2 genes that result in a max-gap cluster containing exactly h homologs, then divide that by the total number of permutations possible.

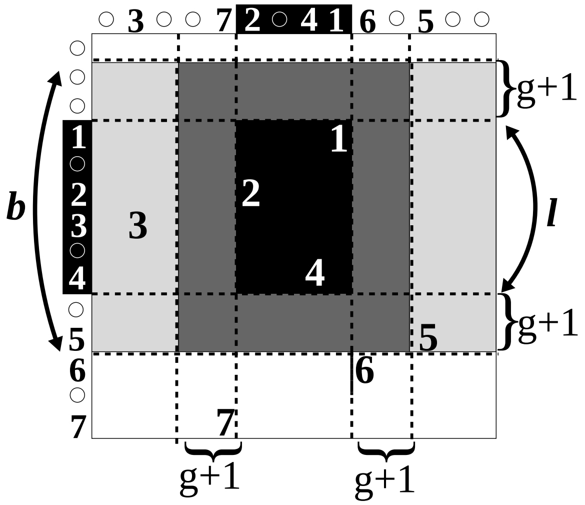


Figure 2.7: A dot plot comparing two genomes—G_1 on the vertical and G_2 on the horizontal axis—that share $m = 7$ homologous gene pairs. Singletons are drawn on the axes as circles, but not shown in the dot plot.

2.3.1 Bounds on Cluster Probabilities

One strategy for counting all permutations that contain a cluster of size exactly h is to first count the ways of creating a cluster of h homologs and then count the number of ways of judiciously placing the remaining $m–h$ homologs so that they cannot extend the cluster to make it larger. The challenge is to determine which regions are "safe" for these $m–h$ *outer* genes.

To determine which regions are "safe" it can be useful to think about a cluster in a two-dimensional space, such as the dot plot in Figure 2.7, where G_1 is on the horizontal axis, G_2 is on the vertical axis, and the cluster is represented in the center. In this example, a non-maximal cluster of size three ($\{124\}$) is contained within a cluster of size five ($\{12456\}$). For a gap size of $g = 1$, how many configurations of the remaining four outer genes are "safe," i.e. do not extend the cluster of size three? Clearly the black rectangle defined by the cluster itself is unsafe, as is the dark gray "moat" of width g around its border, since any gene that lies in these regions will increase the size of the cluster beyond $h = 3$. What about locating a gene within g positions from the cluster in only one of the genomes (e.g. the regions delineated by dotted lines in Figure 2.7)? This region is not necessarily unsafe. For example, gene 7 is within a distance g of the cluster in G_2 yet does not extend the cluster since it is far from the cluster in G_1. On the other hand, though neither genes 5 nor 6 can independently extend the cluster (since each is further than g away from the cluster on one of the genomes), together they successfully extend the cluster of size three to one of size five. Thus it is not clear how to exactly specify the unsafe regions so that we count all valid permutations while at the same time *not* counting those permutations in which the cluster can be extended. Instead, I use the above intuition to devise an upper bound for the probability of finding a shared cluster of size exactly h. The key observation is that an outer gene may be within a distance of g from the cluster in G_1 (like genes 3 and 5), only if its homolog is located at least g genes from the cluster on G_2.

Upper Bound for Incomplete Clusters My upper bound counts the number of ensembles of the m homologs on both genomes which satisfy the following criteria: there exist h homologs that form a chain on both genomes, and there does not exist any other homolog that is within a distance g of the chain on *both* genomes. The key observation is that an outer gene is permitted within a distance of g from the chain in G_1 only if its homolog is located at least $g + 1$ genes from the chain on G_2 (like genes 3 and 5). This strategy is guaranteed to count all permutations that contain a max-gap cluster of size h, but because of its limited look-ahead (as discussed in Section 2.3) it will also incorrectly count some permutations which contain a cluster of size h, but for which that cluster is not maximal (such as the cluster of size three in Figure 2.7). Thus, this approach provides an upper bound on the probability of observing a max-gap cluster of h genes.

Let M be the set of all m homologs shared between the genomes. As stated previously, no gene is permitted in the dark gray region, since any gene in this region will extend the cluster. Thus, the set M can be divided into three subsets corresponding to the three legal regions indicated in Figure 2.7:

1. $H \subset M$ is the set of h homologs that form a chain in both genomes (e.g. the black region in Figure 2.7),
2. $T \subset M{-}H$ is the (possibly empty) set of t homologs that are located within a distance g from the cluster on G_1 but not G_2 (e.g. the light gray regions), and
3. $R = M{-}H{-}T$ is the set of $r{=}m{-}h{-}t$ genes that are *not* within a distance g from the cluster on G_1 (e.g., the unshaded regions).

The upper bound is the number of ways of placing these three subsets of genes on both genomes so that all constraints are satisfied, divided by the total number of ways to place the m homologs. To compute the upper bound on the probability of observing a cluster of size h, we must sum over all possible values of t, which yields

$$P_{up}(h, g, n_1, n_2, m) = \frac{1}{\binom{n}{m}^2} \sum_{t=0}^{\min(m-h,(h+1)g+2)} \frac{h!t!r!}{m!} \cdot q_1 \cdot q_2, \tag{2.8}$$

where q_1 is the number of ways of “safely” placing the genes (according to the constraints on each set) in G_1 and q_2 is the number of ways of “safely” placing the genes in G_2. The factorials account for the different number of ways of ordering the genes within each subset (H, T, and R) versus the unrestricted case in which all m homologs can be permuted indistinguishably. Note that the upper bound on the sum is typically $(h + 1)g + 2$ rather than $m - h$, because when $t > (h - 1)g + 2(g + 1)$ (the maximum number of positions within g of a chain of size h), q_1 will be zero.

Both q_1 and q_2 can be formulated as instances of a more general problem: the number of ways of placing $m = h + y + f + a$ genes in a genome of n genes, such that h genes form a g-chain, y genes are close to the chain (*i.e.*, within g genes), f genes are far from the chain (*i.e.* more than g genes away), and the remaining a genes are anywhere. Let $q[h, y, f, a, n]$ represent this number, then $q_1 = q[h, t, r, 0, n_1]$ and $q_2 = q[h, 0, t, r, n_2]$. To compute $q[h, y, f, a, n]$ we enumerate over all possible values of l, where l is the length of the chain:

$$\begin{aligned} q[h, y, f, a, n] = & \sum_{l=h}^{\min(L_h,n)} \max(0, n{-}l{-}2g{-}1)\; d_g(h-1, 0, l-h) \binom{b-h}{y} \binom{n-b}{f} \binom{n-h-y-f}{a} \\ & + \sum_{i=0}^{\min(g,n-1)} E \cdot d_g(h-1, 0, l-h) \binom{b'-h}{y} \binom{n-b'}{f} \binom{n-h-y-f}{a}, \end{aligned} \tag{2.9}$$

where the length of the chain plus its bounding moats (as shown in Figure 2.7) is given by $b = l + 2(g+1)$, and E is defined below. The last term counts chains within a distance $i \leq g$ of either end of the genome. In this case, the size of the chain plus its bounding moats is $b' = \min(n, l+i+g+1)$. Generally, there are two possible chains of length l within i of either end of the genome: one near the beginning of the genome and one near the end. In this case $E = 2$. However, when $l \geq n - i - g$, the chain spans almost the entire genome, and will be simultaneously close to both ends, so $E = 1$.

When $y = 0$, q can be computed more efficiently, since we do not have to ensure that any genes are close to the chain. We first consider the number of ways of placing the h genes in a chain, and the f genes far away, then multiply this number by $\binom{n-h-f}{a}$, the number of ways of placing the remaining a genes in any of the remaining positions.

The computation is very similar to that for Equation 2.7, except that we are interested in a chain of size *exactly* h, rather than *at least* h. We must ensure that when any of the f far genes are to the right side of the chain, then there is a moat of $g+1$ genes to the right of the chain. We divide the ensembles that contain a chain of size h into $f+1$ disjoint sets. Let F_i represent the permutations containing a chain of size h, such that exactly i of the f homologs are to the left of the chain, and the remaining $f - i$ homologs are to the right of the chain, and none of the f genes are within g genes of the chain. Again, the cardinality $|F_i|$ can be computed easily using the generalized dice equation presented in Section 2.1.2. There are $h-1$ gaps in the chain, each constrained to be no more than g, so $c = h - 1$. The total number of gaps is $h + f + 1$ ($h + f - 1$ between the $h + f$ genes, one left of the leftmost gene, and one right of the rightmost gene). Thus, there are $u = f + 2$ unconstrained gaps. When $i = 0$, all the far genes are to the right of the chain. In this case we have to ensure that there is a gap of at least $g + 1$ between the chain and the gene immediately right of it. In other words, the constrained and unconstrained gaps together must sum to $n - h - f - (g+1)$, so $|F_0| = d(h{-}1, f{+}2, n{-}h - f - (g+1))$. When $i = f$, the calculation is identical. When $0 < i < f$ we have to ensure that there is a gap of at least $g+1$ to the left of the chain, *and* to the right of the chain. In this case, the unconstrained and constrained gaps in this case must sum to $n - h - f - 2(g+1)$, and $|E_i| = d(h-1, f{+}2, n-h-f-2(g-1))$. Putting these terms together, yields:

$$q[h, 0, f, a, n] = \binom{n-h-f}{a} \sum_{i=0}^{f} |F_i|$$

$$= \binom{n-h-f}{a} \cdot [(f-1)d_g(h-1, f+2, n{-}h{-}f{-}2(g+1)) + 2d_g(h, f+2, n{-}h{-}f{-}(g+1))] \tag{2.10}$$

$P_{up}(h, g, n_1, n_2, m)$ can then be computed using Equations 2.9 and 2.10.

Lower Bound for Incomplete Clusters A similar approach can be used to calculate a lower bound on the probability of observing a max-gap cluster of size h, for all $h > \frac{m}{2}$. To compute the upper bound, an outer gene was permitted within a distance g of the chain on G_1 or G_2 but not *both*. However, as explained previously, this constraint on the location of the outer genes is not sufficient to guarantee that the cluster is maximal. For example, both genes 5 and 6 in Figure 2.7 are individually “safe”, but together they extend the cluster. Consequently, the constraint leads to over-counting, and thus the upper bound.

To compute the lower bound we strengthen the constraint so that *no* outsider is allowed within a distance g of the cluster on G_1, regardless of where it is located in G_2. This is unnecessarily restrictive but guarantees that a cluster is maximal. The choice of G_1, however, is arbitrary. A constraint that *no* outsider is allowed within a distance g of the cluster on G_2, regardless of where it is located in G_1 would also guarantee a maximal cluster. My lower bound is the probability of an ensemble that satisfies either of the two constraints above, e.g. the union of the two constraints. By the inclusion-exclusion rule, the union is simply the sum of the probability that each constraint is satisfied minus the probability that both constraints are satisfied.

Assuming equal genome sizes, the first and second scenarios are symmetric, and consequently the probabilities are equal. The probability can be computed by Equation 2.8, replacing q_1 with q_3, defined below. The intersection of the two constraints is not empty, i.e. the two scenarios are not independent; in enumerating all permutations that obey either constraint we will have double counted those permutations in which *no* homolog is within a distance g of the chain in *either* genome. Thus we must subtract out the probability of observing a cluster of size h where there is no homolog within a distance g of the cluster in either genome. This probability can also be computed from Equation 2.8, except we again replace q_1 with q_3, and q_2 is replaced by q_4, also defined below. Combining these two applications of Equation 2.8 yields a lower bound on the probability of observing a cluster of exactly size h:

$$P_{low}(h, g, n_1, n_2, m) = \frac{1}{\binom{n}{m}^2} \sum_{t=0}^{m-h} \frac{h!t!r!}{m!} \; (2 \cdot q_2 \cdot q_3 - q_2 \cdot q_4)\,. \tag{2.11}$$

The expression for q_3 is similar to that for q_1, except the t close genes can no longer appear in the moat on G_1, so the $\binom{b-h}{t}$ and $\binom{b'-h}{t}$ in Equation 2.9 are both replaced by $\binom{l-h}{t}$:

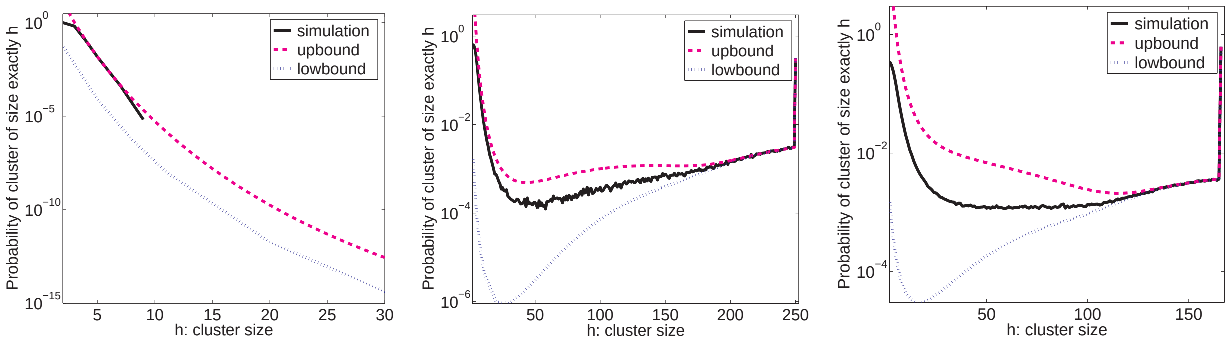
$$\begin{aligned} q_3 = & \sum_{l=h}^{\min(L_h,n)} \max(0, n\text{–}l\text{–}2g\text{–}1)\; d_g(h-1, 0, l-h)\binom{l-h}{t}\binom{n-b}{r} \\ + & \sum_{i=0}^{\min(g,n-l)} E \cdot d_g(h-1, 0, l-h)\binom{l-h}{t}\binom{n-b'}{r}. \end{aligned} \tag{2.12}$$

The expression for q_4 is similar to that for q_2. However, the r genes, rather than allowed anywhere at all, can be anywhere *but the moat*:

$$\begin{aligned} q_4 = (t+1)&\binom{n-h-t-2(g+1)}{r} d_g(h-1, t+2, n-h-t-2(g+1)) \\ +2\sum_{i=0}^{g}&\binom{n-h-t-(g+1+i)}{r} d_g(h-1, t+2, n-h-t-(g+1+i)) \end{aligned} \tag{2.13}$$

In the general case, the moat is of size $2(g+1)$ so we just subtract this in the first binomial. In the two edge cases, in which all the t far genes are on one side of the chain, we now need to know how large the moat is in each case to know how many ways there are to place the r genes so that none falls in the moat. Thus, we sum over $i = 0..g$, where i is the size of the moat left of the chain.

Equation 2.11 is guaranteed to give a lower bound on the probability of observing a cluster of size h for all $h > m/2$. However, when $h \le m/2$, a permutation may contain more than one cluster of size h. The strategy described above enumerates clusters according to their position in the genome, so a permutation with two clusters of size h at different locations will be double counted. As h decreases, the percent of random genomes that contain multiple clusters will increase, and the probability will be correspondingly overestimated. For small values of h, it is possible that the probability computed by Equation 2.11 will actually exceed the true probability.



(a) (b) (c)

Figure 2.8: Comparison of simulation results (solid lines) to upper bound (dashed lines) and lower bound (dotted lines). Probability of finding max-gap clusters of size h when (a) $n=1000$, $m=250$, and $g=10$, (b) $n=1000$, $m=250$, and $g=20$, and (c) $n=500$, $m=166$, and $g=15$.

2.3.2 Experiments

In order to investigate the accuracy of the bounds in different regions of the parameter space, I compared them to the probability of finding max-gap clusters in randomly permuted genomes, estimated through simulation. A number of different parameter values and genome sizes were analyzed. For each set of parameter values, I generated one million random permutations of two genomes, and used the GeneTeams software [10] to find all max-gap clusters. In Figure 2.8, the upper bound (dashed line) and lower bound (dotted line) are compared to the probabilities estimated from the simulations (solid line). Notice that in Figure 2.8(a) the simulated probabilities are only shown for $h \leq 10$ since only one million random trials were generated, and that is the cluster size at which the probabilities drop below 10^{-6}.

First, I considered how the ratio of gap size to genome size affects the accuracy of the bound. As Figure 2.8(a) illustrates, when the maximum gap size is small with respect to n (about 1%), the upper bound is extremely accurate for all values of h. However, when the maximum gap size is larger with respect to n (2% or 3%), then the bounds are only exact when estimating the probability of a large or complete max-gap cluster. This is illustrated in Figure 2.8(c), which shows the behavior of the bounds when $n=500$, $m=166$, and $g=15$. For these parameter values, the bounds are extremely accurate for large values of h, but begin to diverge significantly as h drops below 100. To what extent does the divergence of the upper bound affect the conclusions we may draw about cluster significance? At a significance level of 0.01, for example, the error in the upper bound would lead to the unnecessary elimination of significant clusters of size 8 to 15. At a significance level of 0.001, however, the upper bound could be used to correctly determine that no matter how large the cluster size, the null hypothesis cannot be rejected.

In addition to accuracy, I also considered the monotonicity of the probabilities with respect to cluster size. My analysis shows that, under a null hypothesis of random gene order, the probabilities of observing a max-gap cluster are not always monotonic with respect to cluster size, but often decrease initially and then increase as h approaches m. For example, when $n=1000$, $m=250$, and $g=20$, Figure 2.8(b) shows that the chance probability of observing a cluster of fifty genes is actually smaller than the chance probability of observing a cluster of 100 genes. This non-monotonic behavior can be understood intuitively by observing that, as the size of the cluster increases, the max-gap criterion implicitly increases the maximum allowed window size. As a result, as the size of the cluster sought increases, the probability of observing such a

	Cluster Size				
gap	2-3	4-10	11-26	27-60	> 60
1	108	21	1	0	0
5	112	26	1	0	0
15	144	32	2	0	0
50	165	50	6	2	1
100	0	0	0	0	2

Table 2.2: Number of max-gap clusters of varying sizes shared between *E. coli* and *B. subtilis* for a range of gap values.

cluster may grow substantially as well.

In order to demonstrate the utility of these statistical tests, I conducted a whole genome comparison of the *E. coli* and *B. subtilis* genomes. A mapping of homologs between the two genomes was obtained from a website[1] maintained by A. K. Bansal [6]. The *E. coli* genome has $n = 4108$ known genes and the *B. subtilis* genome has $n = 4245$ known genes. After eliminating ambiguous orthologs, the map yields $m = 1315$ homologous pairs. Using the GeneTeams software [10], I identified all max-gap clusters shared between the two genomes, for values of g ranging from 0 to 110. When $g = 110$, all homologs formed one complete cluster.

A subset of the results selected to show the general trends is shown in Table 2.2. In addition, Figure 2.9 shows the sizes of the clusters found with a range of different gap sizes. The results fall into three regimes. When $g = 0 \ldots 40$, cluster sizes range from two to twelve, except for one larger cluster of size 20 to 30. When $g = 40 \ldots 70$, clusters sizes have a larger range, from two to about 600. Finally, for gap sizes of $g \geq 70$, the homologs form only one or two large clusters.

To assess the accuracy of my upper bound for this bacterial dataset, I again compared it with estimates of the probability of finding max-gap clusters in randomly permuted genomes of the same size, obtained through simulation. I generated one million random permutations of two genomes with $n = 4108$ genes and $m = 1315$ homologs, and again used the GeneTeams software [10] to find all max-gap clusters with gap sizes ranging from $g = 0$ to 100. Figure 2.10 compares my upper bound, calculated from Equation 2.8 (dashed lines), with the probabilities estimated from simulations (solid lines). The accuracy of the bound depends on both h and g. The bound appears to be quite accurate when g is between one and fifteen, but as g becomes larger the bound diverges from the estimated probabilities for small values of h. However, as h approaches m, the bound provides a very accurate estimate of the probability even for large g. Note that although one million random permutations were carried out to estimate the cluster probabilities, clusters of size $20 \leq h \leq 1314$ occurred only infrequently, and thus for $g = 15$ the probability estimates from randomized genomes still have high variance in this region. Although the upper bound appears to drop below the simulated probability for $h = 1312$ and $h = 1306$, this is due to the fact that one million iterates are insufficient to obtain a precise probability estimate in this region of the parameter space.

Since the upper bound is highly accurate for $0 \leq g \leq 15$, it can be used to evaluate the significance of clusters detected through whole genome comparison. If we consider a significance threshold of 0.001, then Figure 2.10 shows that clusters of size three and larger are unlikely to be observed given random gene order when $g = 0$. When $g = 15$, however, only clusters of size seven or larger appear to be significant.

[1] `http://www.cs.kent.edu/~arvind/intellibio/database/orthologs`

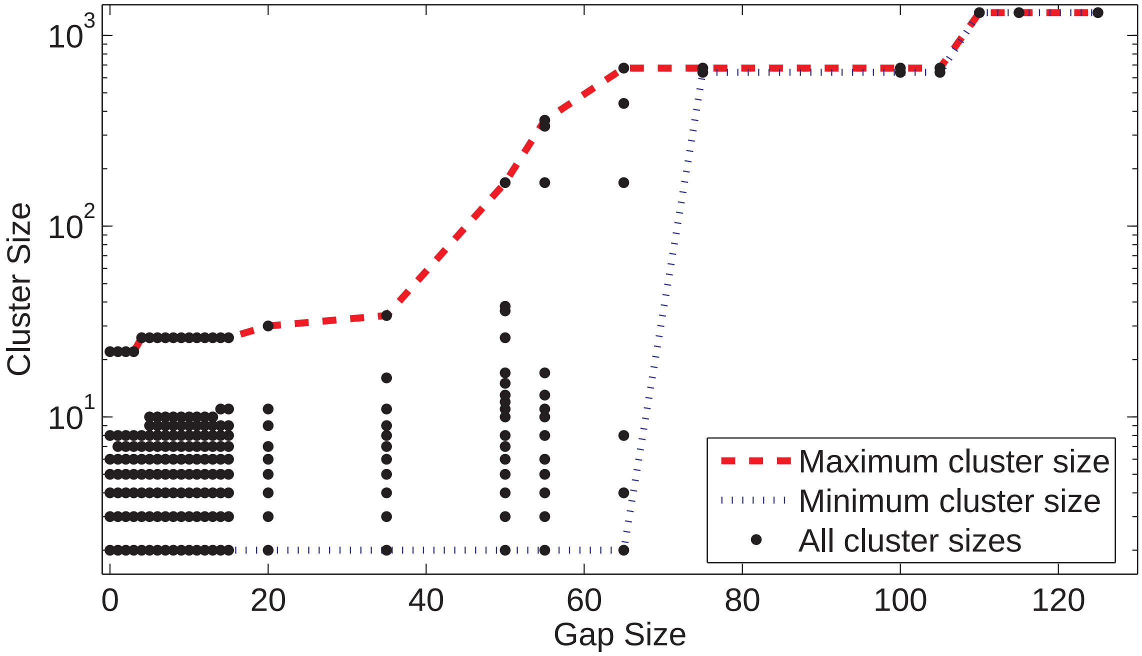


Figure 2.9: The distribution of observed cluster sizes between *E. coli* and *B. subtilis* for g ranging from 0 to 125. The dashed line indicates the largest cluster found and the dotted line indicates the smallest cluster found for each value of g.

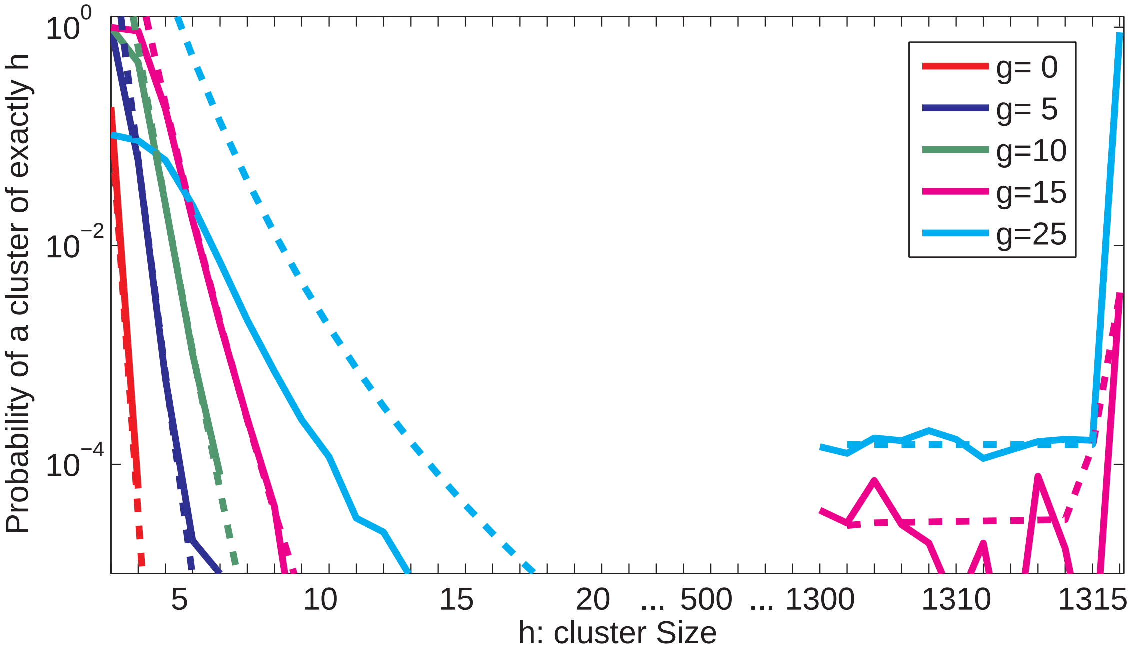


Figure 2.10: Probability of finding max-gap clusters of size h in the *E. coli* and *B. subtilis* genome comparison for $g=0$ to $g=25$. Solid lines show probabilities estimated via randomized genomes and dashed lines indicate the upper bound on the probabilities as calculated by Equation (2.8).

Using these statistics, we find a total of 128 homologs in some significant cluster when $g = 0$, whereas 191 homologs are in a significant cluster when $g = 1$, and only 82 are in a significant cluster when $g = 15$. This suggests that using a gap value of $g=1$ provides more discriminatory power than either $g=0$ or $g=15$ for this dataset.

For $g \leq 40$, most max-gap clusters contain two to ten genes, which corresponds to the range of sizes for typical operons [186]. I compared the clusters to the RegulonDB database of experimentally determined operons in *E. coli* [137], and verified that for gap sizes of zero to ten, over 90% of the clusters are comprised entirely of genes from a single operon. The single large cluster of over twenty genes is composed entirely of ribosomal proteins, which together form the ribosomal "super-operon" in *E. coli*.

An intriguing observation is that the number of large clusters seems to be fewer than expected under the null model. When $g \geq 25$, the model predicts that the probability that all genes will form a complete cluster is close to one. However, a gap size of $g > 100$ is required to obtain a complete max-gap cluster in the bacterial dataset. This discrepancy can be explained by the presence of operons. Since the genes in operons are densely clustered [37], the singletons will be clustered more densely as well. These runs of singletons form large gaps and prevent large clusters from forming as often as they would under a model of random gene order. This is one piece of evidence that the max-gap cluster definition is a good discriminator, since the frequency of both small and large clusters is clearly different than that expected under the null hypothesis, at least for this dataset. In eukaryotes, clusters will generally be due to shared ancestry rather than conserved operons, and so the difference between the observed and predicted cluster sizes may not be so extreme.

The tests developed in this section follow the common practice of using cluster size as the test statistic. Size is the most commonly selected test statistic for a variety of cluster definitions. This choice is based on the natural intuition that the more homologs in a cluster, the lower the probability that it could have occurred by chance, and thus the more confidence we can have that the cluster is truly indicative of common ancestry. For example, the r-window definition constrains the maximum length of a cluster, then evaluates the significance of a cluster according to its size. For r-windows, since the length is constrained, an increase in size corresponds to an increase in global density, which, as shown in Durand and Sankoff [50], does indeed correspond to a reduced probability that such a cluster would occur by chance in randomly ordered genomes.

For max-gap clusters, however, we have demonstrated that the probability of observing a cluster by chance may actually increase with the size of the cluster. Unlike for r-windows, the max-gap definition does not constrain the length of the cluster. This is considered one of the key strengths of the max-gap definition, but it is also a weakness. As the size of the cluster grows, the length of the window containing it is also allowed to grow. Consequently, the probability of observing a max-gap cluster in randomly ordered genomes will often increase as the cluster size increases. We showed that the cluster probabilities under the null hypothesis are not even guaranteed to be monotonic with respect to size: the probabilities may first decrease with size, then eventually begin to increase. Although there is a widespread belief that cluster significance grows with the number of homologs in the cluster, it is critical to recognize that for some cluster definitions, larger clusters do not always imply greater significance. This observation has implications for the design of statistical tests, in particular the choice of test statistic.

In a standard hypothesis test, the p-value is the probability, under the null hypothesis, of obtaining a value of the test statistic that is as extreme or more extreme (*e.g.* less likely) than the observed value. However, if a larger cluster is actually more likely to occur by chance, then a larger value of the test statistic is not more "extreme" from a statistical viewpoint, and such a test is not well-founded. More generally, any model

G_1	G_2	n_1	n_2	m
E. coli	*B. subtilis*	4,108	4,245	1,315
Human	Mouse	22,216	25,383	14,768
Human	Chicken	22,216	17,709	10,338

Table 2.3: The genomes compared (G_1 and G_2), the total number of genes in each genome (n_1 and n_2, respectively), and the number of orthologs identified, excluding ambiguous orthologs (m).

for which the pdf of the test statistic is not unimodal poses difficulties for hypothesis testing. This is not merely an abstract statistical issue, but suggests a failure to accurately capture the full interaction between cluster properties and cluster significance [79]. Thus, before settling on a test statistic, its distribution under the null hypothesis should be investigated. For many of the cluster definitions that have been proposed, there has been little statistical scrutiny. Rarely is the null hypothesis or the test statistic formally stated, and thus it remains to be investigated whether the significance tests being conducting are in fact statistically well-founded.

2.3.3 Are Max-Gap Clusters in Genomic Data Nested?

Cluster definitions that constrain the gap size between marked genes are widely used in genomic studies [6, 11, 20, 37, 102, 110, 124, 151, 162, 171, 175]. An efficient algorithm for finding max-gap clusters (as defined above) via whole genome comparison has been presented by Bergeron *et al.* [10]. However, other groups [7, 11, 30, 32, 37, 73, 82, 110, 124] use a greedy, bottom-up heuristic in which larger clusters are built iteratively from smaller clusters. Each homologous gene pair serves as a cluster seed, and a cluster is extended by looking in its immediate neighborhood for another homologous gene pair close to the cluster on both genomes. In each step, the heuristic "looks ahead" a certain number of positions to see if additional homologs may be added to the clusters without violating the max-gap constraint. It can easily be shown that a simple greedy approach with a look-ahead in either direction of size $g + 1$ will not find all max-gap clusters [10]. For example, given genomes G_1 = `12*34*` and G_2 = `31*4*2`, regardless of the starting point, a greedy approach using a gap size of $g = 1$ will not find the (valid) max-gap cluster {1,2,3,4}. In fact, unless the algorithm "looks-ahead" all the way to the end of the genome, it is not guaranteed to find all max-gap clusters [10].

It is instructive to compare the properties of clusters found by such heuristics with those of *general* max-gap clusters (all clusters that satisfy Definition 2.1.5). Greedy search algorithms implicitly limit the results to nested clusters, where a cluster of size k is *nested* if, for each $h \in 1 \ldots k - 1$, it contains a valid cluster of size h. Intuitively, it may seem that any reasonable cluster definition should have this property. In fact, clusters with no ordering constraints are not necessarily nested, as illustrated in the example above. Nested max-gap clusters comprise only a subset of general max-gap clusters found through whole genome comparison. It can be shown that any greedy search algorithm that constructs max-gap clusters iteratively, *i.e.* by constructing a cluster of size k by adding a gene to a cluster of size $k - 1$, will find *exactly* the set of all maximal nested max-gap clusters, as long as it considers each homologous gene pair as a seed for a potential cluster. In such cases, although order is not explicitly constrained, the search algorithm enforces implicit constraints on gene order: nested clusters can only get disordered to a limited degree. In most cases, however, such constraints are not acknowledged, and perhaps not even recognized.

Such implicit constraints may be particularly problematic when the goal is to characterize the properties

Algorithm 1 A greedy, bottom-up algorithm to find nested max-gap clusters.

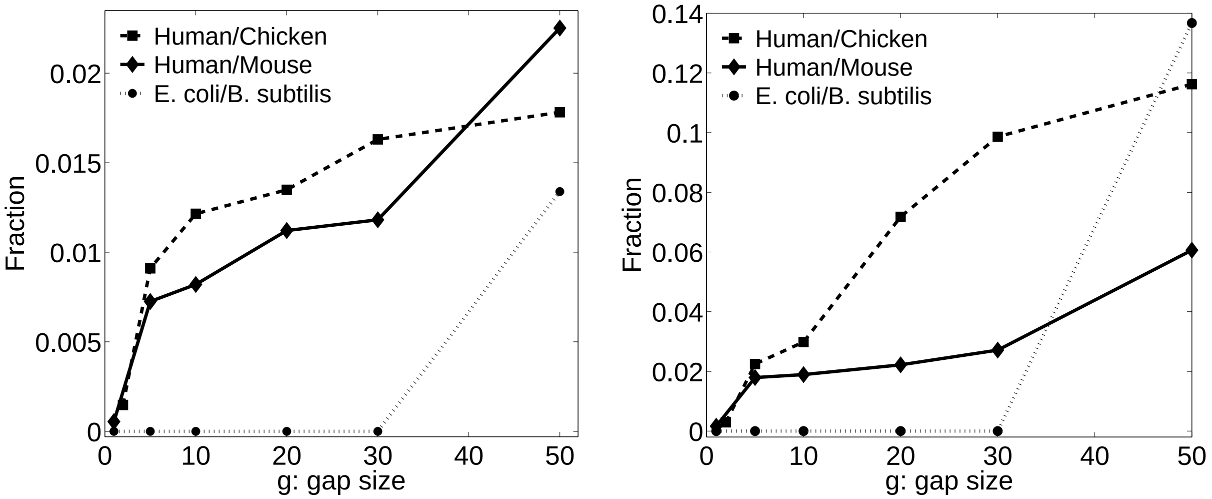
```
clusters $\leftarrow \{\}$
for $i = 1$ to $n$ do                      // $i$ iterates through all genes in $\mathtt{G}_1$
    $\mathtt{C} \leftarrow \{i\}$                        // $\mathtt{C}$ is the cluster being constructed
    $\mathtt{L}_1 \leftarrow \mathtt{R}_1 \leftarrow i$;                 // $\mathtt{L}_i$ and $\mathtt{R}_i$ are the left/rightmost positions of $\mathtt{C}$ on $\mathtt{G}_i$
    $\mathtt{L}_2 \leftarrow \mathtt{R}_2 \leftarrow p(i)$;              //  $p(i)$ indicates the position of gene $i$'s homolog in $\mathtt{G}_2$
    $j \leftarrow \mathtt{L}_1 - g - 1$;                 // $j$ iterates through all genes close to $\mathtt{C}$ on $\mathtt{G}_1$
    while $\mathtt{L}_1$-g-1 $\leq$ j $\leq$ $\mathtt{R}_1$+g+1 do
        if $j \notin \mathtt{C}$ and $p(j) \in \{\mathtt{L}_2 - g - 1, \ldots, \mathtt{R}_2 + g + 1\}$ then        // if $j$ is close to $\mathtt{C}$ in $\mathtt{G}_2$
            $\mathtt{C} = \mathtt{C} \cup \{j\}$;                                         // add the gene to cluster $\mathtt{C}$
            $\mathtt{L}_1 = \min(\mathtt{L}_1, j)$; $\mathtt{L}_2 = \min(\mathtt{L}_2, p(j))$;
            $\mathtt{R}_1 = \max(\mathtt{R}_1, j)$; $\mathtt{R}_2 = \max(\mathtt{R}_2, p(j))$;
            $j = \mathtt{L}_1 - g - 1$;                                          // start the search over
        else
            $j$++
        end if
    end while
    clusters $\leftarrow$ clusters $\cup$ $\{\mathtt{C}\}$
end for
```

of homologous regions. For example, although the CloseUp algorithm was ostensibly designed to identify chromosomal homology using "shared-gene density alone" [73], the greedy nature of the search algorithm means that all clusters with a minimum gene density may not actually be detected. If such an approach was used to evaluate the extent to which order is conserved in homologous regions, incorrect inferences could be made. If clusters with highly scrambled gene order were not found, one might erroneously conclude that no such clusters exist, rather than that the clustering algorithm was simply not capable of finding them. Without a clear understanding of which properties are constrained by the method, and which properties are inherent in the data, it can be difficult to interpret such results.

In this section, we investigate the practical consequences of choosing one search procedure over the other. We compare three pairs of genomes to determine the proportion of max-gap clusters in real genomes that are actually nested. Whole genome comparisons of three pairs of genomes at varying evolutionary distances were conducted. The first comparison was of *E. coli* and *B. subtilis*, with a mapping of orthologs between the two genomes obtained from the GOLDIE database [6]. The other two comparisons were of human and mouse, and human and chicken, with ortholog mappings obtained from the InParanoid database [118]. The total number of genes in each genome, and the number of orthologs identified, is given in Table 2.3.

The GeneTeams software, an implementation of the top-down algorithm of Bergeron *et al.* [10], was used to identify all maximal max-gap clusters shared between the two genomes, for $g \in \{1, 5, 10, 15, 20, 30, 50\}$. In addition, we designed a simple bottom-up, greedy algorithm to identify all maximal *nested* max-gap clusters (Algorithm 1). This algorithm considers each pair of orthologs in turn, treating each as a cluster seed from which a greedy search for additional orthologs is initiated. Occasionally different seeds may yield identical clusters. Any such duplicate clusters are filtered out, as are non-maximal nested clusters (clusters strictly contained within another nested cluster). However, overlapping clusters (*e.g.* properly intersecting sets) are not merged together, since the resulting merged clusters would not be nested.[2]

[2] It is unclear whether those who employ a greedy heuristic merge all overlapping clusters or not, since such heuristics are



(a) (b)

Figure 2.11: Comparison of the set of nested clusters to the set of gene teams, for $g \in \{1, 5, 10, 15, 20, 30, 50\}$. (a) The fraction of gene teams that are *not* nested. (b) The fraction of maximal nested clusters that are *not* gene teams.

For the bacterial comparison, for all gap values except $g = 50$, both methods found the same set of clusters, *i.e.* all gene teams were nested. In all eukaryotic comparisons, however, at least one non-nested gene team was identified. Nonetheless, the percentage of teams that were not nested remained low for all comparisons, ranging from close to 0% to about 2% as the gap size was increased (Figure 2.11(a)). The percentage of nested clusters that were not gene teams (in other words, clusters that could have been extended further if a greedy algorithm had not been used), was also close to zero for small gap sizes, but increased more quickly, peaking at almost 15% for a gap size of $g = 50$ (Figure 2.11(b)). In contrast, in randomly ordered genomes, although large gene-teams are much rarer, a much higher percentage are not nested (data not shown).

Another quantity of interest is the number of *genes* that would be missed altogether if a greedy approach is used rather than a top-down algorithm; that is, the number of genes that are found in a large gene team but not in a large nested cluster. For a minimum cluster size of two, very few genes are missed: the number of genes missed remains under 20 for both eukaryotic datasets, no matter how large the gap size (Figure 2.3.3, circles). For a more realistic minimum cluster size of seven, however, the number of missed genes rises more quickly, peaking near 80 for the human/chicken comparison (Figure 2.3.3, triangles), and near 120 for the bacterial comparison (data not shown).

The gene teams that are not nested tend to be the larger clusters. For example, Figure 2.13 compares the distribution of gene teams sizes to the distribution of non-nested gene teams sizes, for the human/chicken comparison, for the complete set of clusters identified at any gap size. The gene team size distribution peaks very quickly: over 80% of gene teams contain fewer than ten genes. The sizes of non-nested gene teams, however, peak much more slowly: only about 10% of non-nested gene teams contain fewer than ten genes. It is not until the size reaches 270 genes that the CDF reaches 0.8.

generally specified quite vaguely, if at all. In our datasets, only a small percentage of clusters detected with the greedy algorithm overlapped (*e.g.* 2% in the human/chicken comparison).

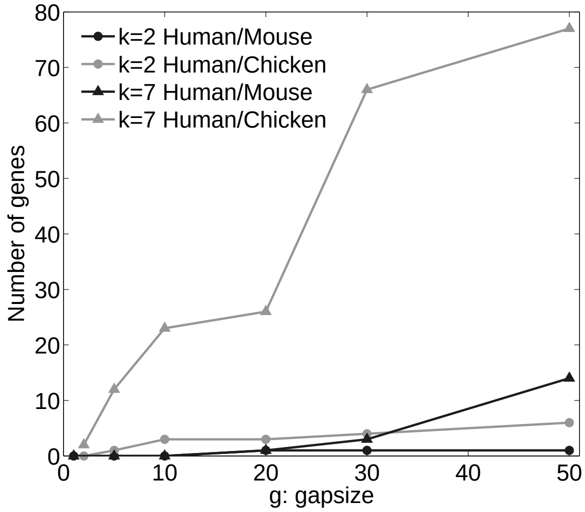


Figure 2.12: The number of genes in a gene team of size $k \geq 2$, that are not in *any* nested max-gap cluster of size $k \geq 2$ (circles). The triangles show the number of genes that would be missed by a nested search when $k \geq 7$.

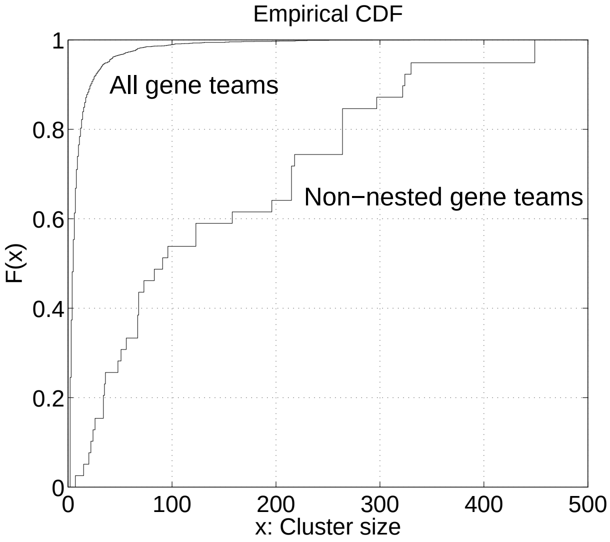


Figure 2.13: A CDF comparing the distribution of gene team sizes to the distribution of nested gene team sizes, for human vs chicken, for all gap sizes tested.

In summary, when comparing *E. coli* with *B. subtilis* with reasonable gap sizes, the nestedness assumption does not exclude any clusters from the data. For the eukaryotic datasets, these results also suggest that for smaller gap sizes few clusters are missed when using a greedy search strategy. For larger gap values, the nestedness assumption does appear to lead to some loss of signal, especially in the human/chicken comparison: large clusters are identified only in fragments, and the spatial clustering of many genes is not detected at all. For more diverged genome pairs, as clusters become more disordered, this loss of signal may be exacerbated. Furthermore, a higher fraction of non-nested clusters may be found when the homology mapping is many-to-many. These questions remains to be investigated, as do the practical implications of the nestedness assumption on the detection of duplicated segments through genome self-comparison.

In Section 2, I presented a statistical model for *general* max-gap clusters identified through whole genome comparison. The results presented there are not applicable to clusters found with a greedy heuristic or for studies in which only nested clusters are of interest. In particular, since nested max-gap clusters are a subset of general max-gap clusters, we expect to find fewer nested clusters than general clusters under the null hypothesis. This is especially true for large clusters. In addition, the enumeration strategy I use to derive statistics relies on the fact that max-gap clusters are disjoint and that gene order is irrelevant. Neither of these properties holds for nested clusters [79]. Statistics for nested max-gap clusters remain an open problem.

The significance of the results reported here goes beyond the vagaries of two competing methods for finding clusters with gaps. Our results also show that, for the datasets considered here, a greedy search strategy for max-gap clusters may actually improve statistical power, at least for small gap sizes. A test of cluster significance will have increased power (*i.e.* a reduced number of false negatives) when the cluster definition is as narrow as possible, while still capturing the properties exhibited by diverged homologous regions. These properties, however, are generally not known, since there is little data about evolutionary histories or processes. In some cases, however, the appropriateness of a particular property can be evaluated even without full knowledge of evolutionary histories [49, 79]. For example, if adding an additional constraint to the cluster definition does not eliminate any of the clusters identified in the data, then I argue that it is not only acceptable to include such a property in the cluster definition, but desirable, in order to increase statistical power. Thus, when comparing *E. coli* with *B. subtilis* with reasonable gap sizes, a nested cluster definition appears to be a good choice: the nestedness assumption does not exclude any clusters from the data, but substantially reduces the probability of observing a cluster by chance, thereby strengthening the statistical significance of detected clusters.

It may be that considering order more explicitly, either in the cluster definition, or in the test, results in additional discriminatory power. Nestedness implicitly enforces order constraints on a cluster, but it is a binary constraint. It may be that this constraint is unnecessarily weak, or unnecessarily strong. Thus, explicitly considering order in the statistical test may be preferable to requiring clusters to be nested. More quantitative measures of order conservation may be found that increase statistical power still further. How to best quantify the degree to which order is conserved, however, remains an open question. A first step in this direction has been taken by Sankoff *et al.* [144], who proposed a number of quantitative measures of gene order. However, analyses comparing the discriminative power of these measures in genomic data have not yet been carried out. How to best quantify and/or constrain the degree to which order is conserved remains an open question.

The use of search heuristics can be particularly dangerous when attempting to draw conclusions about the degree of disorder observed in homologous regions. Researchers may think that they have searched for all max-gap clusters, but by using a greedy heuristic they have implicitly biased their search toward partially ordered clusters, invalidating any conclusions they may draw about conservation of order.

Chapter 3

Cluster Statistics for Three Windows

Existing statistical tests for gene clusters are designed almost exclusively for comparisons of only two genomic regions. With the rapid rate of whole genome sequencing, analysis of gene clusters that span three or more chromosomal regions is of increasing interest. Studies investigating the role of two or more successive rounds of whole genome duplications have searched for multiple homologous regions in the same genome [110, 11, 47]. In addition, a number of methods have been developed for finding sets of clustered genes across multiple genomes [30, 64, 122, 102, 125, 75, 109].

Even when only a pair of regions is under consideration, comparison with additional regions may increase statistical power. In particular, to identify regions duplicated in a whole genome duplication (WGD), comparisons with related genomes may be necessary. Although some evidence of WGD can be found by comparing a genome with itself and looking for pairwise clusters, in many cases duplicated regions may not be identifiable by direct comparison due to *complementary gene loss*: following a WGD, there is no immediate selective advantage for retaining the majority of genes in duplicate, so one copy of most duplicates is lost. As a result, the gene content of duplicated regions is often disjoint, or nearly so.

A solution to this problem is comparison with the genome of a closely related species that diverged shortly before the WGD (a *pre-duplication* species). If two regions in the genome of the *post-duplication* species each have significant similarity to a single region in the genome of the pre-duplication species, they are likely to be homologous even if they share few or no homologous genes. In the example shown in Figure 3.1, the *post-duplication* regions W_{post1} and W_{post2} have only one gene in common. However, they share three and four genes, respectively, with the pre-duplication region W_{pre}. The strategy of comparison with a pre-duplication genome enables the identification of duplicated regions, even when they share no genes. It has been successfully employed to analyze duplications in fish [89], plants [96, 172, 173] and several yeast species [93, 146]. However, statistical analyses for this approach have relied solely on sequential pairwise tests. Statistical tests designed for three regions have the potential to detect more highly diverged duplicated regions, but are also more difficult to design.

In this chapter, I present statistical tests for three regions, developed in collaboration with Narayanan Raghupathy [133]. These tests are based on the r-windows model introduced in Section 1.1.3 and assume a window sampling search strategy. This approach is exemplified in Figure 3.2(a) which shows comparison of two chromosomal regions, or *windows* of adjacent genes, (W_1 and W_2). The number of shared homologs (y_{12}, shown in Figure 3.2(b)) is typically used as the measure of similarity. However, this pairwise approach cannot be directly extended for tests of clusters composed of more than two windows. When comparing three

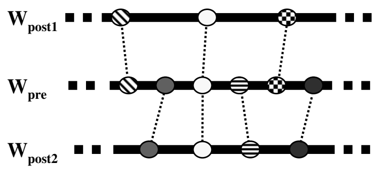


Figure 3.1: A gene cluster spanning three regions with complementary gene loss. Genes are represented as circles. Homologous gene pairs are connected by dotted lines. Intervening genes with no homologous match within the regions are indicated by black circles. The window W_{pre} is sampled from a pre-duplicated genome G_{pre} and the two regions W_{post1} and W_{post2} are sampled from a post-duplicated genome G_{post}. Only the white gene has been retained in duplicate. The remaining genes in W_{pre} occur only once in G_{post}.

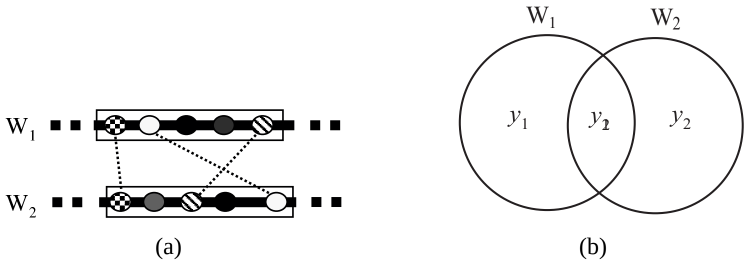


Figure 3.2: A pairwise gene cluster and its Venn diagram representation. (a) A gene cluster of two windows, W_1 and W_2, of size $r_1 = r_2 = 5$, which share $y_{12} = 3$ homologous genes. Genes are represented as circles. Homologous gene pairs are connected by dotted lines. Intervening genes with no homologous match within the regions are indicated by black circles. (b) The Venn diagram representation of the pairwise comparison of W_1 and W_2, which share y_{12} homologous genes.

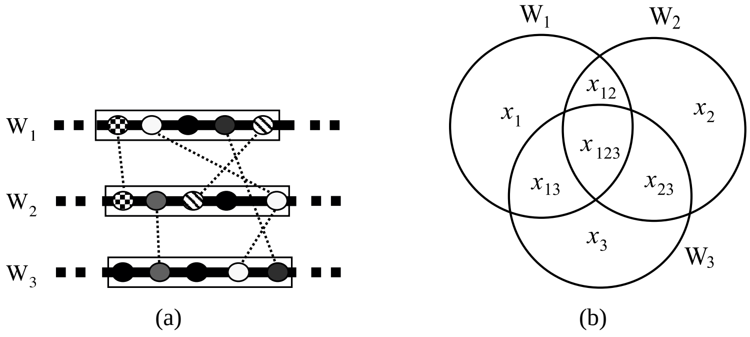


Figure 3.3: A three region gene cluster and its Venn diagram representation. (a) A gene cluster of three windows W_1, W_2, and W_3, in which $x_{123} = 1, x_{12} = 2, x_{13} = 1$ and $x_{23} = 1$ homologs are shared between the three windows. Genes are represented as circles. Homologous gene pairs are connected by dotted lines. Intervening genes with no homologous match within the windows are indicated by black circles. (b) The Venn diagram representation of the three-way comparison of W_1, W_2, and W_3, in which x_{123} homologs appear in all three windows. The variables x_{ij} represent the number of genes that only appear in W_i and W_j, and x_i represents the number of genes that only appear in a single window, W_i.

windows (W_1, W_2, and W_3 in Figure 3.3(a)), there are many more quantities to consider (Figure 3.3(b)): the number of homologs observed in all three windows (x_{123}), the number of homologs observed in each pair of windows (x_{12}, x_{13} and x_{23}), and the number of genes observed only in a single window (x_1, x_2, and x_3). Evidence for homology comes not only from the set of x_{123} homologs that appear in *all* the windows being compared, but also from the number of homologs that are shared by a subset of the windows (the $x_{ij}'s$, which we refer to collectively as the *pairwise overlaps*). How best to combine evidence from different subsets of windows remains an unsolved problem.

In the first attempt to address this issue, we consider the problem of clusters spanning exactly three regions. Given a set of three windows sampled from three genomes, each containing r consecutive genes, we wish to determine whether the windows share more homologous genes than expected by chance. (If duplications are under consideration, the windows may be sampled from non-overlapping regions of a single genome.) This problem, while restricted to three windows, exhibits the basic challenges that arise in the more general problem of clusters spanning $k \geq 3$ windows.

In this chapter, we develop the first statistical tests that consider both x_{123} *and* the $x_{ij}'s$ simultaneously. We obtain expressions for the probability—under the null hypothesis of random gene order—that the number of shared genes is at least as large as the number observed. These expressions are derived for genome models that are appropriate for two common comparative genomics problems: (1) analyses of conserved linkage groups in three regions from three genomes, and (2) identification of segments duplicated by a whole genome duplication, via comparison with the genome of a related, pre-duplication species. We show through simulations that our tests for comparing three regions are more sensitive than existing approaches, and have the potential to detect more diverged homologous regions.

3.1 Related Work

Durand and Sankoff [50] were the first to formally characterize the probability of a cluster in multiple genomes. They derived an expression for the probability that in at least N' of N' genomes there is a window of size r containing at least h of m genes of interest. In this scenario, the m genes of interested are pre-specified. The subset of m that appears in each window can differ, but the subset of genes that appear in more than one window, or even all the windows, is not given additional weight.

Here we consider the following more general: Given three distinct genomic regions of interest, possibly from multiple genomes, devise a test that considers all evidence that these regions are homologous. There are three existing approaches for determining whether the number of genes shared by three regions is statistically significant. Our Venn diagram model (Figure 3.3) can be used to compare these approaches and succinctly illustrate the differences between them. We first introduce some notation. Consider three windows W_1, W_2, and W_3, of length r_1, r_2, and r_3, sampled from three non-overlapping genomic regions. Let $y_{12} = x_{123} + x_{12}$ be the total number of genes shared between window W_1 and W_2. Note that y_{12} includes the genes that are shared by *all* three windows. Similarly, $y_{13} = x_{123} + x_{13}$, and $y_{23} = x_{123} + x_{23}$. The random variable Y_{12} represents the number of homologs shared between two windows of size r_1 and r_2, under the null hypothesis. Y_{13} and Y_{23} are defined analogously.

In order to determine the significance of gene clusters, the goal is to select a test statistic that captures the essential properties of the clusters of interest. For example, when comparing two windows of size r_1 and r_2, the test statistic is typically y_{12}, the number of homologs shared between the two windows. Significance is demonstrated by showing that $P(Y_{12} \geq y_{12})$ is small, under the null hypothesis. In contrast, when comparing three windows it is less obvious how to choose an appropriate test statistic.

The most common strategy for testing significance of multiple regions is to conduct multiple pairwise comparisons (reviewed by Simillion *et al.* [152]). A cluster is considered significant if region W_1 is significantly similar to W_2, and W_2 is significantly similar to region W_3. In this case, homology between all three regions is inferred, even if W_1 and W_3 share few genes. Using the notation from our Venn diagram model, we can express this formally: a cluster is significant at the α level when

$$P(Y_{12} \geq (x_{123} + x_{12})) \leq \alpha \text{ and } P(Y_{13} \geq (x_{123} + x_{13})) \leq \alpha. \tag{3.1}$$

Here the test statistics are Y_{12} and Y_{13}. This approach allows the use of existing statistical methods designed for comparing two regions. However, this strategy is conservative as it will only identify a three-way cluster if at least two of the three pairwise comparisons are independently significant.

In a second approach, once a significantly similar pair of regions (W_1 and W_2) is identified, the genes in these regions are merged to approximate their common ancestral region [152]. Then a second pairwise test is conducted, in which the third region of interest is compared to this inferred ancestral segment. With this approach, a cluster is significant when

$$P(Y_{12} \geq (x_{123} + x_{12})) \leq \alpha \text{ and } P(Y_{1\cup 2,3} \geq (x_{123} + x_{13} + x_{23})) \leq \alpha, \tag{3.2}$$

where $Y_{1\cup 2,3}$ is a random variable representing the number of genes shared between two windows of size $r_1 + r_2 - x_{123} - x_{12}$ and r_3, under the null hypothesis. This approach still allows the use of pairwise statistical tests, but is more powerful than the above approach, since the second step considers the genes that occur in W_2 as well as those that occur in W_1, when comparing to a third homologous region. Nevertheless, it still requires that at least one pair of regions be independently significant.

A third approach also merges two of the three regions (W_1 and W_2), but does not require that the regions are significantly similar [123]. Rather, the only requirement is that the merged region be significantly similar to the third region W_3:

$$P(Y_{1\cup 2,3} \geq (x_{123} + x_{13} + x_{23})) \leq \alpha. \tag{3.3}$$

When constructing the merged region $W_1 \cup W_2$, neither of these two methods (Equation 3.2 and Equation 3.3) distinguish between genes that appear in only W_1 *or* W_2, and genes that appear in both W_1 *and* W_2. Thus, all three approaches fail to explicitly recognize the additional significance of genes that occur in all three regions (x_{123}). Also, the first and the third methods do not consider evidence from all three pairwise overlaps. No existing test considers both the three-way and pairwise overlaps simultaneously.

3.2 Overview

In this chapter we develop statistical tests for three windows, sampled independently from distinct chromosomal regions. This sampling approach is used when a researcher is interested in the region surrounding a particular gene, then compares the regions containing this gene in three different genomes for evidence of common ancestry. As long as the gene of interest is discarded from the statistical computation, our proposed tests are applicable to clusters found by this sampling approach. It is important to note, however, that these tests are not applicable if the windows were selected by a whole genome scanning approach in which all sets of three windows with genes in common are identified. In this case, the probability of observing the cluster by chance will be greater, since the search space is larger. Using the tests proposed here to evaluate the significance of clusters found by whole genome comparison will lead to false positives.

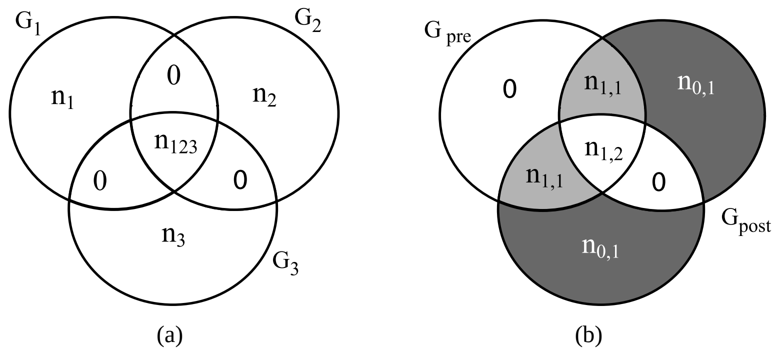


Figure 3.4: Gene content overlap models. The set of genes in each genome is represented as a circle. (a) Orthology model: n_{123} genes are shared between all three genomes. The remaining genes are singletons, *i.e.* they appear in only one genome . (b) Duplication model: G_{pre} is the union of two ancestral, duplicated genomes embedded within it. The $n_{1,2}$ genes that are retained in duplicate appear twice in G_{post} (once in each embedded genome) and once in G_{pre}. The light gray regions correspond to the $n_{1,1}$ genes that appear once in G_{pre} and once in G_{post}. These genes were preferentially lost. The dark gray regions correspond to the $n_{0,1}$ genes that appear once in G_{post}, but do not appear in G_{pre}. These genes are retained in singleton in G_{post} but lost in G_{pre}.

The significance of a cluster depends not only on the search strategy used to identify the cluster, and the properties of the windows (Figures 3.2(b) and 3.3(b)), but also on the properties of the genomes (Figure 3.4). The relevant properties of the genomes are the total number of genes in each genome and the *gene content overlap*— the fraction of genes shared among the three genomes. Depending on which biological questions are being investigated, an appropriate model of gene content overlap will also differ. Here, we develop statistical tests for two different models of gene content overlap. The first, the *Orthology Model*, is designed for comparisons of three regions selected from three distinct genomes. The second, the *Duplication Model*, is for comparison of a pair of regions duplicated by WGD with a reference region selected from a pre-duplication genome. Note that we use Venn diagrams to represent gene content overlap (Figure 3.4), but these differ from the Venn diagrams of gene clusters (Figure 3.3). In the former case, each circle represents the complete set of genes in the genome, whereas in the latter case each circle represents only the set of genes sampled from a specific region of the genome.

For each genome content overlap model we give analytical expressions for three-way statistical tests, and compute cluster probabilities for representative parameter values using Mathematica. We investigate the impact of different gene content overlap models and alternative test statistics on cluster significance, and compare the sensitivity of our tests with that of existing approaches.

3.3 Exact Probabilities for the Orthology Model

We model a genome G_i as an ordered set of N_i genes, $G_i\!=\!1,2,\ldots N_i$. We ignore chromosome breaks and physical distance between genes, and assume that genes do not overlap. We first consider a simpler version of this model, where each genome contains n identical genes, *i.e.* $N_1\!=\!N_2\!=\!N_3 = n$. Here, each gene in genome G_i is assumed to have exactly one homolog each in G_j and G_k.

3.3.1 Genomes with Identical Gene Content

We compute the probability of observing a cluster under the null hypothesis using a combinatorial approach. We first illustrate this approach for the simpler case of a pairwise cluster, then present analytical expressions for the probabilities of three-region clusters under the null hypothesis. Recall that the goal is to determine the probability under the null hypothesis that the test statistic would have at least the observed value. The probability $P(Y_{12} \geq y_{12})$ can be computed by counting the number of ways the two windows can be filled with genes, such that they share at least y_{12} genes, and normalizing by the number of ways of filling the windows without restrictions.

Given two windows, W_1 and W_2 of size r_1 and r_2, sampled from two genomes containing n identical genes, the number of ways the windows can share *exactly* y_{12} genes is $\binom{n}{y_{12}}\binom{n-y_{12}}{r_1-y_{12}}\binom{n-r_1}{r_2-y_{12}}$ [50]. The first binomial is the number of ways of choosing the y_{12} shared genes, and the remaining two binomials give the number of ways of choosing two sets of genes to fill the remainder of each window, such that the sets are disjoint. We normalize by the total number of ways of choosing genes to fill two windows of size r_1 and r_2. Thus, the probability that these windows share *exactly* y_{12} genes is

$$P_2(Y_{12}{=}y_{12}) = \frac{\binom{n}{y_{12}}\binom{n-y_{12}}{r_1-y_{12}}\binom{n-r_1}{r_2-y_{12}}}{\binom{n}{r_1}\binom{n}{r_2}} = \frac{\binom{n}{y_{12}, r_1-y_{12}, r_2-y_{12}}}{\binom{n}{r_1}\binom{n}{r_2}}, \tag{3.4}$$

where we define[1]

$$\binom{n}{i_1, i_2, ..., i_k} \equiv \binom{n}{i_1}\prod_{j=1}^{k-1}\binom{n-\sum_{l=1}^{j} i_l}{i_{j+1}} = \frac{n!}{i_1! i_2! \ldots (n-i_1-i_2 \ldots - i_k)!}.$$

From this, we can obtain the probability that two windows share *at least* y_{12} genes,

$$P_2(Y_{12} \geq y_{12}) = \sum_{h=y_{12}}^{\min(r_1,r_2)} P_2(Y_{12}{=}h). \tag{3.5}$$

We use an analogous approach and notation for computing the probabilities for comparisons of three regions. In a comparison of three windows, the random variable X_{12} represents the number of homologs shared between two windows of size r_1 and r_2, that *do not* appear in a third window of size r_3. The random variables X_{13} and X_{23} are defined analogously. The random variable X_{123} represents the number of genes shared between three windows of size r_1, r_2, and r_3, under the null hypothesis. For notational convenience, we define $\vec{x}{=}(x_{123}, x_{12}, x_{13}, x_{23})$ and use $\vec{X}{=}\vec{x}$ as shorthand for $X_{123}{=}x_{123}$, $X_{12}{=}x_{12}$, $X_{13}{=}x_{13}$, and $X_{23}{=}x_{23}$. Similarly, we use $\vec{Y}_{ij}{=}\vec{y}_{ij}$ as shorthand for $Y_{12}{=}y_{12}$, $Y_{13}{=}y_{13}$, and $Y_{23}{=}y_{23}$.

To compute $P(\vec{X} \geq \vec{x})$, the probability of observing at least $\vec{x}$ genes shared among three regions, we first derive an expression for the probability of observing exactly $\vec{x}$ genes, then sum over this expression. In the above pairwise comparison, we counted the number of ways to form three different sets: the y_{12} shared genes, the $r_1 - y_{12}$ genes unique to W_1, and the $r_2 - y_{12}$ genes unique to W_2. Computing the probability

[1] Note that this is a non-standard use of the multinomial notation since we do not require that $n{=}i_1 + i_2 + \ldots i_k$.

of three windows containing *exactly* the observed number of shared genes is a direct extension of the two-window problem, except there are seven sets to be selected (Figure 3.3(b)):

$$P(\vec{X}=\vec{x}) = \frac{1}{\binom{n}{r_1}\binom{n}{r_2}\binom{n}{r_3}} \cdot \binom{n}{x_{123},\ x_{12},\ x_{13},\ x_{23},\ x_1,\ x_2,\ x_3}. \tag{3.6}$$

The probability of observing *at least* $\vec{x}$ shared genes is obtained by summing over all possible values of X_{123} and X_{ij},

$$P(\vec{X} \geq \vec{x}) = \sum_{v_{123}=x_{123}}^{u_{123}} \sum_{v_{12}=x_{12}}^{u_{12}} \sum_{v_{13}=x_{13}}^{u_{13}} \sum_{v_{23}=x_{23}}^{u_{23}} P(\vec{X} = \vec{v}), \tag{3.7}$$

where $u_{123} = \min(r_1, r_2, r_3)$, $u_{12} = \min(r_1, r_2) - v_{123}$, $u_{13} = \min(r_1 - v_{12}, r_3) - v_{123}$, $u_{23} = \min(r_2 - v_{12}, r_3 - v_{13}) - v_{123}$, and $\vec{v} = (v_{123}, v_{12}, v_{13}, v_{23})$. In the worst case, evaluating this expression takes $O(r^4)$ time. In practice, the computation time can be substantially reduced, because the summand decreases exponentially as x_{123} and the $x_{ij}'s$ increase. Only the smallest values will contribute to the final probability, and most of the terms can be disregarded.

3.3.2 Genomes with Non-Identical Gene Content

In contrast to the assumptions of the identical gene content model, in most cases, a genome will have *singleton* genes that do not have a detectable homolog in related genomes. The greater the number of singletons, the fewer genes available to populate the windows such that the genes are shared between the windows. Here, we develop a statistical test for three-window clusters for the general orthology model in which the gene content of each genome may differ.

In this model, we assume the genomes share a common set of $n_{123} \leq \min(N_1, N_2, N_3)$ homologs (Figure 3.4(a)). In addition, each genome G_i contains $n_i = N_i - n_{123}$ singleton genes. Homology between gene pairs that have no homolog in the third genome is disregarded, with such genes being treated as singletons. This models the situation that would result if homologs were identified according to the triangle method used in COGs [166].

To compute the probability of observing exactly $\vec{x}$ shared genes, we must count the number of ways of choosing the $\vec{x}$ shared genes, as well as the genes that are unique to each window. As in the case of identical gene content, the shared genes must be selected from the n_{123} genes common to the three genomes. However, the x_i genes that are unique to each window W_i can be selected either from the remaining common genes, or from the n_i singletons in genome G_i. In the former case, care must be taken to ensure that a gene is only assigned to one window. As a result, two additional summations are required, since the number of ways to choose the x_3 genes unique to W_3 depends on how many genes from the n_{123} common genes were used to fill W_1 and W_2. The probability is:

$$\begin{aligned} P_S(\vec{X}=\vec{x}) = \binom{N_1}{r_1}^{-1} \binom{N_2}{r_2}^{-1} \binom{N_3}{r_3}^{-1} \binom{n_{123}}{x_{123},\ x_{12},\ x_{13},\ x_{23}} \\ \sum_{i=0}^{x_1} \sum_{j=0}^{x_2} \binom{n_{123}-s}{i, j} \binom{n_1}{x_1 - i} \binom{n_2}{x_2 - j} \binom{N_3 - s - i - j}{x_3}, \end{aligned} \tag{3.8}$$

where $s = x_{123} + x_{12} + x_{13} + x_{23}$ is the total number of shared genes.

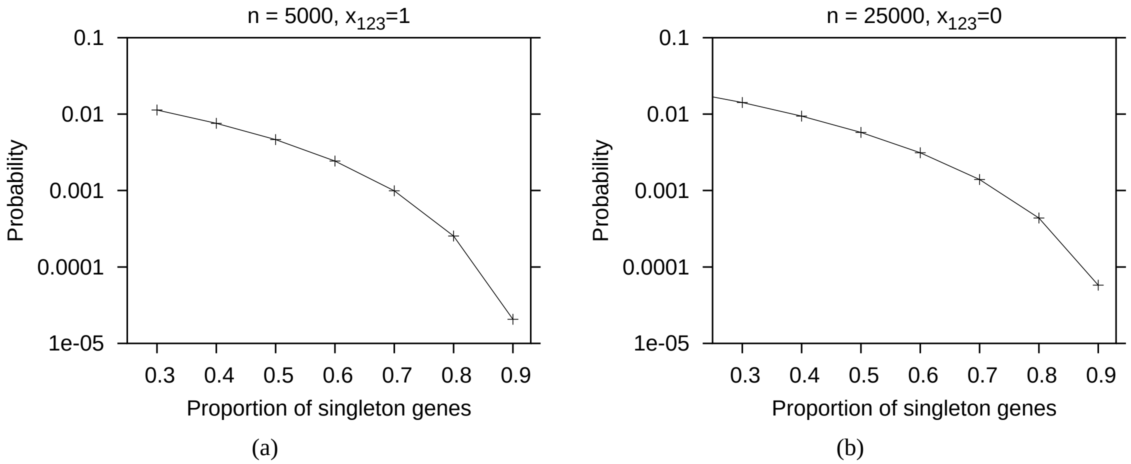


Figure 3.5: Cluster significance as a function of σ/n, the fraction of singleton genes in each genome. (a) The probability $P_S(\vec{X} \geq (1,1,1,1))$, when $n = N_1 = N_2 = N_3 = 5000$, and $r = 100$. (b) The probability $P_S(\vec{X} \geq (0,1,1,1))$, when $n = N_1 = N_2 = N_3 = 25000$, and $r = 100$.

The probability of observing *at least* as many shared genes under this model, can be computed from Equation 3.8 by summing $P_S(\vec{X} = \vec{x})$ over all possible values of X_{123} and X_{ij}:

$$P_S(\vec{X} \geq \vec{x}) = \sum_{v_{123}=x_{123}}^{u_{123}} \sum_{v_{12}=x_{12}}^{u_{12}} \sum_{v_{13}=x_{13}}^{u_{13}} \sum_{v_{23}=x_{23}}^{u_{23}} P_S(\vec{X} = \vec{v}), \tag{3.9}$$

where $u_{123} = \min(r_1, r_2, r_3)$, $u_{12} = \min(r_1, r_2) - v_{123}$, $u_{13} = \min(r_1 - v_{12}, r_3) - v_{123}$, $u_{23} = \min(r_2 - v_{12}, r_3 - v_{13}) - v_{123}$, and $\vec{v} = (v_{123}, v_{12}, v_{13}, v_{23})$.

3.3.3 Properties that Influence Cluster Significance

We use Equation 3.7 and Equation 3.9 to investigate how properties of the genomes, the cluster, and the test itself affect significance. First, we analyze how the proportion of singleton genes affects cluster significance. Next, we investigate how the distribution of the total number of shared genes among the three-way and pairwise overlaps affects significance. Finally, we compare the value of $P(\vec{X} \geq \vec{x})$ for clusters with similar numbers of shared genes, but where the shared genes are distributed differently in the Venn diagram.

How does the proportion of singletons affect cluster significance?

To study how cluster significance depends on the extent of gene content overlap among the genomes, we computed $P_S(\vec{X} \geq \vec{x})$, as a function of σ, the proportion of genes that are singletons. Note that given n and σ, n_{123} is defined by $n(1-\sigma)$. As σ increases, the probability of observing a cluster drops precipitously (Figure 3.5) for both $n = 5000$ and $n = 25000$. Figure 3.5(a) shows when $n = 5000$ and $r = 100$ the probability of a cluster with $x_{123} = 1$ and $x_{12} = x_{23} = x_{13} = 1$ drops from 0.01 to 10^{-5} as the proportion of singleton genes in the genomes increases from 0.3 to 0.9. Similarly, when $n = 25000$ and $r = 100$ the probability of a cluster with $x_{123} = 0$ and $x_{12} = x_{23} = x_{13} = 1$ drops sharply as shown in Figure 3.5(b).

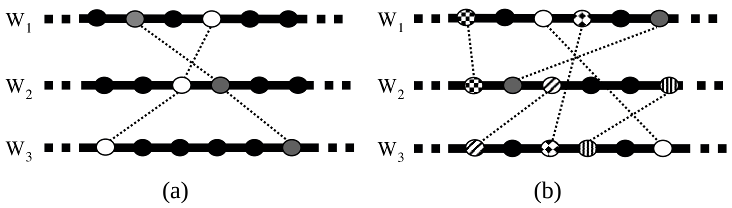


Figure 3.6: Two gene clusters with the same number of genes (h) conserved between each pair of regions. (a) A gene cluster in which two genes are shared by all three regions ($x_{123} = 2, x_{12} = x_{13} = x_{23} = 0$) (b) A gene cluster in which two distinct genes are shared by each pair of regions ($x_{123} = 0, x_{12} = x_{13} = x_{23} = 2$).

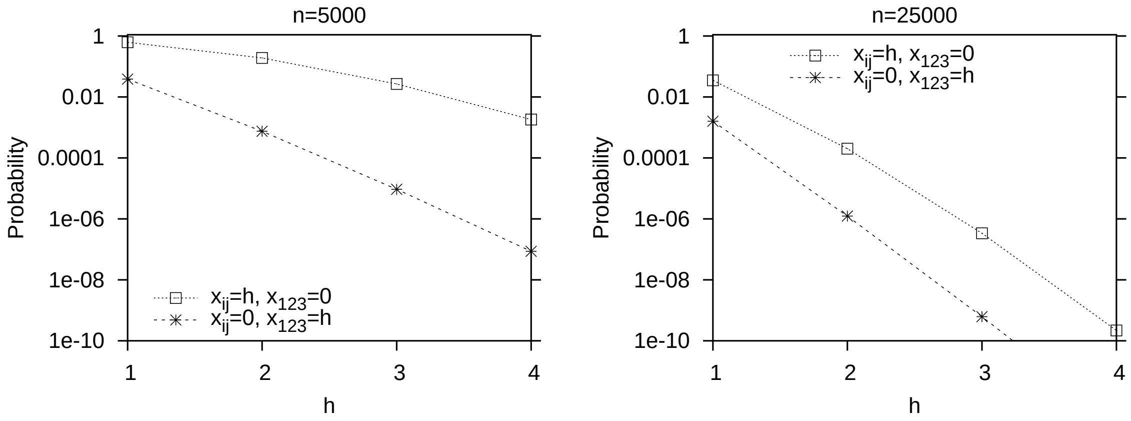
This is because as fewer homologs are shared between the genomes, it becomes much more surprising to find them clustered together. These examples underscore the importance of considering the extent of gene content overlap among the genomes when evaluating cluster significance.

How much more does a gene shared by all three windows contribute to significance?

To answer this question, we compare the significance of clusters in which h genes are shared by *all three* windows (as shown in Figure 3.6(a)), with clusters in which there are h distinct genes shared between each *pair* of windows (as shown in Figure 3.6(b)). Notice that in both examples shown in Figure 3.6 each pair of windows shares $h = 2$ genes. However, in the first case each region only contains $h = 2$ shared genes, whereas in the second case each region shares $2h = 4$ genes with the other regions. Although the total number of shared genes is larger in the second scenario, Figure 3.7(a) shows that the first scenario is much more significant. Even a small increase in x_{123} results in a large increase in significance—much more so than an increase of an equivalent number of homologous matches between pairs of regions. For larger values of n (Figure 3.7(b)), although the difference between the two scenarios is not as great, the second scenario is still more significant than the first.

How does the distribution of shared homologs among the pairwise overlaps influence significance?

We consider how an unequal distribution of the pairwise conserved genes (the x_{ij}'s) affects significance. We compare all possible distributions, ranging from a scenario in which only a single pair of windows shares genes, to a scenario in which the genes are distributed evenly among the three windows ($x_{12} = x_{13} = x_{23}$). Let $t = \sum x_{ij}$ be the total number of genes that appear in exactly two of the three regions. At one extreme, the t genes can be uniformly distributed: $x_{12} = x_{13} = x_{23} = t/3$. In this case, the variance of the x_{ij}'s will be zero. The distribution could be skewed, on the other hand, with the most extreme skew occurring when all t genes appear are shared between exactly one pair of regions: *e.g.* $x_{12} = t$ and $x_{13} = x_{23} = 0$. In this case the variance will be $t^2/3$. Figure 3.8 compares cluster probabilities for all possible distributions of the x_{ij}'s, as a function of the variance of the x_{ij}'s. It shows that the greater the variance of x_{ij}'s, the lower the probability of observing the cluster by chance. In other words, a skewed distribution of the x_{ij}'s is more significant than a uniform distribution. This illustrates why it is preferable to consider the value of each of the three pairwise overlaps independently, rather than considering only their sum.



(a) (b)

Figure 3.7: A comparison of $P(\vec{X} \geq (h,0,0,0))$ and $P(\vec{X} \geq (0,h,h,h))$, showing the impact of x_{123} and $x_{ij}'s$ on cluster significance, (a) when $n\!=\!5000$, $r\!=\!100$, (b) when $n\!=\!25000$, $r\!=\!100$.

3.3.4 Comparisons with Alternative Tests

In this section, to understand which aspects of our test are most important to cluster significance, we derive three alternative tests, and compare them with $P(\vec{X} \geq \vec{x})$. We consider the following alternative tests: $P(X_{123} \geq y_{123})$, to determine when it is necessary to consider the x_{ij}'s; $P(\vec{Y}_{ij} \geq \vec{y}_{ij})$, to determine how much information is lost by not explicitly considering the value of x_{123}; and $P(\vec{X} = \vec{x})$, to see whether it is sufficient to consider only the probability of observing an identical cluster, or whether more extreme ensembles must be considered as well. Finally, we compare our three-way test with two of the pairwise tests reviewed in Section 3.1.

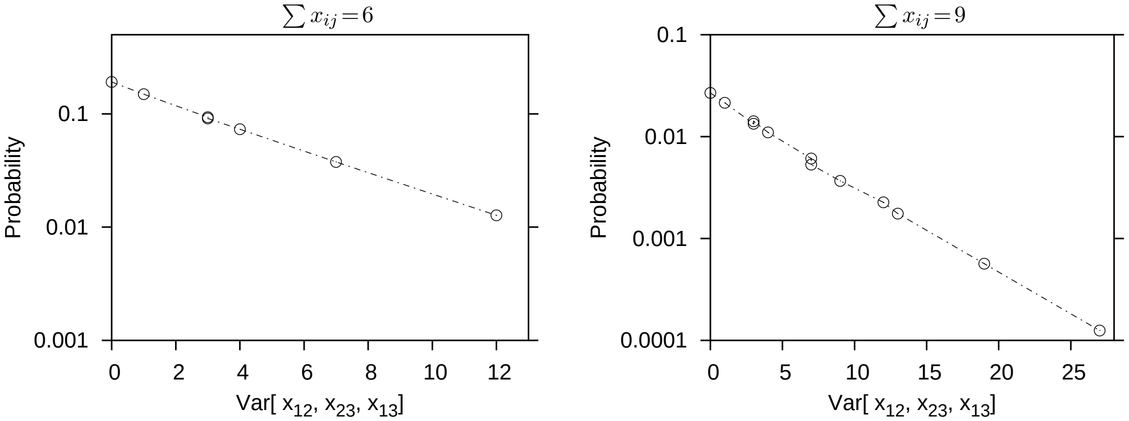
Is a test based only on $\mathbf{x_{123}}$ sufficient, or is it necessary to consider pairwise overlaps as well?

In order to assess the additional sensitivity gained by also considering genes shared between only two of three regions, we compare $P(\vec{X} \geq \vec{x})$ with $P(X_{123} \geq x_{123})$, the probability of observing at least x_{123} homologs shared between all three windows. To enumerate all triples of windows that share *exactly* x_{123} genes with no restrictions on the $x_{ij}'s$, it is necessary to select x_{12}, x_{13} and x_{23} so that they have no homologs in common. Otherwise, X_{123} would be greater than rather than equal to x_{123}. This can be achieved using the following expression for the number of windows that share *exactly* x_{123} genes:

$$q(X_{123} = x_{123}) \; = \sum_{x_{12}=0}^{r_1 - x_{123}} \binom{r_1}{x_{123}, x_{12}} \binom{n - r_1}{r_2 - x_{123} - x_{12}} \binom{n - x_{123} - x_{12}}{r_3 - x_{123}}, \tag{3.10}$$

where the second term ensures that W_1 and W_2 share exactly x_{12} genes, and the third term ensures that exactly x_{123} genes are shared in all three windows. We then obtain the probability of observing *at least* x_{123} genes in common by summing over $q(X_{123} = x_{123})$ as follows:

$$P(X_{123} \geq x_{123}) = \binom{n}{r_2}^{-1} \binom{n}{r_3}^{-1} \sum_{k=x_{123}}^{u_{123}} q(X_{123}\!=\!k). \tag{3.11}$$



(a) (b)

Figure 3.8: The probability of observing a cluster when $n = 5000$, $r = 100$, $x_{123} = 0$, and $x_{12} + x_{13} + x_{23} = t$, as a function of the variance of the x'_{ij}s, where higher variance indicates more skew. (a) When $t = 6$, the variance of the x_{ij}'s ranges from 0 when the x_{ij}'s are uniformly distributed ($x_{12} = x_{13} = x_{23} = 2$) to 12 when the x_{ij}'s are maximally skewed ($x_{12} = 6, x_{13} = x_{23} = 0$). (b) When $t = 9$, the variance of the x_{ij}'s ranges from 0 ($x_{12} = x_{13} = x_{23} = 3$) to 27 ($x_{12} = 9, x_{13} = x_{23} = 0$).

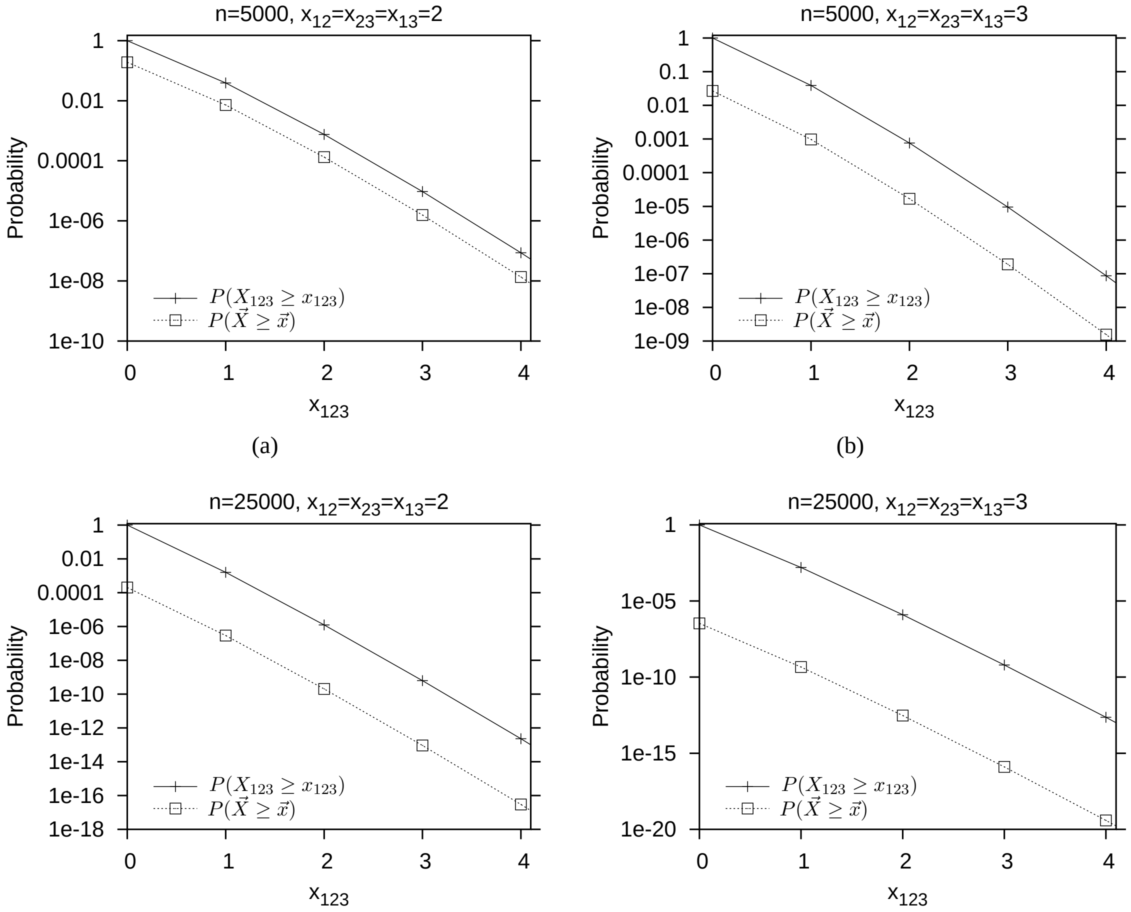
We analyzed the impact of disregarding the $x_{ij}'s$, by comparing Equation 3.11 with Equation 3.7 when $n \in \{5000, 25000\}$ and $x_{12} = x_{13} = x_{23} \in \{2, 3\}$, for a range of values of x_{123} (Figure 3.9). $P(X_{123} \geq x_{123})$ is consistently two orders of magnitude greater than $P(\vec{X} \geq \vec{x})$. This is because a test based only on x_{123} fails to capture evidence of homology from genes that occur in only a subset of the windows (*i.e.* the $x_{ij}'s$), and will severely overestimate the probability of observing a cluster by chance. For example, given a significance threshold of $\alpha = .01$ and the parameters used in Figure 3.9(b), a cluster with $x_{12} = x_{13} = x_{23} = 3$ and $x_{123} = 1$ would not be considered significant using a test based on x_{123} alone, even though the three-way test shows that such a cluster is unlikely to arise by chance. Clearly, a test that considers only x_{123} is overly conservative, and will lead to many false negatives.

Is it necessary to consider explicitly the number of genes that appear in all three windows?

Our test statistic $\vec{X}$ distinguishes between x_{123} and each of the three pairwise overlaps. A simpler alternative would be to consider both x_{123} and the x_{ij}'s, but to not distinguish between the two. To investigate whether it is necessary to consider x_{123} explicitly, we compare $P(\vec{X} \geq \vec{x})$ with $P(\vec{Y}_{ij} \geq \vec{y}_{ij})$. Recall that $y_{ij} = x_{ijk} + x_{ij}$, *i.e.* it is defined as the *total* number of genes shared between windows W_i and W_j, including those genes that are also contained in W_k. Note that $\vec{Y}_{ij} \geq \vec{y}_{ij}$ is strictly a weaker constraint than $\vec{X} \geq \vec{x}$. In addition to all the ensembles in which $\vec{X} \geq \vec{x}$, two additional sets of ensembles will be counted when computing $P(\vec{Y}_{ij} \geq \vec{y}_{ij})$ that would not be counted when computing $P(\vec{X} \geq \vec{x})$:

1. $X_{123} \geq \max(y_{12}, y_{23}, y_{13})$, and $X_{12} + X_{123} < y_{12}$ or $X_{13} + X_{123} < y_{13}$ or $X_{23} + X_{123} < y_{23}$.
2. $X_{123} < \max(y_{12}, y_{23}, y_{13})$, and $X_{12} + X_{123} \geq y_{12}$ and $X_{13} + X_{123} \geq y_{13}$ and $X_{23} + X_{123} \geq y_{23}$.

For example, if we observe a cluster with $x_{123} = 2$, $x_{12} = x_{13} = 1$, and $x_{23} = 0$, then to compute $P(\vec{X} \geq \vec{X})$ we count the number of ensembles in which $x_{123} \geq 2$, $x_{12} \geq 1$, and $x_{13} \geq 1$. To compute



(a)

(b)

(c)

(d)

Figure 3.9: A comparison of $P(X_{123} \geq x_{123})$ with $P(\vec{X} \geq \vec{x})$ as a function of x_{123}, when (a) $n = 5000$, $r = 100$, and $x_{12} = x_{13} = x_{23} = 2$. (b) $n = 5000$, $r = 100$, and $x_{12} = x_{13} = x_{23} = 3$. (c) $n = 25000$, $r = 100$, and $x_{12} = x_{13} = x_{23} = 2$. (d) $n = 25000$, $r = 100$, and $x_{12} = x_{13} = x_{23} = 3$.

$P(\vec{Y}_{ij} \geq \vec{y}_{ij})$, we will also enumerate the number of ensembles in which $x_{123} = 0$, $x_{12} \geq 3$, and $x_{13} \geq 3$, and the number of ensembles in which $x_{123} = 1$, $x_{12} \geq 2$, and $x_{13} \geq 2$.

Thus $P(\vec{Y}_{ij} \geq \vec{y}_{ij})$ will always be an upper bound on $P(\vec{X} \geq \vec{x})$. In particular, with Equation 3.12, the significance of a cluster in which h genes are shared by all three windows (as shown in Figure 3.6(a)) will be the same as that of a cluster in which h distinct genes are shared between each *pair* of windows (as shown in Figure 3.6(b)).

To compute $P(\vec{Y}_{ij} \geq \vec{y}_{ij})$ we simply sum $P(\vec{X} \geq \vec{x})$ over all possible values of X_{123}:

$$\begin{aligned} P(\vec{Y}_{ij} \geq \vec{y}_{ij}) &= \sum_{v_{123}=0}^{u_{123}} \sum_{v_{12}=\delta(y_{12})}^{u_{12}} \sum_{v_{13}=\delta(y_{13})}^{u_{13}} \sum_{v_{23}=\delta(y_{23})}^{u_{23}} P(\vec{V} = \vec{v}) \\ &= P(X_{123} \geq y_{\max}) + \sum_{v_{123}=0}^{y_{\max}-1} \sum_{v_{12}=\delta(y_{12})}^{u_{12}} \sum_{v_{13}=\delta(y_{13})}^{u_{13}} \sum_{v_{23}=\delta(y_{23})}^{u_{23}} P(\vec{V} = \vec{v}) \end{aligned} \tag{3.12}$$

where $\delta(x) = \max(0, x - v_{123})$ and $y_{\max} = \max(y_{12}, y_{13}, y_{23})$.

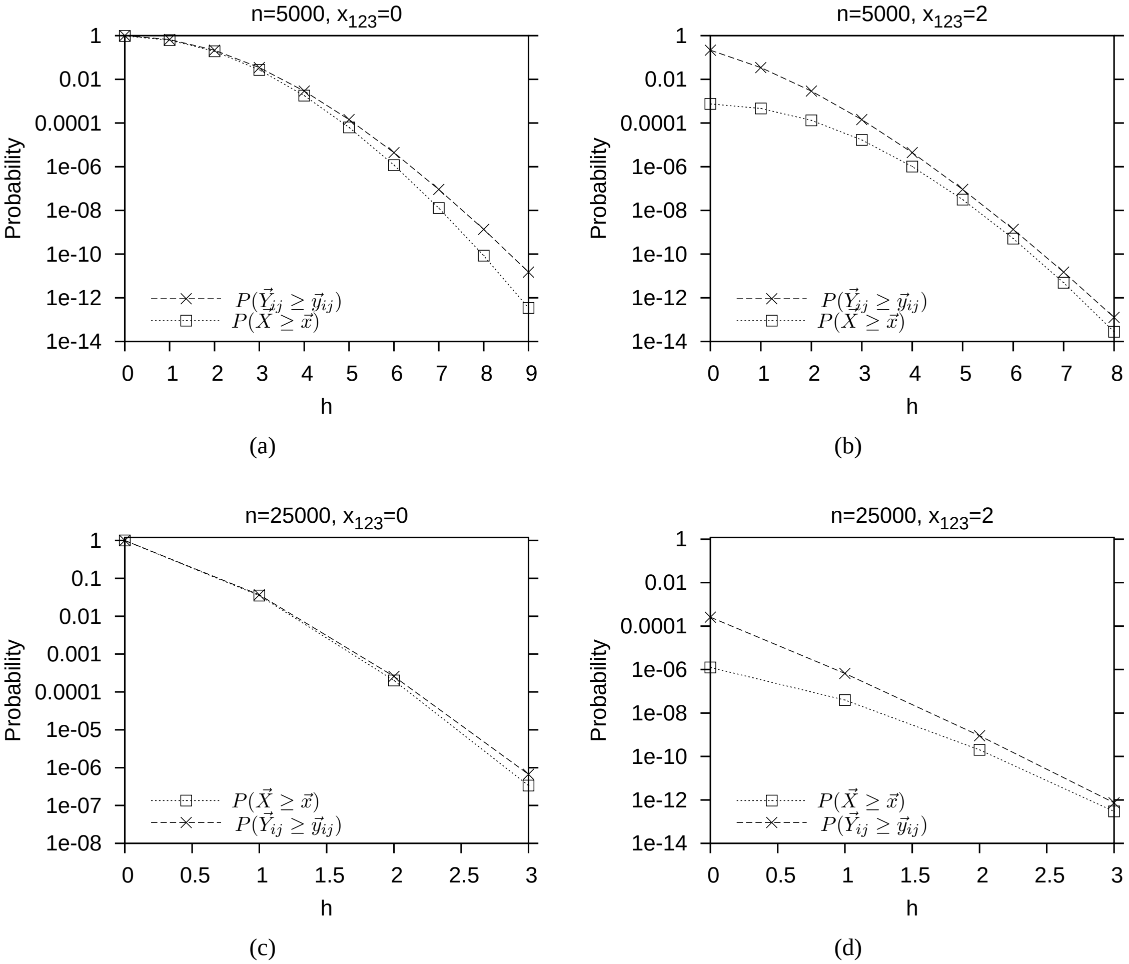
We compared $P(\vec{Y}_{ij} \geq \vec{y}_{ij})$ with $P(\vec{X} \geq \vec{x})$, when $x_{123} \in \{0, 2\}$, for a range of values of x_{ij}'s (Figure 3.10). When $x_{123} = 0$ and h is small, $P(\vec{Y}_{ij} \geq \vec{y}_{ij})$ is very close to $P(\vec{X} \geq \vec{x})$. When $x_{123} = 0$ and h is large, $P(\vec{Y}_{ij} \geq \vec{y}_{ij})$ diverges slightly from $P(\vec{X} \geq \vec{x})$, but in this region a cluster would be significant according to either test. In short, when $x_{123} = 0$, $P(\vec{Y}_{ij} \geq \vec{y}_{ij})$ is a accurate test. On the other hand, when $x_{123} = 2$ and $x_{12} = x_{23} = x_{13} = 0$, $P(\vec{Y}_{ij} \geq \vec{y}_{ij})$ overestimates $P(\vec{X} \geq \vec{x})$, as shown in Figure 3.10(b) and Figure 3.10(d). In this case, the approximation could lead to false negatives, since $P(\vec{Y}_{ij} \geq \vec{y}_{ij})$ does not recognize the greater significance of genes that appear in all three regions.

Is $P(\vec{X} = \vec{x})$ a suitable measure of significance?

It might seem natural to use the probability of observing the *exact* number of shared homologs directly to test cluster significance. To investigate this, we compared $P(\vec{X} = \vec{x})$ with $P(\vec{X} \geq \vec{x})$ when $n = \{5000, 25000\}$, $x_{123} = \{0, 1\}$, and $x_{12} = x_{13} = x_{23} = h$, for a range of values of h (see Figure 3.11). Using $P(\vec{X} = \vec{x})$ is risky: When $n = 5000$ and for small values of x_{ij}, $P(\vec{X} = \vec{x})$ underestimates $P(\vec{X} \geq \vec{x})$ by several orders of magnitude. For example, given the parameters in Figure 3.11(a), even when the three regions share *no* genes ($x_{123} = x_{12} = x_{13} = x_{23} = 0$), the probability $P(\vec{X} = \vec{x})$ is significantly less than one! Therefore, this test will lead to false positives when x_{ij}'s are small. As h increases, the probabilities converge and $P(\vec{X} = \vec{x})$ is a good approximation for $P(\vec{X} \geq \vec{x})$. In contrast, when $n = 25,000$ (Figs. 3.11(c) and 3.11(d)), $P(\vec{X} = \vec{x})$ is a closer approximation to $P(\vec{X} \geq \vec{x})$ even for small values of x_{ij}. In general, $P(\vec{X} = \vec{x})$ is a lower bound on $P(\vec{X} \geq \vec{x})$, and can be computed more efficiently. $P(\vec{X} = \vec{x})$ is a useful first test because if we cannot reject the null hypothesis using $P(\vec{X} = \vec{x})$, then we will not be able to reject using $P(\vec{X} \geq \vec{x})$. However, when $P(\vec{X} = \vec{x})$ is small, then a second test will be required.

How does our three-way test compare to existing pairwise tests?

To assess the difference between existing pairwise tests reviewed in Sec. 3.1 and our joint three-region statistical tests, we compare our Equation 3.7 ($P(\vec{X} \geq \vec{x})$) with Equation 3.1 and Equation 3.3, for a range of representative parameter values. (We did not plot Equation 3.2 as it will always lie between the curves



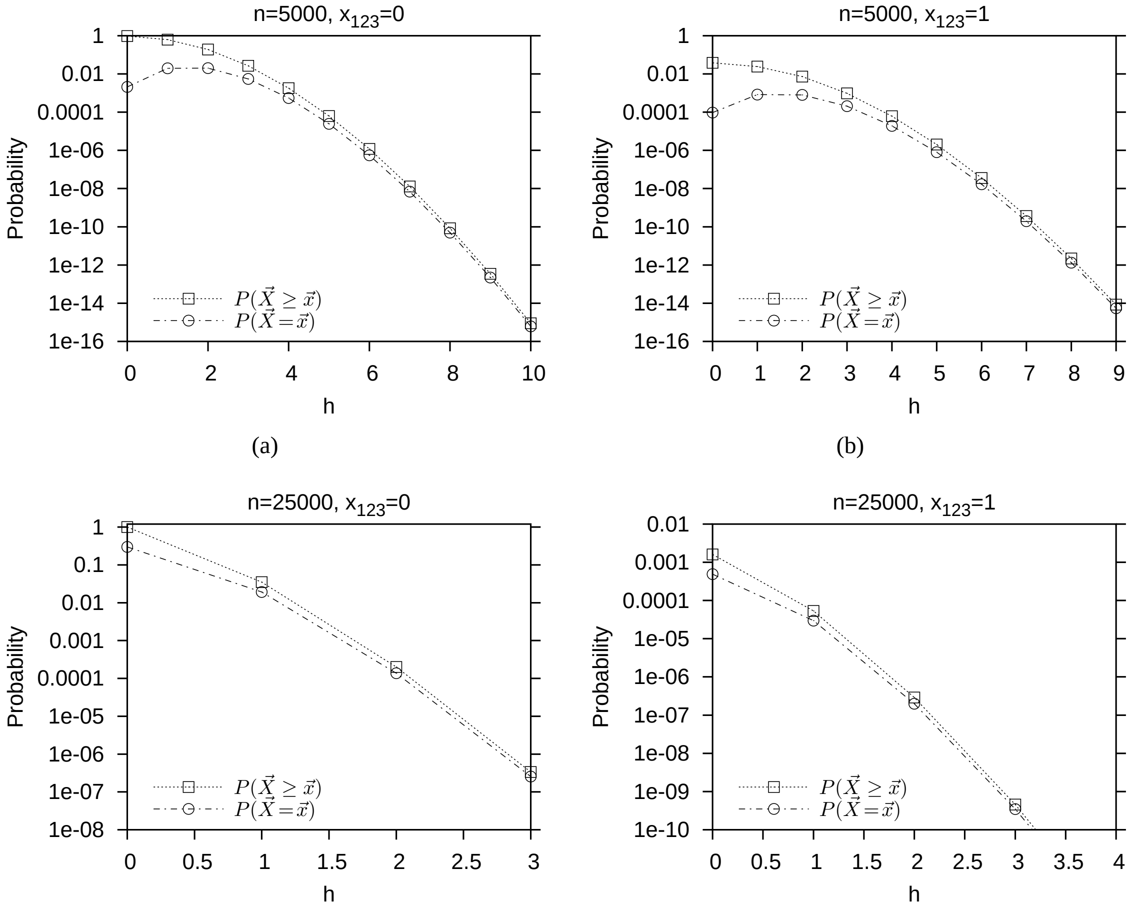
(a)

(b)

(c)

(d)

Figure 3.10: A comparison of $P(\vec{X} \geq \vec{x})$ with $P(\vec{Y}_{ij} \geq \vec{y}_{ij})$, as a function of h, where $x_{12} = x_{13} = x_{23} = h$ and (a) $n = 5000$, $r = 100$, $x_{123} = 0$, (b) $n = 5000$, $r = 100$, $x_{123} = 2$, (c) $n = 25000$, $r = 100$, $x_{123} = 0$ (d) $n = 25000$, $r = 100$, $x_{123} = 2$.



(a)

(b)

(c)

(d)

Figure 3.11: A comparison of $P(\vec{X} \geq \vec{x})$ with $P(\vec{X} = \vec{x})$ as a function of h, where $x_{12} = x_{13} = x_{23} = h$ and (a) $n = 5000$, $r = 100$, $x_{123} = 0$, (b) $n = 5000$, $r = 100$, $x_{123} = 1$ (c) $n = 25000$, $r = 100$, $x_{123} = 0$ (d) $n = 25000$, $r = 100$, $x_{123} = 1$.

for Equation 3.1 and Equation 3.3.) In Figure 3.12 we plot the significance level at which a null model of random gene order would be rejected by each test, when $n = 5000$, $x_{123} = \{0, 2, 3, 5\}$, $x_{12} = x_{13} = x_{23} = h$, and h ranges from zero to twelve. We consider a uniform distribution of the x_{ij}'s, in order to focus on the effect of x_{123} on cluster significance. There are two regions of the parameter space of particular interest.

The first case of interest is when $x_{123} = 0$, but the pairwise overlaps are relatively large. In this case, we can see the importance of considering all pairwise overlaps in the absence of genes conserved in all three regions. When $x_{123} = 0$, both Equation 3.1 and Equation 3.3 overestimate the probability of a cluster (Figure 3.12(a)). Recall that Equation 3.1 conducts two independent pairwise tests of W_1 with W_2, and then W_2 with W_3, whereas Equation 3.3 compares the merged region $W_1 \cup W_2$ with W_3. Equation 3.1 is a conservative test because it requires two of the three pairwise tests to be independently significant, and ignores the overlap between the windows W_2 and W_3, whereas our approach considers the three regions jointly. Equation 3.3 is a better approximation, but is still overly conservative, because it does not consider the overlap between windows W_1 and W_2. As a result, both tests may miss significant clusters. For example, in Figure 3.12(a), given a significance threshold of $\alpha = 0.001$, for a *pair* of regions to be significantly similar (Equation 3.1), they must share at least eight genes. In other words, to find a three-way cluster with a sequential pairwise approach, W_1 must share eight genes each with W_2 and W_3. With the pairwise merging approach, W_1 and W_2 must together share at least six genes with W_3. In contrast, using our test $P(\vec{X} \geq \vec{x})$, a cluster is significant in both the above cases, but also in the case where each pair of regions shares only four genes, *even when none of these genes appear in all three regions*. This example demonstrates the importance of considering all pairwise overlaps in the absence of genes conserved in all three regions.

The second case of interest is when x_{123} is non-zero, and the pairwise overlaps are small. In this case, tests which consider only the pairwise overlaps may fail to reject the null hypothesis, even though it is highly unlikely that such a cluster would occur given random gene order. On the other hand, our test, which considers x_{123}, does not make this error. When x_{123} is non-zero (Figs. 3.12(b), 3.12(c) and 3.12(d)), and the pairwise overlaps are small, both Equation 3.1 and Equation 3.3 overestimate the probability of a cluster, and would result in false negatives. Given a significance threshold of $\alpha = 0.001$, when $x_{123} = 2$, both Equation 3.1 and Equation 3.3 would fail to reject the null hypothesis for clusters in which $h < 5$ (Figure 3.12(b)), and when $x_{123} = 3$, they would fail to reject the null hypothesis for clusters in which $h < 4$ (Figure 3.12(c)). Even when $x_{123} = 5$, and the cluster is undoubtedly significant, the pairwise approaches would still fail to reject the null hypothesis when $h < 3$.

In summary, our three-way test is more sensitive than existing tests based on pairwise comparison. Those tests are overly conservative, and as a result may fail to reject the null hypothesis even when a cluster is highly unlikely to occur by chance.

3.4 Exact Probabilities for the Duplication Model

Following a WGD, in many cases there is no immediate selective advantage for retaining a gene in duplicate, so one of the duplicates is often lost. Since duplicated regions may share few paralogous genes, they are often detected by comparison with a related pre-duplication genome. For example, in the species tree shown in Figure 3.14, WGD occurred after the divergence of *K waltii* and before the speciation event that produced *S. bayanus* and *S. cerevisiae*. Duplicated regions in *S. cerevisiae*, a post-duplication species, can be detected by comparison with *K. waltii*, a pre-duplication species. We propose a second genome overlap model specifically for analyzing such duplications. Let G_{post} be a genome that has undergone a WGD and

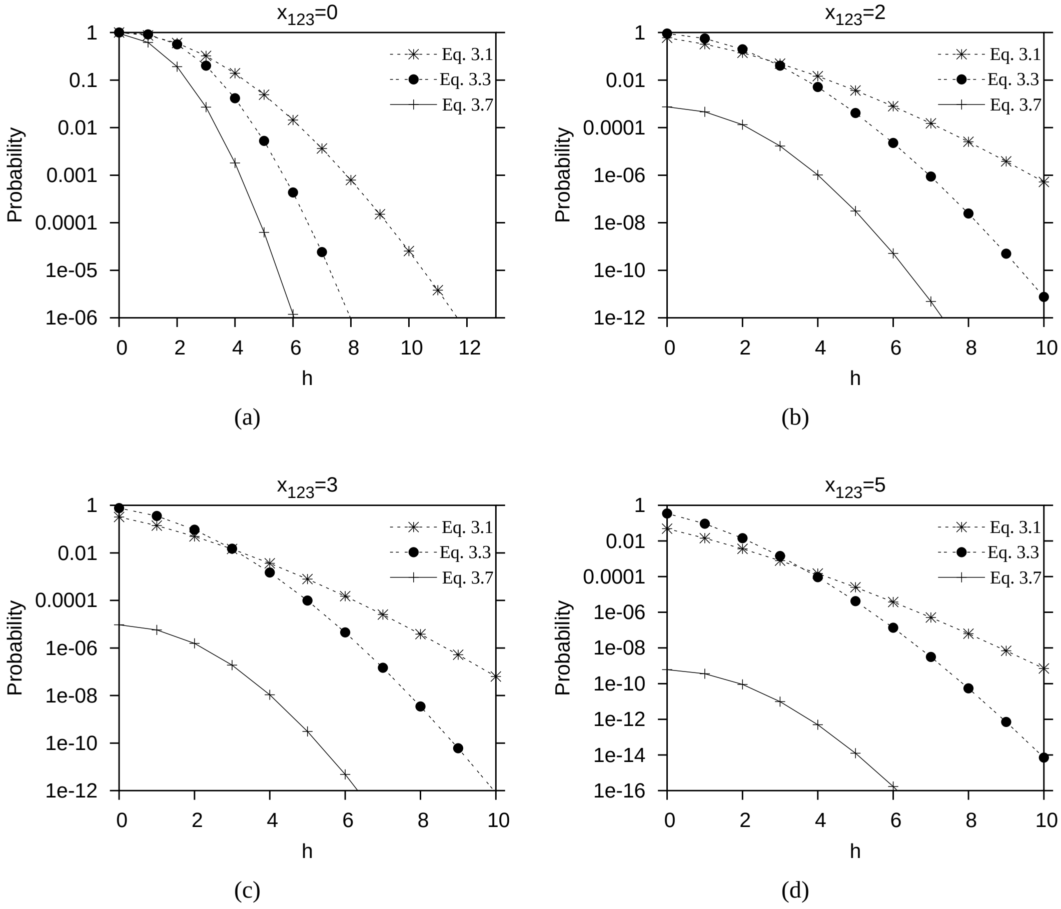


Figure 3.12: A comparison of our three-region test $P(\vec{X} \geq \vec{x})$ (Equation 3.7) with two existing tests based on pairwise comparisons (Equation 3.1 and Equation 3.3). The significance level at which a null model of random gene order would be rejected by each test, when $n = 5000$, $r = 100$, $x_{12} = x_{13} = x_{23} = h$, where h is the independent variable, and (a) $x_{123} = 0$, (b) $x_{123} = 2$, (c) $x_{123} = 3$, (d) $x_{123} = 5$.

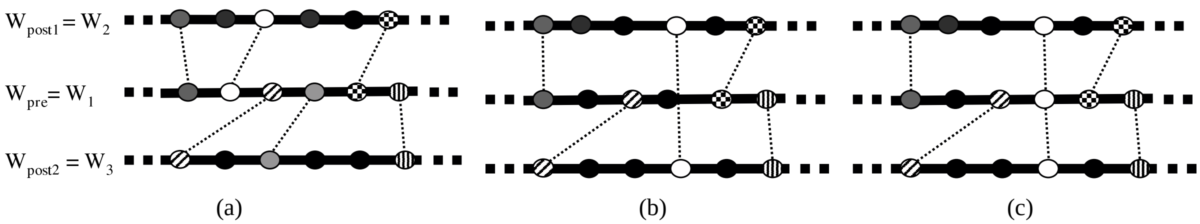


Figure 3.13: Pre-post gene cluster examples with different gene loss scenarios, in which two regions, W_{post1} and W_{post2}, from the genome of a post duplication species are compared with a region W_{pre} from a pre-duplication species. (a) A pre-post gene cluster where W_{pre} share three genes each with W_{post1} and W_{post2} ($x_{13} = x_{23} = 3$). W_{post1} and W_{post2} do not share any genes ($x_{12} = 0$, $x_{123} = 0$). (b) A cluster in which W_{post1} and W_{post2} share two genes with W_{pre} and have a single gene in common ($x_{12} = 1$, $x_{123} = 0$). (c) A cluster in which W_{post1} and W_{post2} share two genes with W_{pre} and there is an additional gene shared by all three regions ($x_{12} = 0$, $x_{123} = 1$).

G_{pre} be a genome that diverged prior to the WGD (Figure 3.4(b)). Let $n_{i,j}$ be the number of genes that appear i times in G_{pre} and j times in G_{post}, where $i \leq 1, j \leq 2$. This model only recognizes paralogs that arose through WGD, ignoring lineage specific duplications. Thus, it assumes that each gene in G_{post} has at most one paralog and that genes in G_{pre} have no paralogs; *i.e.* $n_{2,0} = n_{2,1} = n_{2,2} = 0$. Furthermore, this model assumes that every gene that appears twice in the post-duplication genome also has a homolog in the pre-duplication genome; *i.e.* $n_{0,2} = 0$. This assumption is based on the rationale that genes retained in duplicate are functionally important and, hence, are retained in G_{pre} as well. This assumption is supported by empirical observation. For example, in post-WGD yeast species over 95% of genes retained in duplicate are also present in each pre-WGD yeast genome [29]. Similarly, in this model every gene in G_{pre} has at least one homolog in G_{post} ($n_{1,0} = 0$). We use the convention that W_3 is the window sampled from G_{pre}, and W_1 and W_2 are sampled from distinct chromosomal regions in G_{post}.

To compute the probability of observing *exactly* $\vec{x}$ shared homologs under the null hypothesis, we make the additional assumption that at most one copy of a duplicated gene appears in a given window. Given this condition,

$$P_D(\vec{X} = \vec{x}) = \frac{\binom{n_{1,2}}{x_{123}, x_{12}} \binom{N_{pre} - x_{123} - x_{12}}{x_{13}, x_{23}} \binom{N_{pre} - s}{x_3} \binom{N_{post} - n_{1,2} - s - x_3}{x_1, x_2}}{\binom{N_{pre}}{r_3} \sum_{i=0}^{\min(r_1, r_2)} \binom{n_{1,2}}{i} \binom{N_{pre} + n_{0,1} - i}{r_1 - i} \binom{N_{pre} + n_{0,1} - r_1}{r_2 - i}},$$

where $N_{pre} = n_{1,2} + n_{1,1}$ and $N_{post} = 2n_{1,2} + n_{1,1} + n_{0,1}$. $P_D(\vec{X} \geq \vec{x})$, the probability of observing *at least* $\vec{x}$ shared homologs under the null hypothesis, is then obtained as before by summing over $P_D(\vec{X} = \vec{x})$

How do Retained Duplicates after WGD Affect Cluster Significance? To investigate the importance of the genes conserved in duplicates, we calculated $P_D(\vec{X} \geq \vec{x})$ with parameter values based on recent studies of pre- and post-duplication species in the yeast [29, 146] and bony fish [89] lineages. We compare the significance of clusters for three reciprocal gene loss scenarios: when no genes are shared by the post-duplication windows W_1 and W_2 ($x_{123} = 0, x_{12} = 0$, as shown in Figure 3.13(a)), when a single gene is shared by W_1 and W_2 but none are shared by all three regions ($x_{123} = 0, x_{12} = 1$, as shown in Figure 3.13(b)), and when a single gene is shared among all three regions ($x_{123} = 1, x_{12} = 0$, as shown in Figure 3.13(c)).

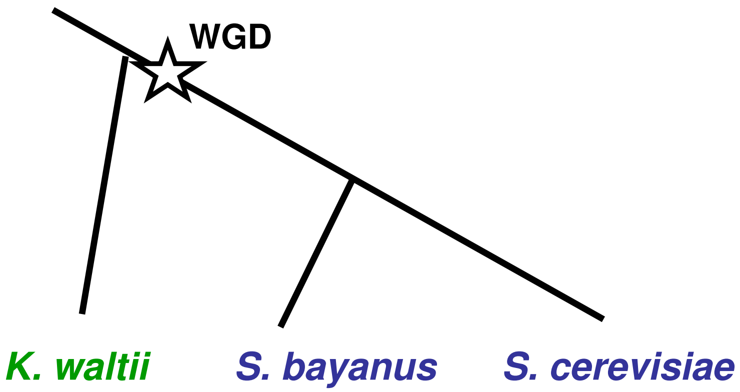
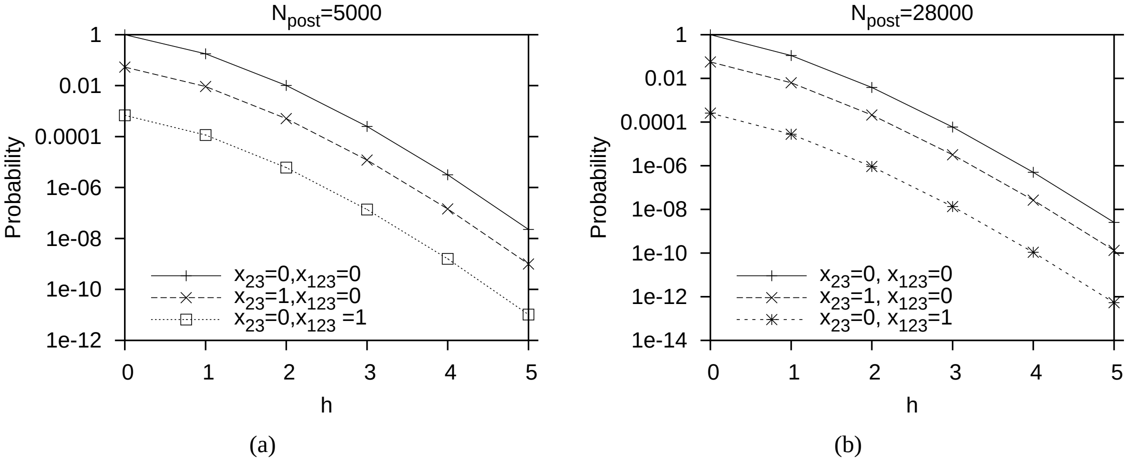


Figure 3.14: A species tree containing three yeast species. A whole genome duplication (indicated by a star) occurred after *K. waltii* diverged from the lineage leading to *S. bayanus* and *S. cerevisiae*.

In our simulations based on yeast, we used $N_{post} = 5000$ and $n_{1,2} = 450$. These parameters are consistent with the observation that only 16% of genes in *S. cerevisiae* are duplicate genes that arose during the WGD. Figure 3.15(a) shows the probabilities for these cluster scenarios when $x_{13} = x_{23} = h$, and h ranges from 0 to 5. The shape of the three curves is similar, but the probabilities drop by an order of magnitude from one to the next. Genes retained in duplicate have a large impact on cluster significance. For example, in Figure 3.15(a), given a significance threshold of $\alpha = 0.001$, if only overlaps between the pre- and post-duplication windows are considered, each pair of windows much share three genes in order to reject the null hypothesis. However, if there is a single gene retained in all three windows, then random gene order can be rejected regardless of how many other genes are shared by the pre- and post-duplication regions.

In our simulations based on bony fish, we selected parameter values from a recent study of WGD in the bony fish lineage, in which duplications in the *Tetraodon* genome were identified by comparison with the human genome [89]. In these simulations we used $n_{1,2} = 3500$, $n_{1,1} = 19500$, and $n_{0,1} = 1500$. Although the *Tetraodon* and human genomes are much larger than yeast genomes, the statistical analysis shows similar trends (Figure 3.15(b)): again, even the addition of a single gene retained in duplicate has a large effect on significance!

Retained duplicates have such a large impact on cluster significance because the number of genes that occur twice in G_{post} is small. This is equivalent to having a very small value of n_{123} in the Orthology model. In the Duplication model, the gene content overlap between the three conceptual genomes in the Venn diagram will always be quite small, and so even small values of x_{123} and x_{23} lead to highly significant clusters. This is particularly noteworthy because most current methods compare the pre-duplication region independently with each of the post-duplication regions, and thus ignore the values of x_{12} and x_{123} entirely [89, 93, 96, 146, 172, 173]. Our results show that existing methods could fail to detect clearly significant clusters, and that by using a multi-region test additional duplicated regions may be uncovered.



(a) (b)

Figure 3.15: The effect of reciprocal loss on cluster significance in comparing pre- and post-duplication genomes, when $r = 50$, $x_{12} = x_{13} = h$, as h ranges from 0 to 5, and (a) $n_{1,2} = 450, n_{1,1} = 3600, n_{0,1} = 500$ (b) $n_{1,2} = 3500, n_{1,1} = 19500, n_{0,1} = 1500$.

3.5 Discussion and Open Problems

In this paper, we presented a simple framework that allows us to understand and compare existing statistical tests for clusters spanning more than two regions. We proposed two different models of gene content overlap suitable for common comparative genomics problems. Based on these models, we developed novel statistical tests for evaluating the significance of gene clusters spanning three regions. Here, we have presented initial results for the design of tests for multi-region clusters, and shown that multi-region tests are able to validate distantly related homologous regions that will be dismissed by pairwise tests, or by a test based on x_{123} alone.

Our three-way tests are the first to combine evidence from genes shared among all three regions and genes shared only between pairs of regions. Unlike tests that consider only x_{123}, our three-way tests also consider $x_{ij}'s$, and thus can detect significant clusters even when x_{123} is small (Figure 3.9(a)). In addition, our tests outperform current approaches based on sequential pairwise tests, as shown in Sec. 3.3.4. These approaches disregard two important pieces of information. They do not always consider evidence from all three pairs of regions. Even more importantly, they do not explicitly consider the number of genes shared among all three regions. Our results show that even a few genes conserved in all three regions dramatically increases the statistical significance of gene clusters (Figure 3.7(a)). This effect is particularly strong when the shared gene content of the genomes is small (Figure 3.5(a)). Thus, unlike pairwise tests, our approach can detect related regions where each pair of regions share only a few genes (*i.e.* $x_{ij}'s$ are small), but where a few genes are also shared among all the regions (*i.e.* x_{123} is non-zero but small).

The difference between our tests and sequential pairwise tests is even more striking in the duplication model. We showed that even the addition of a single gene retained in duplicate has a large effect on significance (Figure 3.15(a)). However, current tests compare the pre-duplication region independently with each of the post-duplication regions, and thus ignore these retained duplicates. Consequently, there could be a large number of highly significant gene clusters for which sequential pairwise tests would fail to reject the null hypothesis of random gene order, but a three-way test would provide strong evidence that the regions

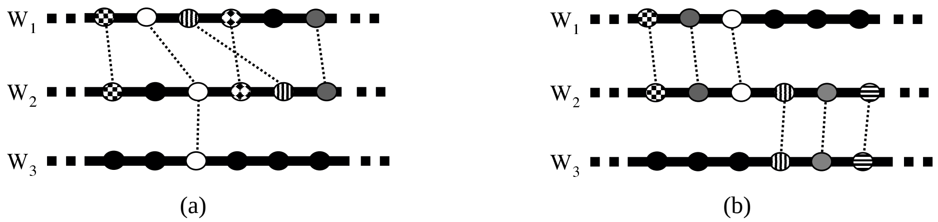


Figure 3.16: Examples of potentially misleading gene clusters. (a) Windows W_1 and W_2 share many genes, but W_3 shares only a single gene with each. Even if this cluster is highly unlikely to occur by chance, concluding that all three regions are homologous would be a mistake in this case. (b) The leftmost three genes in W_2 appear in the leftmost half of W_1, and the rightmost three genes in W_2 appear in the rightmost half of W_3. Even if this cluster is highly unlikely to occur by chance, it may be incorrect to conclude that all three regions arose from a single region.

arose through duplication.

It is important to be precise about the conclusions that can be drawn on the basis of these tests. A small p-value does not guarantee that all three regions descended from a single region in the genome of a common ancestor. Even if only two of the windows descended from a common region, it is quite likely that we will be able to reject the null hypothesis of random gene order. Figure 3.16(a) shows an example in which windows W_1 and W_2 share many genes, but W_3 shares only a single gene with each. Concluding that all three regions are homologous would be a mistake in this case. Furthermore, even if the cluster is significant, this does not mean that the regions arose from a common ancestor spanning the entirety of all three regions. It could be that only a small portion of each region is homologous, but the signal from this sub-region is still strong enough to reject the null hypothesis that the regions are completely unrelated. Figure 3.16(b) shows an example in which the leftmost three genes in W_2 also appear in the leftmost half of W_1, and the rightmost three genes in W_2 also appear in the rightmost half of W_3. Given this scenario, it may in fact be the case that the region of W_1 that is homologous to W_2 is distinct from the region of W_2 that is homologous with W_3. In this case it may be incorrect to conclude that all three regions arose from a single region. One possibility would be to flag such clusters, or screen them out entirely, in a post-processing step.

The work presented here can be extended in many ways. Our genome overlap models make certain assumptions that may not always hold. For example, in the orthology model we assume that there are no genes that appear in only two of the three genomes. In our duplication model, we assume there are no genes that appear in G_{pre} but not G_{post}. In our orthology models we disregard paralogs entirely, and in our duplication model, we consider only those paralogs that arose via WGD. Also, our test for duplicated regions assumes that there will never be two copies of a gene in a window selected from G_{post}. A more general test would loosen these restrictions, and take all paralogs into account. Another important extension is the modification of these tests for clusters found via a whole genome scanning approach. Finally, to investigate hypotheses of multiple WGDs within the same lineage, tests for more than three regions sampled from the same genome are required.

Chapter 4

Ortholog Detection

In this chapter, I use gene cluster statistics to develop a new method for identifying orthologs, motivated by the idea that orthologs will appear in similar genomic contexts more often then paralogs. Recall that two genes in different species are orthologous if they arose from a single gene in the most recent common ancestor (MRCA) of the two species, and paralogous if they arose through a duplication event that preceded the divergence of the species [59, 61]. These relationships are illustrated in Figures 4.1(a) and 4.1(b).

Orthologs are thought of as direct evolutionary counterparts: when we refer to ’the same gene in different species’, we typically mean orthologs. Thus, orthologs are the fundamental unit of comparison in many comparative genomics studies, and there are a variety of applications that require high-throughput methods for accurately identifying orthologs in genome-scale datasets. Traditional methods for ortholog identification are based on comparison of gene sequences. However, many additional sources of information can be used in addition to sequence comparison. Comparisons of genomic spatial organization have recently been used to augment sequence information, and improve ortholog prediction.

In this chapter, we combine our previous statistical work on testing the significance of max-gap clusters with a new algorithmic approach for finding max-gap clusters. By joining these two components, we design a novel method for orthology prediction based on both sequence comparison and spatial organization. We show that the use of the flexible max-gap cluster definition combined with our statistical approach for ranking gene clusters consistently reduces the number of orthologs missed (false negatives), without increasing the number of paralogs identified as orthologs (false positives), compared to previous approaches based on spatial analysis.

The rest of this chapter is organized as follows. In Section 4.1, I describe some of the applications that require genome-wide ortholog detection, and review the approaches that have been developed for this problem. In Section 4.2, I introduce a general graph-based framework that is used in the majority of context-based orthology detection methods. In Section 4.3, I describe existing methods that consider spatial organization in order to improve ortholog identification, and discuss the limitations of these methods. I give an overview of my approach in Section 4.4, then discuss each of the main contributions in detail. In Section 4.6, I present empirical results on a set of alpha-bacterial genomes, and compare my method’s performance with previous results on this dataset. Finally, in Section 4.7, I end by outlining possible improvements to this approach.

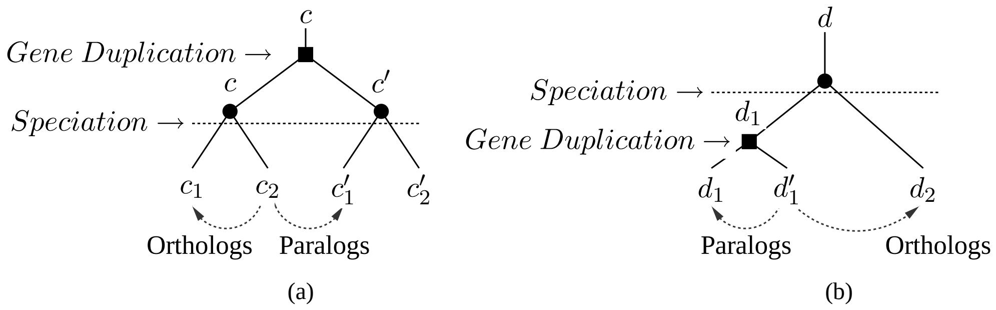


Figure 4.1: (a) A gene tree showing the evolution of the hypothetical c gene family. Gene c in genome G undergoes a gene duplication, giving rise to its paralog c'. A speciation event occurs, which gives rise to genes c_1, c'_1, c_2, and c'_2. Genes c_1 and c_2 are orthologs that arose from gene c in the MRCA, whereas genes c'_1 and c'_2 are orthologs that arose from gene c' in the MRCA. The remaining gene pairs are paralogs. (b) A gene tree showing the evolution of the hypothetical d gene family. A single copy of the d gene family exists in genome G. A speciation event occurs, which gives rise to d_1 and d_2. A subsequent duplication of gene d_1 in G_1 gives rise to d'_1. Genes d_1 and d'_1 are paralogs, and are both orthologous to gene d_2.

4.1 Background

Identification of orthologs is a prerequisite for a wide range of functional and evolutionary problems that can be approached through comparative genomics.

One application is predicting the functions of genes in newly sequenced genomes. The number of sequenced genomes is growing rapidly, too quickly for gene functions to be determined experimentally. Given a newly sequenced genome, we would like to infer the function of its genes from the function of related genes in well-studied model organisms. Since orthologs share a direct evolutionary relationship, they often have similar functions [56, 100, 111, 147]. Distinguishing orthologs from paralogs is considered an essential step for accurate transfer of experimental knowledge between species [119].

Other types of functional investigations also rely on orthologs. In phylogenetic foot-printing, transcription factor binding sites and other functionally important non-coding sequences are identified by searching for conserved sequences near orthologous genes. In addition, researchers often find it useful to distinguish orthologs from paralogs when studying the evolution of gene expression or how protein interaction networks differ among related organisms.

Finally, since orthologs arise through speciation, they play a key role in inferring evolutionary histories. To infer phylogenetic relationships among species, it is essential that only orthologous genes are analyzed. In addition, in comparisons of genome organization and genome rearrangements, orthologs are often used as markers, in order to identify orthologous chromosomal segments.

Existing Methods for Orthology Detection

Most methods for assigning orthologs start by constructing a set of ortholog candidates via sequence comparison. An all-against-all comparison of genome G_1 and genome G_2 is conducted to identify homologous gene pairs. For each gene, a set of homologs is selected, which serve as candidate orthologs. Frequently,

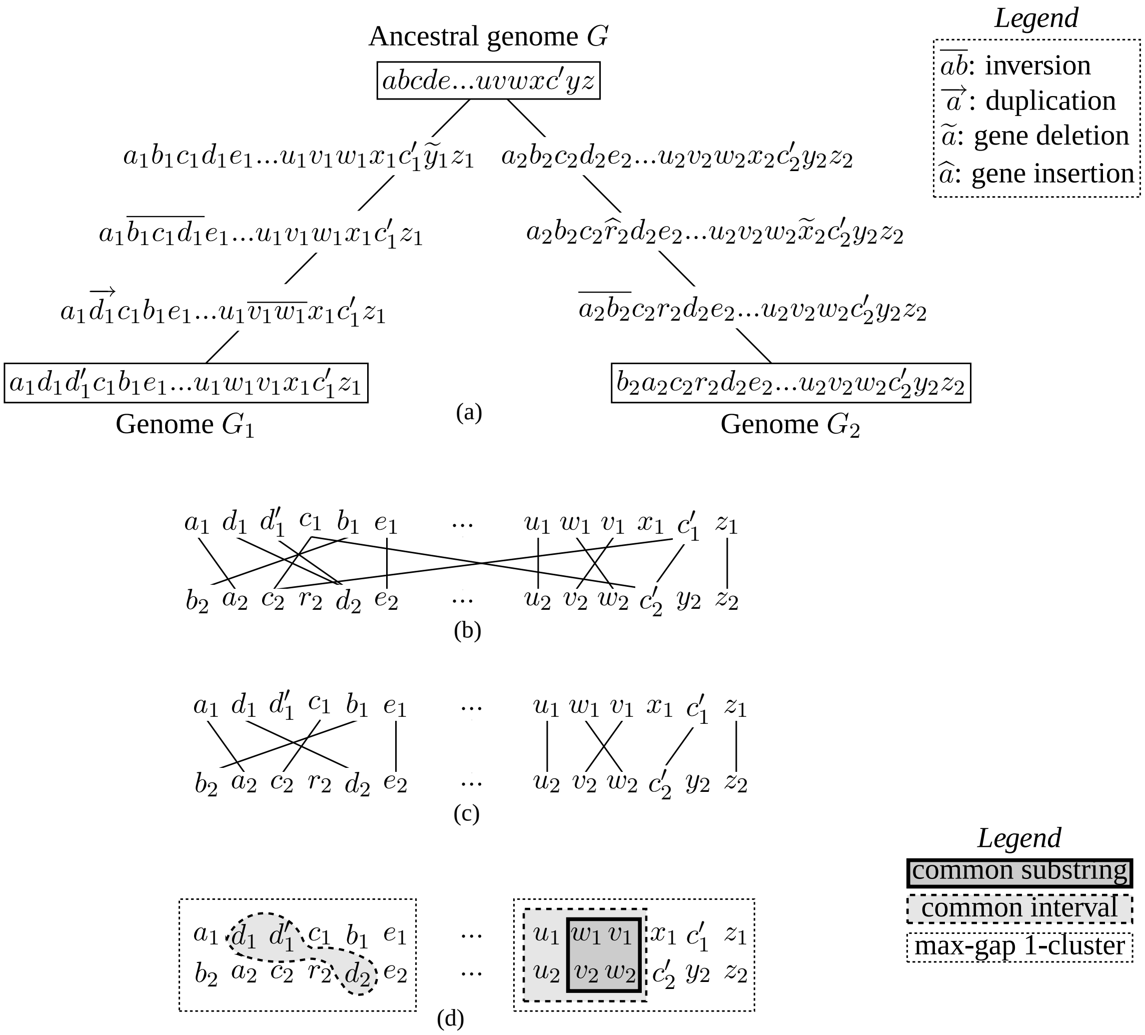

Ancestral genome G
$abcde...uvwxc'yz$
$a_1b_1c_1d_1e_1...u_1v_1w_1x_1c'_1\widetilde{y}_1z_1$
$a_2b_2c_2d_2e_2...u_2v_2w_2x_2c'_2y_2z_2$
$a_1\overline{b_1c_1d_1}e_1...u_1v_1w_1x_1c'_1z_1$
$a_2b_2c_2\widehat{r}_2d_2e_2...u_2v_2w_2\widetilde{x}_2c'_2y_2z_2$
$a_1\overrightarrow{d_1}c_1b_1e_1...u_1\overline{v_1w_1}x_1c'_1z_1$
$\overline{a_2b_2}c_2r_2d_2e_2...u_2v_2w_2c'_2y_2z_2$
$a_1d_1d'_1c_1b_1e_1...u_1w_1v_1x_1c'_1z_1$
Genome G_1
(a)
$b_2a_2c_2r_2d_2e_2...u_2v_2w_2c'_2y_2z_2$
Genome G_2
Legend
$\overline{ab}$: inversion
$\overrightarrow{a}$: duplication
$\widetilde{a}$: gene deletion
$\widehat{a}$: gene insertion
a_1 d_1 d'_1 c_1 b_1 e_1 ... u_1 w_1 v_1 x_1 c'_1 z_1
b_2 a_2 c_2 r_2 d_2 e_2 ... u_2 v_2 w_2 c'_2 y_2 z_2
(b)
a_1 d_1 d'_1 c_1 b_1 e_1 ... u_1 w_1 v_1 x_1 c'_1 z_1
b_2 a_2 c_2 r_2 d_2 e_2 ... u_2 v_2 w_2 c'_2 y_2 z_2
(c)
Legend
common substring
common interval
max-gap 1-cluster
a_1 d_1 d'_1 c_1 b_1 e_1 ... u_1 w_1 v_1 x_1 c'_1 z_1
b_2 a_2 c_2 r_2 d_2 e_2 ... u_2 v_2 w_2 c'_2 y_2 z_2
(d)

Figure 4.2: (a) Two hypothetical modern-day genomes, and the genome of their most recent common ancestor (MRCA). Genome G_1 is in species S_1, genome G_2 is in species S_2, and genome G is in the ancestral species S. Rearrangement events are shown to illustrate the evolution of spatial organization. (b) A map comparison of genomes G_1 and G_2, represented as a bipartite graph. (c) A matching of the genes in G_1 and G_2. (d) The conserved blocks shared between G_1 and G_2, according to three different definitions: common substrings, common intervals, and max-gap clusters ($g = 1$).

any pair of genes with sequence similarity above a set threshold is considered homologous. In other cases the requirement is more stringent: a gene must not only be similar to the query gene, but must score within a fixed percentage of the highest scoring match, or be one of the k highest scoring matches. Sometimes no fixed similarity threshold is applied—for each query gene the k most similar genes are kept as candidate orthologs. Many more variants have been proposed, but regardless of the details of the method, the end result is a set of homologous gene pairs. The problem is then to determine which of these homologous pairs are orthologs, and which are paralogs.

One way to distinguish orthologs from paralogs is to construct a gene family tree, then reconcile it with the corresponding species tree to infer speciation and duplication events [39, 60, 68, 187, 158]. This approach is challenging to apply on a genome-wide scale, however, because it is resource-intensive and error-prone [22]. With this method, the accuracy of ortholog assignments depends on the accuracy and information content of the multiple sequence alignment (MSA), and the accuracy of the estimated phylogeny. However, current methods for automatically generating MSAs yield alignments of poor quality when sequences are not highly similar, and so MSAs often require hand-curation. Even with the best possible MSA, there is often not sufficient information in the MSA to infer an accurate gene tree. Furthermore, this method requires building a new tree for each family of interest. Building gene trees is NP-hard; even the best heuristics are time-consuming, and are not guaranteed to find the correct tree topology, particularly when gene sequences are highly divergent. Although accuracy of the inferred tree can be assessed through bootstrap analysis, this type of analysis is impractical for genome-scale datasets.

Thus, many orthology predictions methods do not try to explicitly build a tree, but instead consider only pairwise sequence similarity. The simplest approach assumes genes are orthologs if they form reciprocal best hits, or *bi-directional best hits* (BBHs) [112, 158, 86, 166]. However, this method assumes that protein similarity accurately reflects evolutionary distance, that all genes within a family evolve at equal rates, and that gene predictions are correct and complete. As a result, domain shuffling, fused proteins, high sequence diversity within a family, incomplete genome sequencing, and errors in gene prediction can all lead to errors. For example, in Figure 4.1(b), if the best hit of gene d_2 is d_1', then the orthology of d_1 and d_2 will not be detected. Furthermore, if d_2 was later duplicated, giving rise to d_2', then the BBH method may identify only a single pair of orthologs. Since d_2 and d_2' were duplicated recently, they will have very similar sequences, and it could easily be the case that the best hit of gene d_1' is gene d_2, the best hit of d_2' is d_1, and the best hit of d_1 is d_2'. In this case, only (d_1',d_2) would be returned as an orthologous pair. Gene loss also leads to errors. For example, in Figure 4.1(a), if c_1 and c_2' are lost, then c_1' and c_2 would be BBHs, and would be incorrectly classified as orthologs.

More complex approaches have been designed to overcome some of these limitations. The COGs method [165] tries to reduce false positives by identifying orthologs only if they form triangles of BBHs shared between three distantly related species. Triangles that share a side are then merged into a single orthologous group. This merging step is designed to decrease false negatives by allowing many-to-many orthology relationships. Given a particular pair of species of interest, however, the COG groupings are often too coarse. Orthology sets are often very large, and contain genes that diverged prior to the speciation event of interest. OrthoMCL [99] and InParanoid [134] attempt to reduce false negatives by using clustering algorithms that group together similar sequences even if they do not form BBHs. In addition, the OrthoMCL algorithm attempts to eliminate spurious matches due to shared domains and protein fusions. Even these more sophisticated approaches are limited by their reliance on sequence information alone.

Other approaches have been developed that augment sequence data with orthogonal information sources, such as functional or regulatory data. For example, Bandyopadhyay *et al.* [5] infer orthologs based on the gene interaction network. They assume that genes whose network neighbors are orthologs are more likely to be orthologs. A Markov Random Field is created that models the orthology relation between each pair of proteins as a probabilistic function of the orthology relations of their immediate network neighbors. Gibbs sampling is used to compute the probability of orthology for each gene pair. Che *et al.* [36] supplement sequence data with operon boundaries. The assumption is that if two genes are in the same operon, then their orthologs are likely to also be in the same operon. Zheng *et al.* [185] identify BBHs, but then filter these predictions based on functional annotations: if the pair of proteins are classified in different functional subfamilies, then they are not considered orthologs. These methods are not applicable to most newly sequenced genomes in which little functional, transcriptional, or regulatory data is available, however, or if orthologs are being identified in order to infer gene function.

Comparisons of spatial organization also contribute evidence of orthology that is orthogonal to evidence provided by gene sequence comparisons. Figure 4.2(a) shows a hypothetical genome S that is replicated by speciation, yielding genomes S_1 and S_2, that subsequently diverge through small-scale and large-scale evolutionary changes. Shared genomic context combined with sequence similarity is thought to be a better indicator of orthology than sequence similarity alone. For example, consider the members of gene family c in Figure 4.2(a). Genes c and c' are paralogs that arose through a single gene duplication prior to the separation of species S_1 and S_2. They are located in distinct chromosomal regions in G, the genome of the MRCA of species S_1 and S_2. A speciation event results in two copies of c and c', one in G_1 and the other in G_2. Immediately following the speciation, the orthologs c_1 and c_2 appear in identical contexts, *i.e.* they have the same neighboring genes in the same order. The same is true of c'_1 and c'_2. The paralogs c_1 and c'_2, and c_2 and c'_1, on the other hand, appear in very different genomic contexts. Thus, by comparing gene neighborhoods, it is possible to determine that c_1 is orthologous to c_2 and not to c'_2. Over time, the genomic context of the orthologs will diverge due to genomic rearrangements. However, in many cases the regions will remain similar enough to detect orthology. For example, in the genomes of S_1 and S_2, c_1 and c_2 are both within two genes of a, b, d, and e. Similarly, c'_1 and c'_2 are both within three genes of u, v, w, and z. In contrast, there are no shared genes in the local neighborhoods of c_1 and c'_2.

4.2 A Graph-Based Framework for Orthology Detection

Before we review existing methods for incorporating spatial organization into ortholog prediction, we introduce the graph-based representation of the data used by many of these approaches, and describe the various types of output they generate.

4.2.1 Input

Given a set of homologous gene pairs, a bipartite homology graph $\mathcal{H} = (V_1 \cup V_2, E)$ is constructed. Vertices in V_1 and V_2 represent genes in G_1 and G_2 respectively. Given v_1 from V_1 and v_2 from V_2, $(v1, v2)$ is an edge in E if v_1 and v_2 are homologous. Most often, $\mathcal{H}$ is an undirected[1], unweighted graph. In a few cases, edge weights are assigned based on sequence similarity scores, and a weighted graph is constructed.

If true homology relationships were known, genes would form *gene families*, in which every gene in

[1] Depending on the strategy for identifying homologous pairs, the inferred homology relationship may not be symmetric, *i.e.* gene a's list of homologs may contain gene b, but not vice versa. In this case, an additional pre-processing step is required to enforce symmetry.

the family is homologous to every other gene in the family since each gene in a family arose from a single ancestral gene. However, due to noise, limitations of sequence comparison methods, or a stringent similarity threshold, not all homologous pairs will be identified, and the inferred homology relationship may not always be transitive, *e.g.* in the bipartite homology graph there may be a gene a that is homologous to b and c, and another gene d that is homologous to gene b but not gene c. Since many of the algorithms designed for this problem require gene families as input, the transitive closure[2] of $\mathcal{H}$ is often used as the input graph (omitting edges between genes in the same genome). We call this the family graph, $\mathcal{F}$. An example is shown in Figure 4.2(b). There is an edge between two genes in $\mathcal{F}$ if and only if they are in different genomes, and they are in the same connected component in $\mathcal{H}$. The graph $\mathcal{F}$ is composed of a set of connected components, each corresponding to a family. Gene a is said to be in the same family as gene b iff a and b are in the same connected component in $\mathcal{F}$.

Applying the transitive closure has the effect of adding edges to the homology graph. In some cases, these edges will correspond to homologs that were not identified due to weak sequence similarity. In other cases, these edges may be false predictions due to *domain chaining*, in which genes are erroneously considered homologous because they share an inserted domain. For orthology identification, it is critical that all orthologs be identified as homologs, but it is not important that all homologs be represented in the input graph. In fact, ideally, the input graph would contain the *smallest* set of homologous genes that is likely to contain the true ortholog. For ortholog identification, adding edges may just introduce noise, and decrease performance. That said, taking the transitive closure is still a common practice in orthology-detection methods, because it is the norm in other applications in which a homology graph is constructed, and because gene families often simplify algorithms.

4.2.2 Output

Given the graph $\mathcal{F}$ as input, the typical output is a graph $\mathcal{O}$, called the orthology graph, that is a sub-graph of $\mathcal{F}$ that forms a matching, *i.e.* each vertex is incident to at most one edge. Genes connected by edges in $\mathcal{O}$ are considered orthologs. Genes in the same family that are not connected by an edge in $\mathcal{O}$ are considered paralogs. The orthology graph may take one of three forms, as follows.

In the *exemplar* approach, a single exemplar gene is selected from each family. In other words, edges are pruned from $\mathcal{F}$ until each connected component representing a family contains exactly two genes, one from each genome. The *exemplar* of each family (also called the *main ortholog* [63], or the *positional ortholog* [27]) is thought to represent the gene that best reflects the original position of the ancestral gene family progenitor [138]. One motivation for seeking exemplars is that they are "more likely to be functional counterparts since they are both evolutionary and positional counterparts." [63]. The assumption underlying this approach is that the MRCA S had only a single gene in each gene family, and *all* duplications occurred after speciation, by separate lineage specific expansions in the lineages leading to S_1 and S_2. If the ancestral genome contained paralogs, then there may be more than one pair of orthologs within a family, and this approach will identify only a subset of the orthologs.

A second approach seeks a maximal matching of $\mathcal{F}$. In this case more than one orthologous pair can be identified per family. This approach assumes that all copies of a gene family were present in the MRCA, and *no* duplications occurred after speciation. With this approach, when co-orthologs are present, only one will be identified. For example, in Figure 4.2(b), gene d_2 can only be matched with gene d_1 *or* d_1', but not

[2] The transitive closure of a graph $G = (V, E)$ is a graph $G+ = (V, E+)$ such that $E+$ contains an edge (v, w) iff G contains a non-null path from v to w.

both. Note that this method generates a maximal matching, but it is not guaranteed to be a *perfect* matching. When the number of representatives of each gene family is not be the same in both genomes, some genes will not be assigned an ortholog. This is illustrated in the maximum matching shown in Figure 4.2(c), in which d_1' is not assigned an ortholog.

In the most general approach, edges are pruned from $\mathcal{F}$, but a one-to-one matching is not required. The output may include one-to-many or even many-to-many mappings between genes. These genes are considered *co-orthologs*, genes that arose by duplication subsequent to the speciation. For example, in Figure 4.2(a), d_1 and d_1' are co-orthologs to d_2, since the duplication of gene d_1 occurred subsequent to the speciation of S_1 and S_2. This model makes no assumptions about the relative timing of speciation and duplication events.

4.3 Related Work

The use of genomic context to augment sequence data in orthology detection has received considerable attention in recent years, both in practical efforts to build orthology databases, and in theoretical work on genome rearrangements.

A number of software tools for identifying orthologs use genomic context as auxiliary information to improve ortholog predictions based on sequence similarity. Typically, these heuristics identify unambiguous orthologs (often BBHs) that form collinear blocks, *i.e.* regions with perfectly conserved gene order. Gene pairs with sufficiently strong sequence similarity are matched if they appear within or near a collinear block, even if they have a better sequence match elsewhere in the genome [29, 28, 31, 41, 94, 27, 178, 185]. In addition, such methods sometimes feature a post-processing step in which genes with extremely low or even no detectable similarity are assigned as orthologs if they appear in a collinear block, and no other potential ortholog was identified [29].

The use of multiple genome comparisons can increase the accuracy of these methods since a gene is likely to be in a collinear block in at least a subset of the genomes. Once a subset of the orthologs have been identified, additional orthologs may be assigned by comparison with a third genome. For example, if genes $bcde$ are adjacent in genome G_1, genes abc are adjacent in G_2, and genes $cdef$ are adjacent in G_3, it can be inferred that gene c in G_2 is orthologous to gene c in G_3, even though they share no genomic context.

Methods based on collinear blocks of unambiguous orthologs have been successful when comparing genomes in which local gene order is well-conserved, such as ascomycete fungi [94, 29, 28]. However, in more diverged genomes such an approach may be less successful, because fewer orthologs will be immediately unambiguous, and order within orthologous segments will be more scrambled. More complex methods based on the family graph presented in Section 4.2 have been developed to handle these cases.

Perhaps the earliest attempt to solve this problem within a graph-based framework is that of Bansal *et al.* [7]. They propose a heuristic consisting of two steps. In the first step the Hungarian method [17] is used to find a maximal matching in the weighted bipartite graph. Based on these matches, in the second step max-gap gene clusters are identified. The weights of edges between genes in all large clusters are increased, and the remaining edge weights are decreased. The algorithm iterates between these two steps, but does not converge. This method has no statistical basis, nor explicit optimization criteria.

Many of the recent methods for orthology identification based on genomic context can be classified into one of two basic approaches. The first seeks to select orthologs that minimize some measure of distance between the two genomes, and the second strives to selects orthologs to maximize conservation of spatial organization. Unlike the methods discussed above, all of these approaches discard sequence similarity scores, and use an unweighted family graph.

4.3.1 Minimizing Rearrangement Distance

The first exemplar approach to the problem of orthology identification based on genomic context sought a set of orthologs that minimized the number of *breakpoints*, the number of pairs of genes that are adjacent in one genome but not in the other [138]. When local order is conserved, minimizing breakpoints may be helpful for orthology detection. When local order is scrambled, however, the breakpoint distance will be less useful, since it only considers adjacent genes, not local neighborhoods. For example, the breakpoint distance does not help us choose between the two possible assignments of the c gene family in the genomes shown in Figure 4.2(a). If c_1 is matched with c_2, then it creates two breakpoints, but if c_1 is matched with c'_2, it also creates two breakpoints. Similarly, regardless of whether c'_1 is matched with c'_2 or c_2, two breakpoints result.

More recent approaches define the distance between two genomes in terms of a specified set of rearrangement operations. Given this set of rearrangement operations, a matching that corresponds to the most parsimonious evolutionary history of rearrangements is sought. Ortholog assignment is then formulated as the problem of transforming one genome into the other with the smallest number of rearrangement events. Within both the exemplar and matching framework, different sets of rearrangement operations have been applied to this problem, including reversals [38, 138, 160], reversals and translocations [63], or duplications, transpositions, and reversals [52]. This approach is challenging because for even a simple set of operations, finding the most parsimonious scenario is NP-hard [38]. In addition, this approach is based on the assumption that the underlying evolutionary model can be explained by a small set of rearrangement operations. Finally, relative costs must be assigned to each operation to reflect the underlying frequency of such events, but such frequencies are often genome-dependent, and typically not known.

4.3.2 Maximizing Spatial Conservation

Another common approach is to select an ortholog assignment that maximizes conservation of spatial organization. These approaches are typically based on some notion of a *conserved block*. The underlying assumption is that chromosomal segments that form a conserved block arose from a single chromosomal segment in the MRCA, *i.e.* the regions are orthologous. Thus, the genes within the block are also likely to be orthologous. Each conserved block can be thought of as specifying a local ortholog assignment. A global mapping can then be constructed based on these local mappings. However, if a gene appears in more than one conserved block, these blocks may imply different orthology assignments, in which case they are *inconsistent*.

With this approach, the goal is to select a consistent subset of the conserved blocks, such that every gene appears in at least one conserved block, *i.e.* the blocks *cover* both genomes. Typically, a greedy heuristic is used [15, 14, 161]. The set of all maximal[3] conserved blocks is identified in a pre-processing step. The procedure repeatedly selects the longest maximal conserved block from this set, assigns orthologs within the block, and then removes all remaining blocks that are inconsistent with the new partial assignment [15, 14, 161].

[3] A conserved block is considered maximal if it is not included within any larger conserved block.

A number of methods based on this framework have been proposed. Two methods differ in two main ways: the precise definition of conserved block and the optimization criterion used to select the set of blocks that specify the matching. Two definitions of a conserved block have been investigated within this framework. The most constrained definition equates a conserved block to a *common substring*: two sets of contiguous genes in identical (or reversed) order, with identical gene content [161, 14]. This definition is very stringent, as it does not allow a single insertion, deletion, or inversion to occur within a conserved block. Like the breakpoint distance, this measure of conservation is most useful when gene order is highly conserved. For example, in Figure 4.2(d), the only common substring with length greater than one is vw, so identifying common substrings will not help determine orthology relationships for the genes in family c.

The *common interval*, another block definition that has been used, is less constrained. A common interval is defined to be two sets of contiguous genes, representing the same set of gene families, in any order [19, 14, 13]. In other words, the set of gene families contained within the block must be identical in both genomes, although the number of representatives of each family may differ. For example, in Figure 4.2(d), the common intervals are $(\{u_1, w_1, v_1\}, \{u_2, v_2, w_2\})$ and $(\{d_1, d'_1\}, \{d_2\})$. Common intervals are much more inclusive than common substrings. They allow rearrangements, as well as many-to-one or many-to-many relationships within the interval. This means there can be local duplications or deletions after the speciation, as long as at least one representative gene for each family remains in the interval. However, insertions of unrelated genes are still not allowed, nor is the deletion of a single-copy gene. As a consequence, this definition of conserved block is still not general enough to identify the two conserved, but scrambled, regions $(\{a_1, b_1, c_1, d_1, d'_1, e_1\}, \{a_2, b_2, c_2, d_2, e_2\})$ and $(\{c'_1, u_1, v_1, w_1, z_1\}, \{c'_2, u_2, v_2, w_2, z_2\})$, in Figure 4.2(d).

Different optimization criteria have been proposed for determining which subset of conserved blocks is best. One approach is to select a set of consistent, maximal conserved blocks, such that the the total number of conserved blocks is *minimized* [161, 14, 13]. This is based on the assumption that genes that appear in longer conserved blocks are more likely to be orthologs. Whether or not this assumption is justified has never been investigated. Since these optimization problems have been shown to be NP-hard [13, 16, 15, 19, 24, 35], existing methods rely on greedy heuristics.

A somewhat different approach is used by Bourque *et al.* [19], who seek a *maximum* cardinality subset of consistent blocks. These blocks need not be maximal however; in fact, they may even be nested, such that one block completely contains another. The motivation for maximizing the number of blocks is that genomes with similar gene order will have many conserved blocks, whereas randomly ordered genomes will have few. Bourque *et al.* [19] reduce the problem of finding the maximum number of compatible blocks to a MAX-SAT problem. They design their own MAX-SAT heuristic since their clauses are not in conjunctive normal form, and so no direct MAX-SAT solver can be used.

There is a close relationship between maximizing spatial conservation and minimizing rearrangement distances. For certain block definitions it has been proven that minimizing the number of maximal blocks is equivalent to minimizing the rearrangement distance. For example, the ortholog assignment corresponding to the smallest set of common substrings that cover both genomes will also be the ortholog assignment that requires the fewest inversions to transform one genome into the other [161]. More generally, choosing a definition of a conserved block is comparable to choosing a set of rearrangement operations. For example, allowing gaps in a conserved block definition is similar to adding insertion or deletion to a set of rearrangement operations.

4.4 Our Approach

Previous methods based on maximization of spatial conservation rely on very restrictive definitions of a conserved block. Neither the common substring, nor the common interval definition allows gaps within the block. When comparing more distantly related genomes, these conservative definitions may fail to detect orthologous regions in which neither gene order nor gene content are identical. To address this, we present a new orthology detection algorithm on a more general definition of a conserved block, the max-gap cluster presented in Section 2. The max-gap cluster is the most general definition of a conserved block that is used in practice, and for which efficient search algorithms have been developed. This cluster definition allows for scrambled gene order, gene loss, gene insertions, as well as tandem duplications. In addition, unlike the methods described in Section 4.3.2, our method also accepts a weighted homology graph, and we propose a number of ways to integrate sequence similarity scores into our framework.

Allowing a conserved block to contain gaps poses a number of new challenges that do not arise with more conservative block definitions. Since conserved blocks may be very sparse, and gene order scrambled, truly orthologous chromosomal regions are more easily confused with regions that share a few genes just by chance. Although using a more flexible definition should decrease the number of false negatives that arise due to failure to detect spatial conservation, this is offset by the risk of generating more false positives due to incorrectly identifying regions that simply share a few genes by chance as orthologous regions.

A second, related challenge arises with the introduction of gaps. If a gene appears in two distinct, but inconsistent, clusters, we must decide which cluster is more likely to represent a pair of orthologous chromosomal segments. With previous definitions of conserved blocks, whether common substrings or conserved intervals, larger blocks were always preferred over smaller blocks. This reflects the intuition that larger blocks are less likely to occur by chance, and thus are more likely to indicate orthology of the entire region. This assumption has never been tested, however. Although this assumption seems reasonable with the common substring definition, in which gene content and order are identical, once duplicates are allowed within the cluster, such as with common intervals, it is more speculative. Furthermore, once gaps are allowed this assumption clearly no longer holds—a longer block with more genes but with large gaps may be less indicative of common ancestry of a region than a smaller block with fewer gaps. A key challenge of this more general framework is therefore how to compare two conserved blocks of different sizes and lengths, and determine which one is more likely to represent a pair of orthologous chromosomal segments.

We address the two challenges above by using statistical significance of a cluster as a measure of conservation. The key idea of this approach is to rank clusters based on their probability of occurring by chance under a null model of random gene order. The underlying assumption is that the smaller the probability that a cluster would occur by chance, the more likely the cluster indicates orthology of the entire region, and thus the more likely the genes within the cluster are orthologous. This approach can be used for any cluster definition, including one with gaps. The only requirement is that a test statistic be selected, such that the probability of a cluster decreases as the value of the test statistic increases.

A third challenge that arises when gaps are introduced is it becomes more difficult to determine when blocks conflict. Maximal common substrings conflict whenever the gene span of one substring overlaps the gene span of another. In this case, all conflicting blocks can be identified and removed in time proportional to the length of the block. Once gaps are introduced, however, identifying all clusters that conflict with a selected cluster is more difficult. Even if the gene spans of two clusters overlap, the clusters may still be compatible. To address this issue, rather than removing all conflicting clusters immediately upon selecting a lowest-cost cluster, we take a lazy approach: we check whether a cluster is invalid only when it is selected as the current lowest-cost cluster.

One last challenge is that our approach requires an algorithm for finding all highly significant max-gap clusters. Existing algorithms only find maximal max-gap clusters. Since a shorter cluster with fewer gaps may be more significant than a longer cluster with more gaps, it is desirable to consider non-maximal clusters, as well as maximal ones, when assigning orthologs. To this end, below we will define a new type of max-gap cluster, called a *dominant* max-gap cluster, which can be proven to always be more significant than any of the sub-clusters it dominates, in the case of clusters without duplicates.

We design a general ortholog detection approach by combining our previous statistical work on max-gap gene clusters with an extension of the max-gap cluster search algorithm designed by He and Goldwasser [76], to find dominant max-gap clusters. Before presenting our algorithm, we first introduce some technical preliminaries. We then give a high-level overview of our algorithm, which is followed by detailed presentations of our main contributions.

Technical Preliminaries

In this chapter, unlike previous chapters, we assume genes are partitioned into equivalence classes, *i.e.* gene families. In this case, the homology graph will have many-to-many homology relationships. In this section we revisit the definitions of a max-gap chain and cluster given in Section 2.1.1, and extend them to allow for a many-to-many homology mapping. In addition, in this chapter we require a new notion of a *dominant* cluster, which is also defined below.

As before, we model a genome as an ordered list of genes, ignoring gene orientation, physical distances between genes, and overlapping genes. If a genome contains multiple chromosomes, we assume they are concatenated in a fixed (but arbitrary) order. Each gene is now associated with a gene family in addition to its position on the genome.

Definition 4.4.1. *A genome is a triple $G = (\Sigma, X, F)$, where Σ is a set of gene families, $X = \{1, .., n\}$ is a sequence of genes, ordered by their position in the genome, and $F : 1, .., n \to \Sigma$ is a function mapping genes to gene families. $F^{-1}(f)$ denotes the subset of genes assigned to family f.*

From a pairs of genomes we can construct the corresponding family graph:

Definition 4.4.2. *Given two genomes $G_1 = (\Sigma_1, X, F_1)$ and $G_2 = (\Sigma_2, Y, F_2)$, $\mathcal{F} = (V_1 \cup V_2, E)$ is a bipartite graph, where a vertex $v_1 \in V_1$ represents a gene in G_1, a vertex $v_2 \in V_2$ represents a gene in G_2, and $(v_1, v_2) \in E$ iff $v_1 \in V_1$, $v_2 \in V_2$, and $F(v_1) = F(v_2)$.*

A maximum matching of $\mathcal{F}$ is of size $\nu = \sum_{f \in \Sigma_1 \cup \Sigma_2} \min(|F_1^{-1}(f)|, |F_2^{-1}(f)|)$

In order to define a max-gap cluster in the presence of gene families, we first recall from Section 2.1.1 the definitions of the max-gap of a set of genes on a single chromosome, and of a g-chain:

Definition 4.4.3. *Given genome $G = (\Sigma, X, F)$ containing two genes i and j, the **gap** between i and j is defined as $\Delta(i, j) = |i - j| - 1$, if the genes are on the same chromosome, and $\Delta(i, j) = \infty$ if the genes are on different chromosomes. Given a (not necessarily contiguous) subset of genes $X' \subseteq X$, we define $\Delta(X')$, the **max-gap** of X', as the maximum gap over all pairs of adjacent genes in X'. We say that $X' \subseteq X$ is a g-**chain** of C if $\Delta(X') \leq g$. The set of families occurring in X' is denoted $\Sigma(X') = \{F(i) \mid i \in X'\}$.*

For example, consider the two genomes shown in Figure 4.2(b):

$$G_1 = a_1 d_1 d'_1 c_1 b_1 e_1 * * * u_1 w_1 v_1 * c'_1 z_1$$
$$G_2 = b_2 a_2 c_2 * d_2 e_2 * * * u_2 v_2 w_2 c'_2 * z_2$$

where genes in the same family are assigned the same letter, and stars indicate genes with no homolog in the other genome. In this example, the gap between gene w_2 and gene c'_2 is zero, and the gap between w_2 and z_2 is two. The set $\{c'_2, w_2, z_2\}$ forms a 1-chain in G_2, as does $\{u_2, v_2, w_2\}$, which also forms a 0-chain.

Definition 4.4.4. *Given two genomes $G_1 = (\Sigma_1, X, F_1)$ and $G_2 = (\Sigma_2, Y, F_2)$, and two sets of genes, $X' \subseteq X$ and $Y' \subseteq Y$, the pair (X', Y') forms a **cluster** if X and Y contain the same gene families;* i.e. $\Sigma(X') = \Sigma(Y')$. *A cluster (X', Y') is a **sub-cluster** of (X^*, Y^*) if $X' \subseteq X^*$ and $Y' \subseteq Y^*$. The **max-gap** of a cluster (X', Y') is $\Delta(X', Y') = \max(\Delta(X'), \Delta(Y'))$.*

Definition 4.4.5. *A cluster (X', Y') forms a g-**cluster** if its max-gap $\Delta(X', Y') \leq g$. A g-cluster X is **maximal** if it is not contained within a larger g-cluster,* i.e. *there is no g-cluster (X^*, Y^*) such that $X^* \supseteq X'$, $Y^* \supseteq Y'$, and $(X', Y') \neq (X^*, Y^*)$.*

This definition of a cluster requires that the set of gene families be the same in each chain, but the number of representatives of each family in the two chains may differ. The order of the genes within the two chains may also differ, but the number of gaps between any pair of adjacent genes in a chain is constrained.

For the purpose of identifying orthologs, if there is a sub-cluster with a smaller gap, it may be useful to distinguish it from the larger cluster that contains it. For example, $S = (\{u_1, v_1, w_1\}, \{u_2, v_2, w_2\})$ is a 1-cluster, but it is not a maximal 1-cluster because it is contained in the larger 1-cluster $T = (\{c'_1, u_1, v_1, w_1, z_1\}, \{c'_2, u_2, v_2, w_2, z_2\})$. However, the max-gap of S is actually smaller than the max-gap of T, since S is also a 0-cluster. To address this issue, it is convenient to define the following:

Definition 4.4.6. *A g-cluster $C_1 = (X^*, Y^*)$ **dominates** a g-cluster $C_2 = (X', Y')$ if $X^* \supseteq X'$, $Y^* \supseteq Y'$,* and *the maximum gap of C_1 is at least as small as the maximum gap of C_2;* i.e. $\Delta(X^*, Y^*) \leq \Delta(X', Y')$. *A g-cluster is **dominant** if there is no cluster that dominates it.*

For example, $U = (\{c'_1, u_1, v_1, w_1\}, \{c'_2, u_2, v_2, w_2\})$ is dominated by $T = (\{c'_1, u_1, v_1, w_1, z_1\}, \{c'_2, u_2, v_2, w_2, z_2\})$ since both have a max-gap of $g = 1$, and T contains U. The cluster, $V = (\{u_1, v_1, w_1, z_1\}, \{u_2, v_2, w_2, z_2\})$ is also dominated by T since T has a smaller max-gap and contains V. However, although $S = (\{u_1, v_1, w_1\}, \{u_2, v_2, w_2\})$ is contained by T, it is not dominated by T, since S has a smaller max-gap. In fact, both S and T are dominant clusters. The list of all dominant clusters in G_1 and G_2 is given in Table 4.1.

Recall that a key idea of this approach is that to find a global matching of two genomes we identify significant gene clusters, which can be used to select a local matching. However, as defined above, a g-cluster does not necessarily specify a local matching. Note that a cluster (X', Y') is in essence a sub-graph of $\mathcal{F}$ with vertices corresponding to the subset of genes in X' and Y', and all edges whose endpoints are in these subsets. We call this the subgraph of $\mathcal{F}$ **induced** by (X', Y'). In many cases this sub-graph will be a matching. For example, each vertex in the sub-graph induced by $(\{u_1, v_1, w_1\}, \{u_2, v_2, w_2\})$ has degree exactly one. In other cases, the induced sub-graph will not be a matching since the two chains may contain *different* numbers of genes from a gene family. For example, the sub-graph induced by $(\{a_1, b_1, c_1, d_1, d'_1\}, \{a_2, b_2, c_2, d_2\})$ (shown in Figure 4.2(b)) contains two edges incident to d_2: one to d_1 and one to d'_1. Given such a cluster, in order to assign a unique ortholog from within the cluster to each gene, the problem is to select a maximum matching associated with the cluster.

g	Dominant g-clusters
0	$(\{u_1, v_1, w_1\}, \{u_2, v_2, w_2\}),\ (\{c_1\}, \{c_2\}),\ (\{c_1\}, \{c'_2\}),\ (\{c'_1\}, \{c_2\}),\ (\{c'_1\}, \{c'_2\})$ $(\{a_1\}, \{a_2\}),\ (\{b_1\}, \{b_2\}),\ (\{d_1, d'_1\}, \{d_2\}),\ (\{e_1\}, \{e_2\}),\ (\{z_1\}, \{z_2\})$
1	$(\{a_1, b_1, c_1, d_1, d'_1 e_1\}, \{a_2, b_2, c_2, d_2, e_2\}),\ \ (\{c'_1, u_1, v_1, w_1, z_1\}, \{c'_2, u_2, v_2, w_2, z_2\})$
2	none
3	$(\{a_1, b_1, c_1, c'_1, d_1, d'_1 e_1, u_1, v_1, w_1, z_1\}, \{a_2, b_2, c_2, c'_2, d_2, e_2, u_2, v_2, w_2, z_2\})$

Table 4.1: The set of max-clusters of $G_1 = a_1 d_1 d'_1 c_1 b_1 e_1 * * * u_1 w_1 v_1 * c'_1 z_1$ and $G_2 = b_2 a_2 c_2 * d_2 e_2 * * * u_2 v_2 w_2 c'_2 * z_2$.

Definition 4.4.7. *We say that the cluster $S_m = (X_m, Y_m)$ is an **associated matching** of the cluster $S = (X', Y')$, if S_m is a sub-cluster of S, and the sub-graph of $\mathcal{F}$ induced by S_m forms a matching. The **size** of a matching is the number of edges in the sub-graph it induces, which is equivalent to $|X_m| = |Y_m|$. An associated matching of S is a **maximum matching** associated with S if there is no associated matching of greater cardinality.*

There may be more than one maximum matching associated with the same cluster. For example, there are two maximum matchings associated with cluster $W = (\{a_1, b_1, c_1, d_1, d'_1\}, \{a_2, b_2, c_2, d_2\})$ depending on whether d_1 or d'_1 is matched with d_2: $(\{a_1, b_1, c_1, d_1\}, \{a_2, b_2, c_2, d_2\})$ and $(\{a_1, b_1, c_1, d'_1\}, \{a_2, b_2, c_2, d_2\})$. Both matchings have size four. The sub-cluster $(\{a_1, b_1, c_1\}, \{a_2, b_2, c_2\})$ is also a matching associated with W, but it is not maximum since it is only of size three.

Finally, given a cluster (X', Y') whose associated matching (X_m, Y_m) has size $h = |X_m|$ and max-gap $g = \Delta(X_m, Y_m)$, we introduce the notation $\Phi(h, g)$ to denote the **cost** of the cluster. Our implementation will allow any non-negative cost function to be used, but ideally, the cost of a cluster should be inversely related to the probability of observing such a cluster by chance.

4.5 The Algorithm

An overview of our ortholog detection algorithm is given in Algorithm 2. This algorithm takes as input two genomes of size n_1 and n_2, and a family assignment for each gene. In addition, the user must specify a maximum gap parameter $g_{\max}$. Although in theory the algorithm could identify all dominant g-clusters, for any value of g, for efficiency we restrict the search to only those dominant clusters with max-gap no greater than $g_{\max}$. Algorithm 2 follows the general framework described in Section 4.3.2. It differs from previous approaches in two ways. First, the max-gap definition is used to specify the conserved blocks. Second, rather than selecting the *largest* remaining cluster at each step, Algorithm 2 selects the lowest cost cluster, where the cost of a cluster is based on its probability of occurring by chance in a random genome. This strategy is designed to find a matching such that genes within clusters that are most significant are assigned as orthologs preferentially.

There are four main components: pre-computing matching costs, finding clusters, scoring clusters, and assigning orthologs. In the first step, we pre-compute matching costs $\Phi(h, g)$, for all possible values of $h > 0$ and $g \in 0..g_{max}$. Costs are either computed analytically, based on the equations presented in Section 2.3, or computed empirically, by randomly permuting gene order and counting how many clusters of different sizes and gaps are observed.

Algorithm 2 Ortholog Detection Algorithm

```
1: Compute the cost of a matching of size $h$ and max-gap $g$, for all $1 \leq h \leq H$ and $0 \leq g \leq g_{\max}$
2: Identify all dominant $g$-clusters, for all $g \in \{0..g_{\max}\}$
3: for each dominant cluster do
4:    Select a maximum matching
5:    Compute the cost of the matching.
6:    Insert the cluster into a priority queue, with priority equal to the cost of the associated matching
7: end for
8: while queue is not empty do
9:    Remove the lowest cost cluster from the queue.
10:   if the cluster is no longer valid then
11:      Add to the queue any sub-clusters that are now dominant.
12:   else
13:      Assign orthologs within the cluster, as specified by the associated matching.
14:   end if
15: end while
```

In the second step we identify all dominant g-clusters, where $g \in \{0..g_{\max}\}$. Next, for each cluster that we identified, we select an associated matching, based either on gene order or sequence similarity. We compute the size and max-gap of the matching, and from those quantities look up the cost of the cluster. We then insert the cluster into a priority queue, with priority equal to the cost of the associated matching. Note that it is possible that the max-gap of a cluster containing duplicates may be smaller than the max-gap of its maximum associated matching. As a result, in rare cases, the gap size of the associated matching might actually be larger than g_{max}. In this case the cluster is not inserted into the priority queue.

Finally, in the last step we construct a genome matching. We iteratively remove the lowest cost cluster from the priority queue. If any of the genes in the cluster have already been assigned an ortholog, then the cluster is no longer valid, and is discarded. In this case, sub-clusters that were previously dominated by the cluster may now be dominant. We identify any newly dominant sub-clusters, and add them to the queue. If the cluster is valid, then we assign all the gene pairs in its associated matching as orthologs. We continue this procedure until the queue is empty, and a global maximum matching has been selected.

Below we discuss in more detail our solution to the four main components of our algorithm: finding dominant clusters, finding an associated matching of a cluster, scoring clusters, and keeping the list of dominant clusters up to date.

4.5.1 Finding all dominant g-clusters

Line 2 of Algorithm 2 requires a method to identify all dominant g-clusters, for all values of g in $0..g_{\max}$. Given a fixed value of g, and a one-to-one homology mapping, the GeneTeams algorithm [8] has been designed for finding all maximal g-clusters in two genomes. He and Goldwasser [76] extended this approach to handle gene families, in a software tool called HomologyTeams.[4] However, the HomologyTeams algorithm identifies only maximal g-clusters, for a fixed value of g. For example, given $g = 3$, of the fourteen

[4] Although the software is entitled HomologyTeams, note that it cannot be applied to the general homology graph $\mathcal{H}$, but only the family graph $\mathcal{F}$.

dominant clusters shown in Table 4.1, HomologyTeams would return only the single large cluster shown in row $g = 3$. What we seek to determine, rather, is all dominant g-clusters, for all values of g in $0..g_{\max}$. There is a close relationship between maximal and dominant clusters, however, that suggests a modified version of the HomologyTeams algorithm for finding dominant clusters. In order to describe the algorithm, we first review the He and Goldwasser algorithm, then explain how we modify it to find dominant clusters. The HomologyTeams algorithm handles only single-chromosome genomes. We have modified their algorithm to compare multi-chromosomal genomes as well, but in order to simplify the exposition, in this section I assume each genome contains only one chromosome.

Both GeneTeams and HomologyTeams use a divide-and-conquer algorithm which begins by breaking one genome into runs of genes separated by a gap greater than g. For example, consider again the two genomes shown in Figure 4.2(b):

$$G_1 = a_1 d_1 d_1' c_1 b_1 e_1 * * * u_1 w_1 v_1 * c_1' z_1$$
$$G_2 = b_2 a_2 c_2 * d_2 e_2 * * * u_2 v_2 w_2 c_2' * z_2$$

Given $g = 2$, genome G_1 would be split into two runs, $X_1 = a_1 d_1 d_1' c_1 b_1 e_1$ and $X_2 = u_1 w_1 v_1 * c_1' z_1$, since they are separated by a gap greater than g. In GeneTeams, each division of G_1 specifies a unique division of G_2 into disjoint subsequences. In HomologyTeams, however, a gene may have more than one homolog, and so each run in G_1 is compared with the subsequence of G_2 formed by taking all genes with homologs in the run on G_1. For example, X_1 would be recursively compared with $Y_1 = b_2 a_2 c_2 * d_2 e_2 * * * * * * c_2'$, and X_2 would be compared with $Y_2 = c_2 * * * * * * u_2 v_2 w_2 c_2' * z_2$. Notice that the subsequences Y_1 and Y_2 are not disjoint since both contain the genes c_2' and c_2.

The HomologyTeams algorithm alternates between splitting genomes G_1 and G_2 on gaps greater than g, recursively breaking each one down into runs, and updating the current set of shared families (the *alphabet*), until two subsequences with no gap greater than g are reached. For example, in the comparison of X_1 and Y_1, Y_1 would be broken into two runs: $Y_{11} = b_2 a_2 c_2 * d_2 e_2$ and $Y_{12} = c_2'$. At this point X_1 has the same alphabet as Y_{11}, and since neither have a gap greater than $g = 2$, the recursion would halt, and (X_1, Y_{11}) would be returned as a maximal g-cluster. Y_{12} would be compared with the subsequence of X_1 with the same alphabet: $X_{11} = c_1$. Since neither X_{11} nor Y_{12} has a gap greater than $g = 2$, (X_{11}, Y_{12}) would be returned as a maximal g-cluster.

In order to modify this algorithm to identify dominant clusters, we note the relationship between maximal and dominant clusters:

Proposition 4.5.1. *Every maximal g-cluster is a dominant g-cluster.*

Proof. Let (X', Y') be a g-cluster with $\Delta(X', Y') = g' \leq g$. If (X', Y') is maximal then every cluster that contains it has gap greater than g. Therefore, there exists no cluster with a max-gap less than or equal to g' that contains (X', Y'). □

Proposition 4.5.2. *Every dominant g-cluster is a maximal g'-cluster, for some $g' < g$.*

Proof. Let (X', Y') be a dominant g-cluster with $\Delta(X', Y') = g' \leq g$. Since it is dominant, there is no cluster with max-gap $g^* \leq g'$ than contains it. Thus it is a maximal g'-cluster. □

These two propositions suggest a possible algorithm: run the HomologyTeams algorithm multiple times, for each value of g in $0..g_{\max}$. Proposition 4.5.1 guarantees that only dominant clusters will be returned, and Proposition 4.5.2 guarantees that all dominant clusters will be found. However, this naive approach is inefficient, since for small values of g much of the work of the algorithm is the same as for larger values of g. In addition, the same clusters could be output multiple times, since a dominant cluster may be a maximal g-cluster for many values of g. Thus, an additional post-processing step would be required to filter out the many redundant clusters returned. We present a more efficient algorithm based on the the following observations.

Lemma 4.5.1. (Beal *et al.* [8]) *If X_1 and X_2 are two g-chains of genome G, and $X_1 \cap X_2 \neq \emptyset$, then $X_1 \cup X_2$ is also a g-chain.*

Proof. The proof is given in the GeneTeams paper, as proof of Lemma 1. The existence of gene families does not alter this lemma. □

Lemma 4.5.2. *If (X_1, Y_1) and (X_2, Y_2) are two g-clusters, $X_1 \cap X_2 \neq \emptyset$, and $Y_1 \cap Y_2 \neq \emptyset$, then $(X_1 \cup X_2, Y_1 \cup Y_2)$ is also a g-cluster.*

Proof. Since (X_1, Y_1) is a cluster, $\Sigma(X_1) = \Sigma(Y_1)$. Similarly, $\Sigma(X_2) = \Sigma(Y_2)$. Thus, $\Sigma(X_1 \cup X_2) = \Sigma(Y_1 \cup Y_2)$, and $(X_1 \cup X_2, Y_1 \cup Y_2)$ is a cluster. By Lemma 4.5.1, $X_1 \cup X_2$ is a g-chain, as is $Y_1 \cup Y_2$. Hence, $\Delta(X_1 \cup X_2, Y_1 \cup Y_2) \leq g$, and $(X_1 \cup X_2, Y_1 \cup Y_2)$ is a g-cluster. □

Proposition 4.5.3. *Either a g-cluster is a maximal g-cluster, or there exists a unique maximal g-cluster that contains it.*

Proof. Let (X', Y') be a non-maximal g-cluster with $\Delta(X', Y') = g' \leq g$. Since it is non-maximal, there is some maximal g-cluster (X_1, Y_1) that contains it. Assume there is another maximal g-cluster (X_2, Y_2) that also contains (X', Y'). Clearly, $X_1 \cap X_2 \neq \emptyset$ and $Y_1 \cap Y_2 \neq \emptyset$ since both X_1 and X_2 contain X', and both Y_1 and Y_2 contain Y'. By Lemma 4.5.2, $(X_1 \cup X_2, Y_1 \cup Y_2)$ is also a g-cluster. However, (X_1, Y_1) is maximal, so there is no larger g-cluster that contains it. Thus $(X_1, Y_1) = (X_2, Y_2)$. □

These propositions guarantee that the following modification of the He and Goldwasser algorithm efficiently identifies all dominant clusters. The existing HomologyTeams algorithm is used to find all maximal $g_{\max}$-clusters (which are guaranteed to be dominant $g_{\max}$-clusters by Proposition 4.5.1). When a maximal $g_{\max}$-cluster (X', Y') with max-gap $g \leq g_{\max}$ is found, rather than outputting it and halting, we reduce the maximum allowed gap from $g_{\max}$ to $g - 1$, and recursively identify all the sub-clusters of (X', Y') that form maximal $g - 1$-clusters. Only when two subsequences with no gaps ($g = 0$) are reached does the algorithm halt.

Since only maximal g-clusters are output, for $g \leq g_{\max}$, Proposition 4.5.1 guarantees that only dominant clusters will be output. Proposition 4.5.2 guarantees that this strategy will identify all dominant g-clusters. Proposition 4.5.3 guarantees that no duplicates will be produced, since a dominant cluster will never be a sub-cluster of more than one maximal cluster.

The main FindDominantClusters function is shown in pseudo-code in Algorithm 3, and the recursive procedure is shown in Algorithm 4. The GetSharedFamilies and SplitIntoRuns functions are not given here, but are implemented identically to the HomologyTeams implementation, except that they ignore genes which have already been assigned orthologs, and treat them as gaps. These functions rely on the key innovation of the HomologyTeams approach: a succinct representation of subproblems that maintains an overall space bound proportional to the size of the genome. Our modified algorithm also uses this representation.

Algorithm 3 FindDominantClusters(A, B, $g_{\max}$)

```
Q ← emptyset
shared_families ← GetSharedFamilies(A, B)
runs_in_A ← SplitIntoRuns(A, shared_families, $g_{\max}$ )
for each A_run in runs_in_A do
    Q ← Q $\cup$ FindDominantClusters'(B, A_run, $g_{\max}$)
end for
return Q
```

Algorithm 4 FindDominantClusters'(A, B, g)

```
Q ← emptyset
shared_families ← GetSharedFamilies(A, B)
runs_in_A ← SplitIntoRuns(A, shared_families, $g$)
if |runs_in_A| $= 1$ then
    $g \leftarrow$ max($\Delta$(A), $\Delta$(B)) $-1$
    Q ← Q $\cup$ (A,B)
end if
if $g \geq 0$ then
    for each A_run in runs_in_A do
        Q ← Q $\cup$ FindDominantClusters(B, A_run, $g$)
    end for
end if
return Q
```

4.5.2 Selecting a Maximum Matching Associated with each Cluster

A maximum associated matching must be selected for each dominant cluster found on line 4 of Algorithm 2. We designed two methods for selecting a maximum matching associated with each cluster. The first relies only on gene order within the cluster, and the second also considers sequence similarity.

The first method uses a simple left-to-right strategy for choosing a local matching of a cluster (X', Y'). Starting with the leftmost gene in the chain X', we match each gene in X' with the leftmost gene in Y', such that the gene is in the same family, and the gene has not yet been matched. For example, given the cluster shown in Figure 4.3, b_1 would be matched with b_2, b'_1 would be matched with b'_2, and e_1 would be matched with e_2. When order is preserved this strategy will perform well. If there has been an inversion or additional scrambling of gene order, this strategy may match genes quite poorly.

The second method uses a greedy strategy to select gene pairs that have similar sequences, *i.e.* with the lowest E-values. Starting with the leftmost gene in the chain X', we match each gene in X' with the most similar gene in Y', such that the gene is in the same family, and the gene has not yet been matched. For example, given the cluster shown in Figure 4.3, b_1 would be matched with b_2, b'_1 would be matched with b'_2, and e_1 would be matched with e'_2.

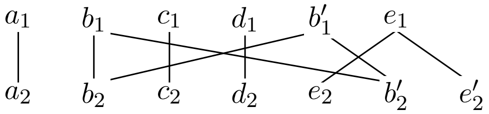


Figure 4.3: An example gene cluster with many possible maximum associated matchings. The e-values are: e-val$(b_1, b_2) = 10^{-100}$, e-val$(b'_1, b'_2) = 10^{-98}$, e-val$(b'_1, b_2) = 10^{-10}$, e-val$(b_1, b'_2) = 10^{-12}$, e-val$(e_1, e_2) = .0001$, e-val$(e_1, e'_2) = 0$.

4.5.3 Computing cluster costs: Estimating the Expected Number of Clusters

A cost must be assigned to each dominant cluster found on line 1 of Algorithm 2. The cost of a cluster depends on the statistical significance of its associated matching. The parameters that we use to determine the significance of a matching are its size h and max-gap g. Note that although all maximal matchings associated with a cluster will have the same size, the max-gap will depend on which matching is selected, which may differ depending on which of the two matching algorithms is used.

More precisely, the cost is based on the number of matchings of size h and max-gap g we expect to observe when comparing two genomes that contain the same genes, in the same gene families, if all possible permutations of genes were equally likely. Let $X_{h,g}$ be a random variable representing the number of clusters with matchings of size h and gap g, in a comparison of two genomes. We define $\phi(h, g) = E[X_{h,g}]$ as the expected value of $X_{h,g}$ under the null hypothesis. The cost $\Phi(h, g)$ of a cluster is then the expected number of clusters with size $\geq h$ and maximum gap $\leq g$:

$$\Phi(h, g) \; = \; \sum_{k=h}^{\nu} \sum_{d=0}^{g} \phi(k, d),$$

where ν, the size of the maximum matching, is the largest possible value of h. For reasonably small values of g, as h increases the probability of observing a cluster decreases rapidly. Thus, for some sufficiently large value of H

$$\Phi(h, g) \; \approx \sum_{k=h}^{H} \sum_{d=0}^{g} \phi(k, d).$$

Thus, rather than summing from $k = h..\nu$, we sum only from $k = h..H$, where H is relatively small, and is set by the user.

For genomes with arbitrary gene family sizes, an exact expression for $E[X_{h,g}]$ is not known. Thus, we propose two methods for estimating the expected number of clusters with a matching of size h and gap g. The first method estimates the number of clusters that would be observed under the null hypothesis through a Monte-Carlo procedure in which random permutations of the genes in each genome are selected at each iteration. With this procedure the number of genes assigned to each family remains the same, but the locations of each family within the genome are randomized. The number of matchings of each size and gap are tabulated at each iteration. This procedure is repeated for r iterations. Let $x_i(h, g)$ be the number of clusters with associated matching of size h and gap g observed in the i^{th} iteration. The average number of clusters observed provides an estimate of the expected number under the null hypothesis:

$$\phi(h, g) \approx \frac{1}{r} \sum_{i=1}^{r} x_i(h, g). \qquad (4.1)$$

This approach will provide very accurate estimates for clusters that occur frequently. However, it will not provide accurate estimates for clusters that have only a very small probability of occurring by chance. In the most extreme case, any cluster that is not observed in any random permutation is assigned a cost of zero. Given two zero cost clusters, this method cannot determine which is more likely to represent the orthologous region.

Our second estimate is based on the upper bound $P_{up}(h, g, n_1, n_2, m)$ derived in Section 2.3. Recall that $P_{up}(h, g, n_1, n_2, m)$ is an upper bound on the probability of observing a maximal max-gap cluster of size h and gap no greater than g, in a comparison of randomly ordered genomes containing n_1 and n_2 genes respectively, and m shared gene families, each of size exactly two. We estimate the expected number of clusters of size h and max-gap *exactly* g as:

$$\phi(h, g) \approx P_{up}(h, g, n_1, n_2, \min(n_1, n_2)) - P_{up}(h, g - 1, n_1, n_2, \min(n_1, n2)). \tag{4.2}$$

This will only be a rough approximation for a number of reasons. It is an upper bound on the probability of observing at least one cluster, rather than the expected number of clusters. In addition, chromosome boundaries are disregarded, which will cause the number of clusters to be slightly overestimated. Most importantly, however, it assumes that all gene families are of size at most two, so it may severely underestimate the number of clusters. Unlike the randomization approach, however, with this analytical method even very small probabilities can be computed.

We experimented with two different strategies for prioritizing clusters. The first strategy ranks clusters according to their cost Φ, which is based solely on the spatial characteristics of the cluster. Often, however, there will be multiple clusters in the queue with the same cost. With the first strategy, these clusters are ranked randomly. Our second strategy first ranks clusters according to their cost Φ, but uses sequence similarity to break ties. Given two clusters of equal cost, we can sort them in the priority queue by their minimum E-value. More precisely, the secondary sorting criterion is the minimum E-value of the associated matching, where the minimum E-value of a matching (X_m, Y_m) is defined to be

$$\min\{\text{e-val}(x, y) \mid x \in X_m, y \in Y_m, F(x) = F(y)\},$$

where e-val(x, y) is the E-value of gene x and gene y, and is computed as described in Section 4.6.2.

This secondary sorting criterion is most important for selecting between clusters with associated matchings of size one, *i.e.* clusters containing only a single gene, that provide no spatial evidence of orthology. In this case, sequence similarity is the only information available.

4.5.4 Updating the queue of dominant clusters

Algorithm 2 starts by computing the set of all dominant g-clusters in the original homology graph $\mathcal{H}$. As orthologs are assigned, however, the homology graph may change, and thus the set of dominant clusters may change. On line 13 of Algorithm 2, genes are assigned as orthologs. When a pair of genes (g_1, g_2) is determined to be an orthologous pair, all other edges to g_1 and g_2 must be pruned from the graph. Thus, after the t^{th} iteration of line 13 of Algorithm 2, there will be a new homology graph $\mathcal{H}_t$. Since $\mathcal{H}_t$ contains fewer edges then $\mathcal{H}_{t-1}$, it may also contain a different set of dominant g-clusters. After removing edges the size of gaps may have increased, and a g-cluster (X, Y) that was dominant at time t may have gap greater than $g_{\max}$ in $\mathcal{H}_{t+1}$. In this case, the cluster (X, Y) is considered invalid at time $t + 1$, and should be removed from the priority queue. In addition, a cluster (X', Y') that was previously dominated by (X, Y) could become a dominant cluster, and thus should be added to the priority queue at time $t + 1$.

We use a lazy strategy to handle these deletions and insertions. We wait until cluster (X, Y) reaches the front of the queue to remove it. We check whether the cluster if invalid, and only then do we insert its newly dominant sub-clusters into the priority queue. Since the function FindDominantClusters treats matched genes as gaps, we can re-use this function to identify the newly dominant sub-clusters of cluster (X, Y), by passing X and Y as the input gene sequences rather than the entire genome.

Even with this lazy insertion strategy, in most cases the lowest-cost valid cluster at time t will in fact be in the queue at time t. This is because if a cluster C' is valid at time t, but is not yet in the queue, it must be dominated by some invalid cluster C that is associated with a graph from a previous time step, but has not yet been removed from the queue. However, the fact that C is still in the queue means that there are clusters with smaller cost. These clusters most likely have a smaller cost than C' as well, since C' is a sub-cluster of C. In fact, if a cluster C has no duplicates, then any sub-cluster C' it dominates *cannot* have a smaller cost.

Theorem 4.5.3. *Let (X, Y) be a g-cluster that contains no duplicates,* i.e. *if $x_1 \in X, x_2 \in X$, and $F(x_1) = F(x_2)$ then $x_1 = x_2$, and if $y_1 \in Y, y_2 \in Y$, and $F(y_1) = F(y_2)$ then $y_1 = y_2$. Any sub-cluster (X', Y') dominated by (X, Y) must have equal or higher cost.*

Proof. Let h and g be the size and max-gap of (X, Y), respectively and h' and g' be the size and max-gap of (X', Y'). Since $X' \subseteq X$ and $Y' \subseteq Y$, $h \geq h'$. Since (X, Y) dominates (X', Y'), $g \leq g'$. By definition,

$$\begin{aligned}
\Phi(h', g') - \Phi(h, g) &= \sum_{k=h'}^{H} \sum_{d=0}^{g'} \phi(k, d) - \sum_{k=h}^{H} \sum_{d=0}^{g} \phi(k, d) \\
&= \sum_{k=h'}^{h-1} \sum_{d=0}^{g'} \phi(k, d) + \sum_{k=h}^{H} \sum_{d=0}^{g'} \phi(k, d) - \sum_{k=h}^{H} \sum_{d=0}^{g} \phi(k, d) \\
&= \sum_{k=h'}^{h-1} \sum_{d=0}^{g'} \phi(k, d) + \sum_{k=h}^{H} \sum_{d=0}^{g} \phi(k, d) + \sum_{k=h}^{H} \sum_{d=g+1}^{g'} \phi(k, d) - \sum_{k=h}^{H} \sum_{d=0}^{g} \phi(k, d) \\
&= \sum_{k=h'}^{h-1} \sum_{d=0}^{g'} \phi(k, d) + \sum_{k=h'}^{H} \sum_{d=g+1}^{g'} \phi(k, d)
\end{aligned} \tag{4.3}$$

Regardless of the method used to estimate $\phi(k, d)$, it is always non-negative. The sum of non-negative terms is non-negative, therefore $\Phi(h', g') \geq \Phi(h, g)$. □

Even when a cluster contains duplicates, it is typically the case that all its sub-clusters have a higher cost. This is because a sub-cluster will generally be smaller, and have a larger gap. In rare circumstances, it is possible that a sub-cluster will have a lower cost. For example, consider this very simple example:

$$\begin{aligned}
G_1 &= a_1 * a_1' * * b_1 \\
G_2 &= b_2 * * a_2.
\end{aligned}$$

The cluster $C = (\{a_1, a_1', b_1\}, \{a_2, b_2\})$ is a dominant 2-cluster. If we select the associated maximum matching $(\{a_1, b_1\}, \{a_2, b_2\})$ then the matching has size $h = 2$ and max-gap $g = 4$. The sub-cluster $C' = (\{a_1', b_1\}, \{a_2, b_2\})$ is also a 2-cluster, but it is contained within C, so it is not dominant. If a_1 is assigned an ortholog, C becomes invalid and C' becomes a dominant cluster. However, the maximum matching associated with C' has size $h = 2$ and max-gap $g = 2$, so it will be assigned a lower cost than C.

This scenario occurs as a result of selecting a poor matching for C. In the majority of cases, however, the cost of a sub-cluster will never be less than the cost of its dominating cluster. Thus, it is reasonable to use a lazy strategy for adding newly dominant clusters to the queue.

4.6 Experiments

Evaluating ortholog prediction methods is challenging. Although there are many databases of predicted orthologs, there is no clear gold standard. A wide variety of evaluation strategies have been used. Methods that do not consider genomic context often use spatial organization to evaluate their predictions [68], but clearly that is not appropriate for a method based on genomic context. Another approach is to use functional genomics data [85], since orthologs are believed to have similar functions. Experimentally determined functions are known for only a small fraction of genes, so more indirect measures must be used, such as expression profiles, protein-protein interactions, and participation in metabolic pathways. Methods that consider spatial organization are often evaluated on synthetic datasets [19, 63], but these datasets are typically generated so as to conform to the method's underlying evolutionary model, which is often not realistic. Ortholog predictions can also be evaluated by comparing gene names and annotations, under the assumption that genes with similar names and annotations are more likely to be orthologs than genes with distinct names and no shared annotations [63, 14, 13]. Obviously, this assumption will hold to varying degrees depending on the genes under consideration, and how they were annotated.

4.6.1 Data and Evaluation Metrics

Our main goal in this evaluation is to test whether by relaxing the conserved block definition and incorporating sequence similarity scores we are able to improve ortholog prediction compared to previous methods that try to maximize spatial conservation. Thus, in order to compare our results with two previous spatial methods, we use the same evaluation approach as LCS [14] and CIGAL [13], which were tested on a γ-proteobacteria dataset. The resulting predictions were evaluated using gene name annotations, as described below.

The dataset consists of eight single-chromosome species (listed in Table 4.2), that span the phylogeny of γ-proteobacteria (Figure 4.5). The MRCA of these species is thought to have lived at least 300 million years ago [98]. Table 4.2 gives the number of genes in each genome. The genome sizes range from only 598 genes, in the aphid endosymbiont *Buchnera aphidocola*, to 5642 genes, in the environmentally versatile, opportunistic pathogen *Pseudomonas aeruginosa*.

For consistency, we used the gene families constructed by Blin *et al.* [14]. They place an edge between two genes in the homology graph if the sequences have at least 25% identity (in both directions), and the BLAST alignment covers at least 65% of both sequences. Then, they take the transitive closure to generate the family graph. Figure 4.4 shows the distribution of the number of genes per family per genome. *Pseudomonas aeruginosa* contains two families with more than forty genes. All other families are represented by fewer than 35 genes in each genome. The majority of families are represented by fewer than ten genes in each genome. These gene families determine the maximum matching size for each pair of genomes (Table 4.2).

We compared our method with three previous methods. In the BBH method, orthologs are assumed to be those gene pairs that form bi-directional best Blast hits. LCS and CIGAL assign orthologs greedily based

		Ec	Hi	Pa	Pm	St	Xf	Yp
		4345	1732	5642	2015	4532	2821	3954
Ba	598	564	466	521	450	563	464	557
Ec	4345		1319	2209	1460	3348	1104	2325
Hi	1732			1131	1329	1320	820	1270
Pa	5642				1220	2231	1222	2035
Pm	2015					1459	829	1422
St	4532						1123	2543
Xf	2821							1072

Table 4.2: Number of genes in each bacterial genome (first row and first column), and maximal matching size ν. Abbreviations: Ba, *Buchnera aphidicola*; Ec, *Escherichia coli*; Hi, *Haemophilus influenzae*; Pa, *Pseudomonas aeruginosa*; Pm, *Pasteurella multocida*; St, *Salmonella typhimurium*; Xf, *Xylella fastidiosa*; Yp, *Yersinia pestis CO_92*.

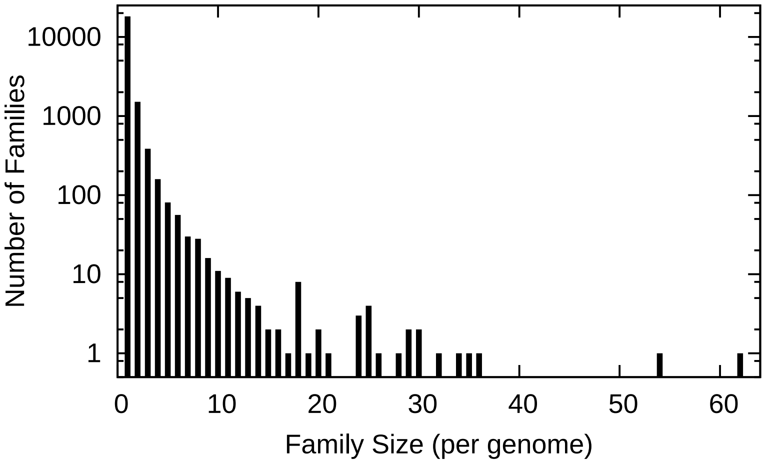


Figure 4.4: The distribution of family sizes, over all eight genomes.

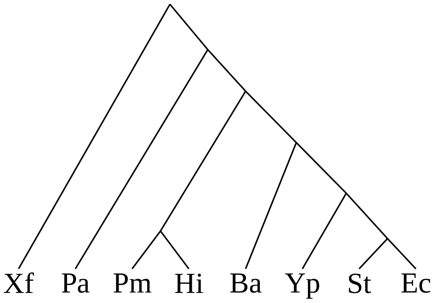


Figure 4.5: Phylogenetic tree showing the estimated branching order of the eight γ-bacteria species used in the evaluation. [98]. Branch lengths are not representative. Abbreviations are given in Table 4.2.

Measure	Formula	Intuitive Meaning
Precision	$\frac{TP}{TP+FP}$	The percentage of predicted orthologs that are correct.
Recall / Sensitivity	$\frac{TP}{TP+FN}$	The percentage of orthologs predicted to be orthologs.
Specificity	$\frac{TN}{TN+FP}$	The percentage of paralogs predicted to be paralogs.
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	The percentage of predictions that are correct.
$F1$ measure	$\frac{2TP}{2TP+FN+FP}$	The harmonic mean of precision and recall.

Table 4.3: Common evaluation metrics for binary classification tasks.

on longest common substrings [14], and largest common intervals [13], respectively. We applied BBH, LCS, CIGAL, and our new algorithm to all 28 pairs of genomes and evaluated the results using a "ground truth" dataset constructed as follow. Each gene is associated with a (possibly empty) list of UniProt [3] names, including the gene name field and the synonyms field. We consider two genes to be true (T) orthologs if they share a name. If both genes have UniProt names, but no common name, we consider them paralogs, or a false ortholog pair (F). Otherwise, if one or both of the genes in the pair has no UniProt name, we consider the pair to be unknown (U). Note that this approach does not guarantee that a gene will be assigned only one ortholog. In fact, a number of genes have two or more matches in a single genome.

We ran each method on all 28 pairs of genomes. Note that with the exception of BBH, these methods are guaranteed to be symmetric. In other words, it is possible that switching the order of the input genomes will yield a slightly different set of orthologs. We always order the genome pairs alphabetically. For each pair of genomes, the output of each method is a matching, a set of predicted orthologous pairs. These are the set of positive predictions (P). All gene pairs that were not matched are considered paralogs, and are labeled negative ortholog predictions (N). Combining the known labels with the predicted labels, each gene pair is classified as a true negative (TN), true positive (TP), true unknown (TU), false negative (FN), false positive (FP), or false unknown (FU).

Table 4.3 summarizes the five metrics typically used in evaluating prediction systems. For orthology prediction, the majority of the examples are negative (*i.e.* paralogs), and thus specificity and accuracy will always be high as long as the classifier does not predict too many positives. For this reason, we selected precision, recall, and the $F1$ measure as our evaluation metrics. For each method we report the precision and recall for all 28 genome pairs, as well as the *average* precision and recall and the *overall* precision and recall. The average[5] precision is simply the average of the 28 precision measurements, whereas the overall precision reports the percentage of all predicted orthologs in all 28 datasets that are correct. The average and overall precision and $F1$ measure are defined similarly. The average precision weights all *genome* pairs equally. The overall precision weights all *gene* pairs equally, and thus is influenced more by pairs of large genomes with many true orthologs than by pairs of small genomes with only a few orthologs.

[5] Often, the average precision is referred to as the *macro-average*, and what we call the overall precision is referred to as the *micro-average*.

4.6.2 Methods

The gene sequences, gene orderings, and UniProt annotations for all eight species were obtained from a website[6] maintained by Cedric Chauve.

We implemented Algorithm 2 and the Monte Carlo method described in Section 4.5.3 in C. The implementation of Algorithm 2 re-uses much of the code from the HomologyTeams[7] software. The analytical cluster probabilities were computed using Mathematica.

The cluster cost $\Phi(h, g)$ was computed separately for each pair of genomes. As described in Section 4.5.3, $\phi(h, g)$, the expected number of clusters with associated matching of size h and max-gap g, was estimated in two ways, by Monte Carlo sampling and using an analytical method. The Monte Carlo sampling procedure was conducted as follows. For each genome, $r = 1,000,000$ random permutations of gene order were generated. For each pair of randomized genomes, all dominant max-gap clusters with $1 \leq h \leq 50$ and $0 \leq g \leq 20$ were identified. The order-based strategy described in Section 4.5.2 was used to select a matching, and the size and max-gap of the associated matching were tabulated, yielding a table of cluster frequencies for all values of g and h. These frequencies were used to estimate $\phi(h, g)$, as specified in Equation 4.1. In the analytical method, $\phi(h, g)$ was computed from Equation 4.2, for all matchings of size $1 \leq h \leq 50$ and max-gap $0 \leq g \leq 20$.

E-values were calculated using an all-against-all BLAST [1] comparison using default parameters on a combined FASTA file with the list of gene sequences from all eight genomes. E-values are not, in general, symmetric because Blast statistics are length dependent. If sequences a and b are of different lengths, e-val(a, b) will differ from e-val(b, a). In this case, we set both e-val(a, b) and e-val(b, a) to be the smaller of the two E-values. These E-values were used to compute BBHs for each pair of genomes G_1 and G_2. Given sequence a in G_1 and b in G_2, the pair (a, b) is a BBH iff there is no pair (a, b') such that b' is in G_2 and e-val$(a, b') \leq$ e-val(a, b), and there is no pair (a', b) such that a' is in G_1 and e-val$(a', b) \leq$ e-val(a, b).

4.6.3 Results

In this section, we compare nine different variants of our method, summarized in Table 4.4. These strategies differ in terms of four factors: the method used to compute cluster costs, the method for selecting an associated matching, whether E-values were used to rank clusters with equal costs, and the value of $g_{\max}$. By comparing different strategies, we investigate the affect of allowing gaps, and the importance of incorporating sequence information along with spatial information. We also compare our methods with three existing methods for ortholog predictions: CIGAL, LCS, and BBHs.

In all the graphs below, the genome pairs are ordered by the value of the $F1$ measure achieved when using BBHs to assign orthologs. In other words, gene pairs on the left are “easy”: orthologs can be identified accurately using gene sequences alone. Gene pairs on the right are “hard”: sequence-based methods have lower precision and recall on these datasets.

[6] Available at `http://arnt.bioinfo.uqam.ca/~genoc/CG06`

[7] Available at `http://euler.slu.edu/~goldwasser/homologyteams/`.

	Computing $\phi(h, g)$	**Selecting a Local Matching**	**Ranking clusters**	$g_{\max}$
MG0	Monte Carlo	Order	Φ	5
MG1	Analytical	Order	Φ	5
MG2	Analytical	Order	Φ	0
MG3	Analytical	Order	Φ	10
MG4	Analytical	E-values	Φ	5
MG5	Analytical	Order	Φ + E-values	5
MG6	Analytical	E-values	Φ + E-values	5
MG7	Analytical	E-values	Φ + E-values	10
MG8	Analytical	E-values	Φ + E-values	0

Table 4.4: Summary of prediction methods evaluated.

Analytical versus Monte Carlo

In Section 4.5.3 we proposed two methods for estimating the significance of a cluster: a Monte Carlo method and an estimate based on the analytical equations presented in Section 2.3. The $F1$ measure for these two methods are compared in Figure 4.6, when $g_{\max} = 5$. Although there are a few datasets for which the Monte Carlo method yields a larger $F1$ measure, the overall performance is slightly better with the analytical estimates. This difference occurs because, even with one million samples, the Monte Carlo method is not able to estimate very small probabilities accurately. Hence, the queue initially contains a large number of clusters that are all assigned a cost of zero. The analytical method can rank these clusters more accurately, giving preference to those with smaller gaps and larger size. In Figure 4.6, E-values were not used to select a local matching, nor to rank clusters. When local matchings are selected based on E-values, the trends are very similar. If E-values are also used to rank clusters, then the difference between the two methods is reduced, since the inability of the Monte Carlo method to rank highly significant clusters is mitigated by the use of E-values to rank these clusters.

Figure 4.6 also illustrates that the average (macro-average) performance is better than the overall (micro-average) performance. This trend is observed regardless of the method used to predict orthologs, since the genome pairs with more orthologs tend to be the more difficult datasets.

Allowing Gaps

A central tenet of our approach is that ortholog prediction can be improved by using a more flexible cluster definition that allows insertions and deletions. We also claim that our statistical approach to scoring clusters enables us to identify more true positives without increasing the number of false positives. In order to test these assumptions, we examined how the performance of our method changes as $g_{\max}$ is increased from 0 to 20. No sequence information is considered in this analysis, because we want to investigate the effect of allowing gaps when using a purely spatial approach. Figure 4.7 shows that as $g_{\max}$ is increased from 0 to 5, both precision and recall increase. As expected, the improvement to recall is larger than to precision. The largest improvements are achieved on the hardest datasets. Although for certain datasets, such as Pm-Yp,

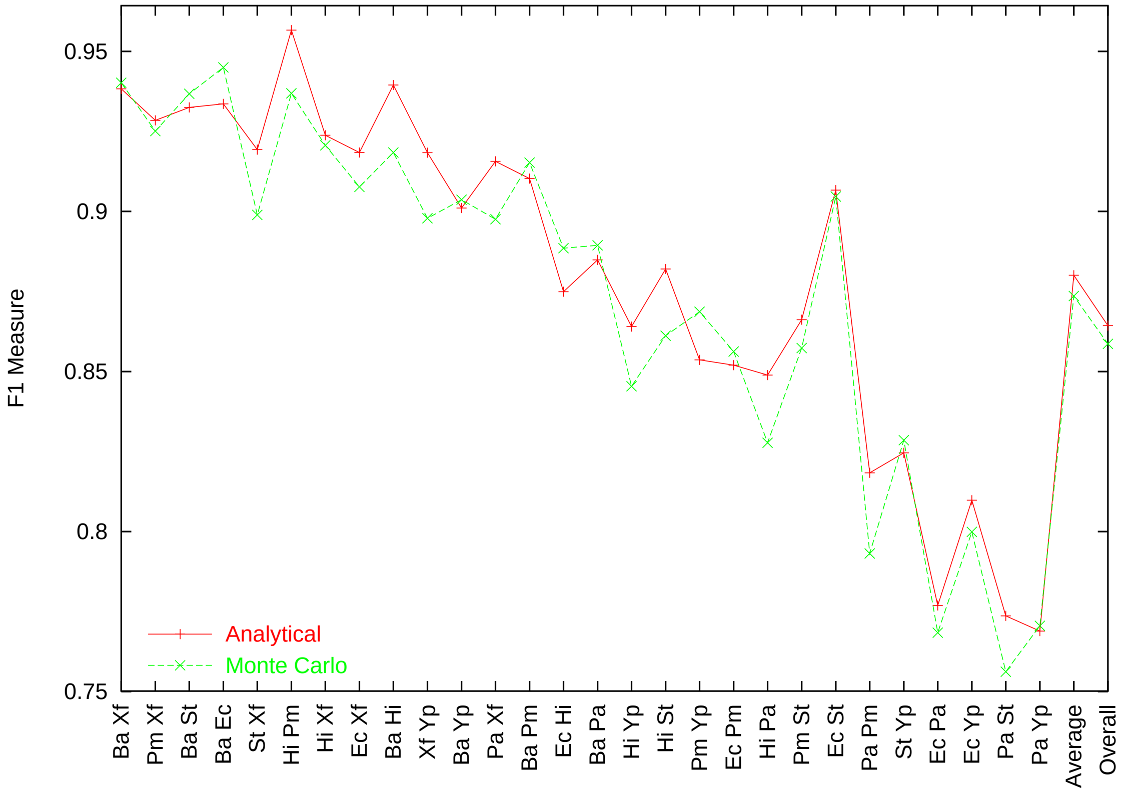


Figure 4.6: Performance comparison of two methods for estimating the significance of a cluster: Monte Carlo (MG0) and Analytical (MG1).

allowing gaps larger than five does improve precision and/or recall, increasing $g_{\max}$ above five does not yield an increase in overall performance. However, neither does it substantially decrease performance. Even allowing gaps as large as ten or twenty, the performance decreases only very slightly or not at all. This shows that our use of cluster statistics is effective in eliminating clusters that are not biologically meaningful.

Figure 4.7 also compares our method to CIGAL. Recall that CIGAL is based on common intervals, which are max-gap clusters with $g = 0$. By allowing gaps in conserved blocks, our method obtains a substantial performance improvement over CIGAL. This difference is due almost entirely to the more liberal cluster definition, since when $g_{\max} = 0$, the overall performance of the two methods, as expected, is very similar. Max-gap with $g_{\max} = 0$ performs slightly better than CIGAL on the easiest datasets, which is probably due to differences in how a local matching is selected. Unlike CIGAL, our matching strategy explicitly tries to preserve gene order, which appears to work better than CIGAL's matching strategy, especially for the easier datasets. This small difference is not sufficient to explain the improvement over CIGAL when $g_{\max} = 5$. Thus, the majority of the improvement must be due to using a more liberal cluster definition.

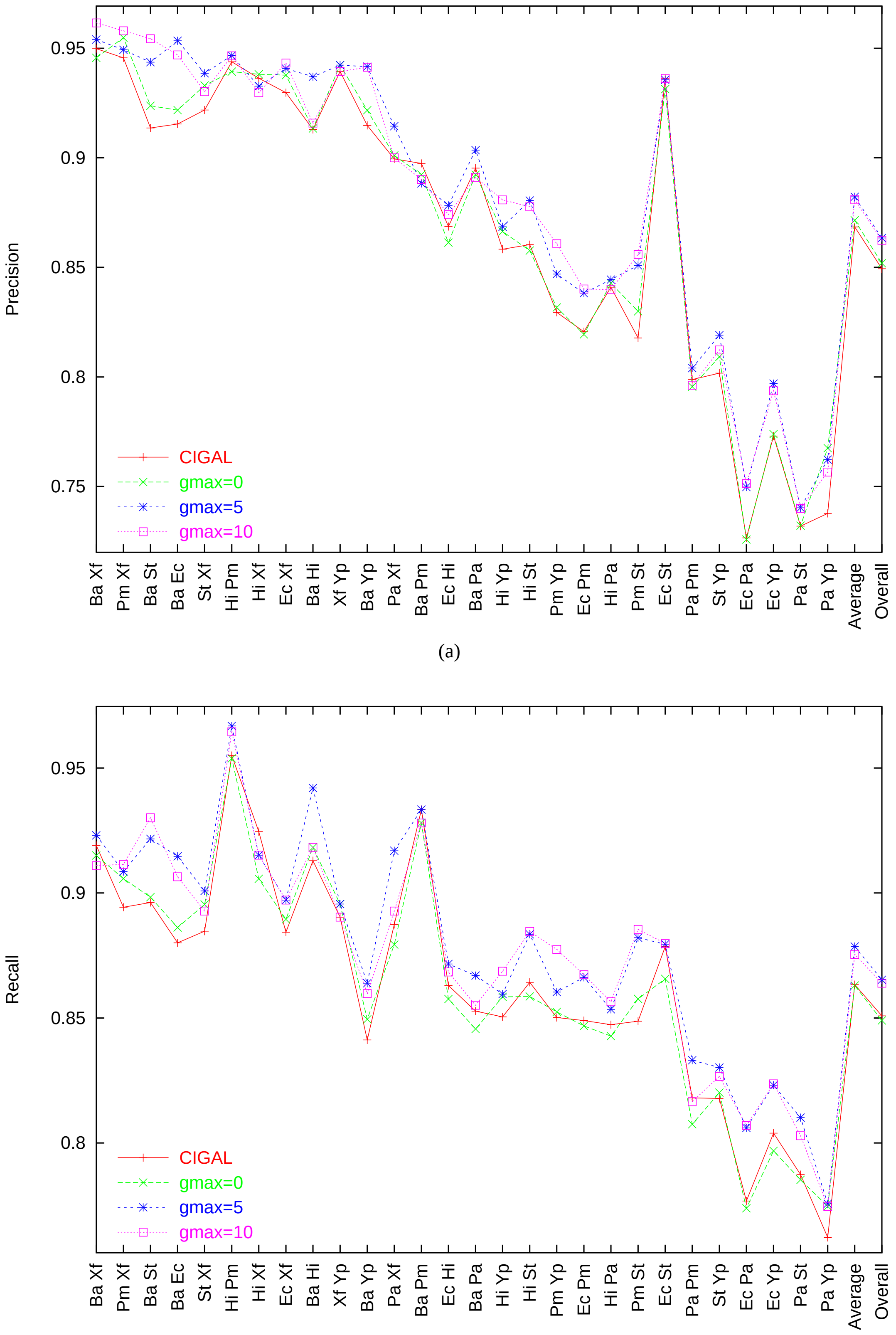
The advantage of a cluster definition that includes gaps is exemplified by the cluster in Figure 4.8, which shows a dot plot of regions in the *E. coli* and *B. aphidocola* genomes. This region contains a gene cluster characterized by numerous insertions and deletions. A method that does not recognize conserved blocks that contain gaps would fail to detect this cluster, and thus would be unlikely to correctly identify orthologs for the genes in these regions.

Considering Gene Order

To evaluate the relative importance of gaps versus gene order, we compare the max gap method, based on spatial information alone and disregarding sequence information, with LCS, a method based on common substrings. For the γ-proteobacteria considered in this evaluation, order tends to be very conserved. Consequently, LCS achieves better performance than CIGAL (Figure 4.9). Not only does CIGAL have lower precision than LCS, but it improves recall for only one of the 28 genome pairs (not shown). This poor performance occurs because only a small fraction of the additional clusters that CIGAL identifies are biologically meaningful—more often they are just chance clusters, and thereby increase the number of false positives. This illustrates that for this dataset, relaxing the gene order constraint was not helpful for ortholog prediction.

If rearrangements *and* gaps are allowed ($g_{\max} = 5$), then our method performs similarly to LCS overall. It performs slightly better on roughly a third of the datasets and slightly worse on the remaining datasets.

For these eight γ-proteobacteria, even when clusters contain gaps, gene order tends to be conserved. For example, the cluster shown in Figure 4.8 contains many gaps, but order is almost perfectly preserved within the cluster—it contains only a single inversion. This suggests that a method that considers both gene order and gaps would yield more accurate predictions. One approach would be to select a cluster definition that requires identical gene order but allows gaps. Such a stringent definition might not work well, however, when analyzing more rearranged genomes. We discuss alternative ways of incorporating gene order into our approach in Section 4.7.1.



(a)

(b)

Figure 4.7: Comparison of (a) precision and (b) recall for CIGAL and Max-gap when $g_{\max} = 0$ (MG2), $g_{\max} = 5$ (MG1), and $g_{\max} = 10$ (MG3).

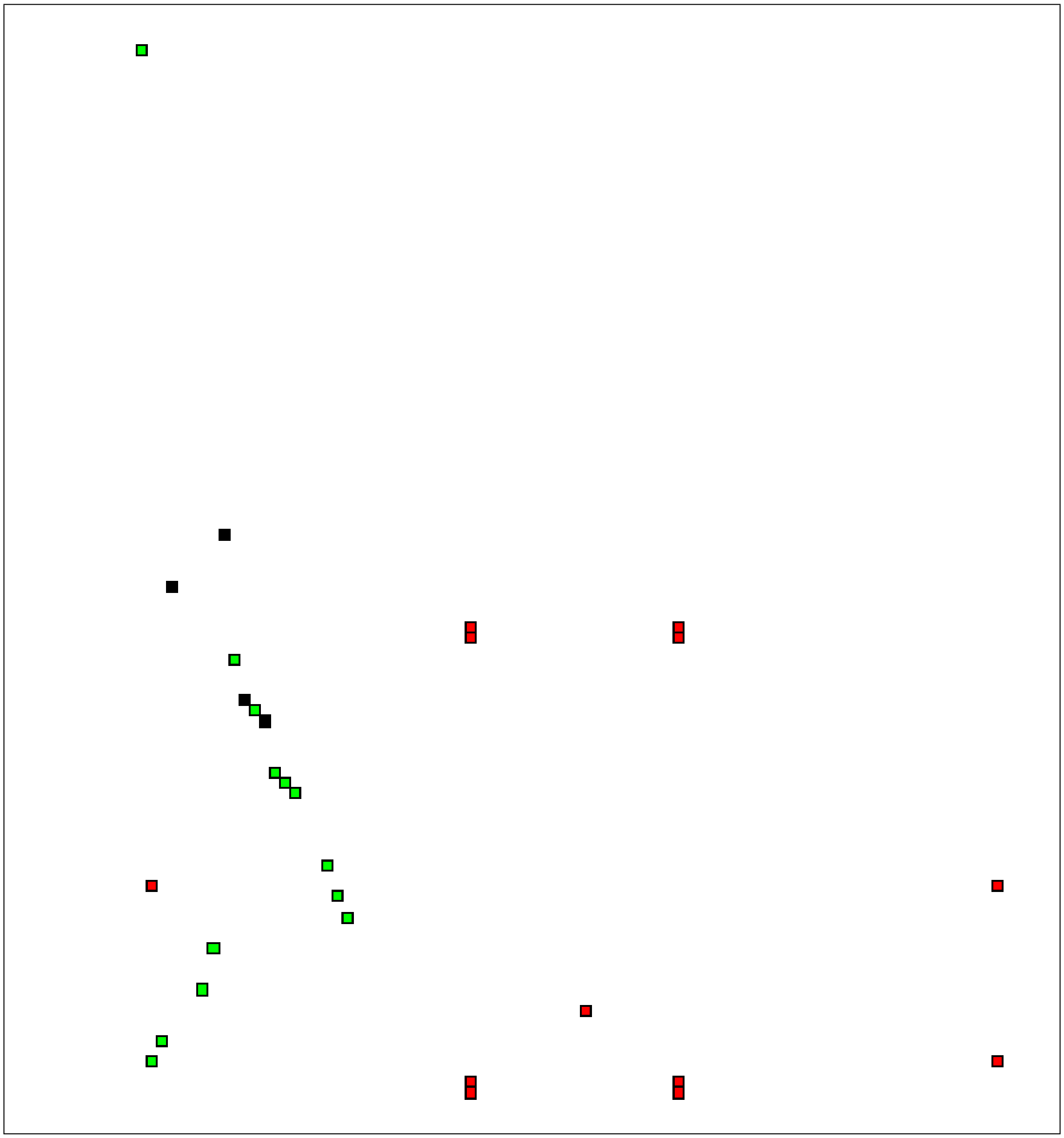


Figure 4.8: A dot plot showing a region of *Buchnera aphidicola* compared with a region of *E. coli*. Each box indicates a pair of genes in the same family, one from Ba and one from Ec. True orthologs are shown in green, false orthologs in red, and unknown pairs in black.

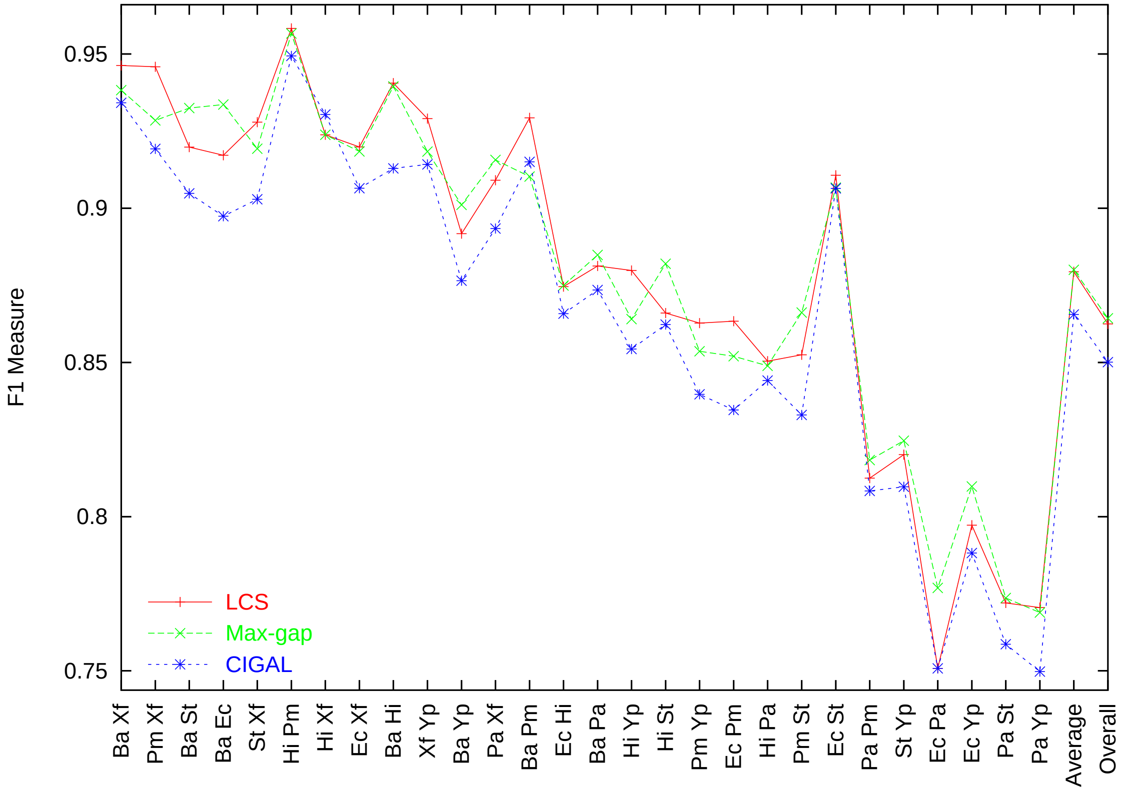


Figure 4.9: Performance comparison of LCS, CIGAL, and Max-gap (MG1).

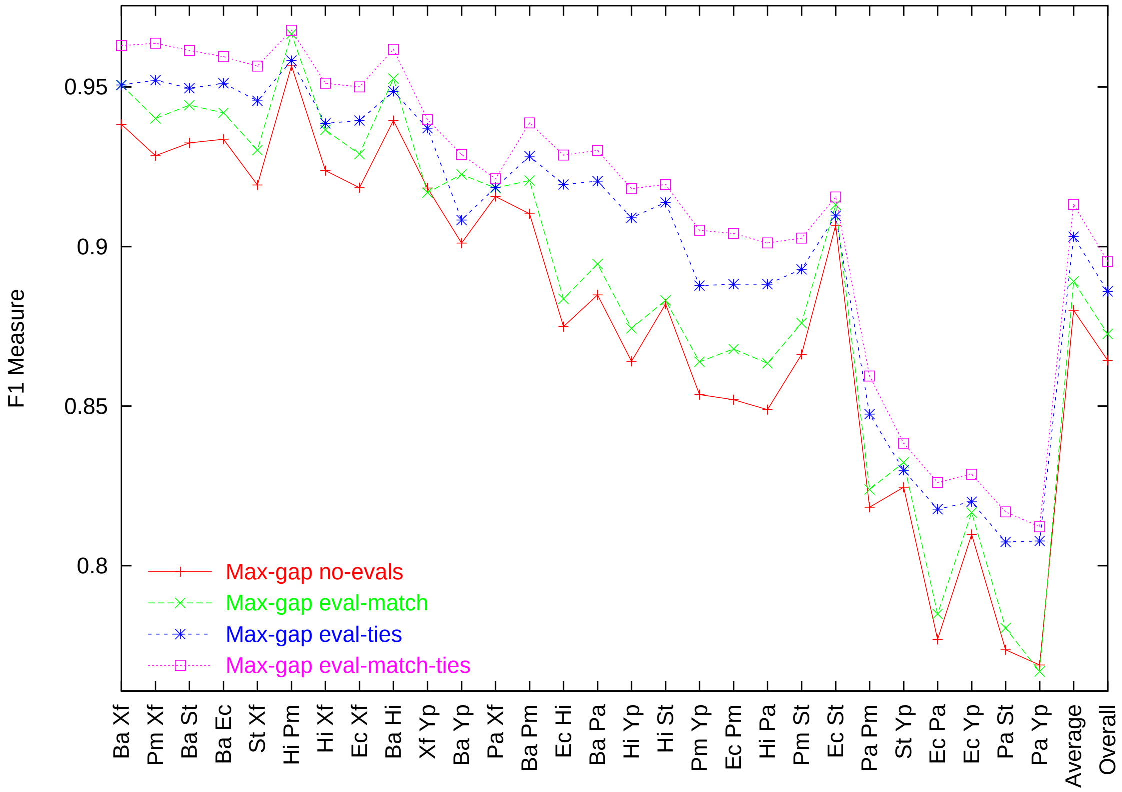


Figure 4.10: Performance comparison of different ways of incorporating sequence similarity. In the first method (Max-gap no-evals, MG1), sequence information is disregarded, and a local matching is selected based on gene order. In the second method (Max-gap eval-match, MG4), a local matching is selected based on E-values. In the third method (Max-gap eval-ties, MG5), a local matching is selected based on gene order; E values are used to rank clusters of equal cost. In the last method (Max-gap eval-match-ties, MG6), E-values are used both to select a local matching and to rank clusters.

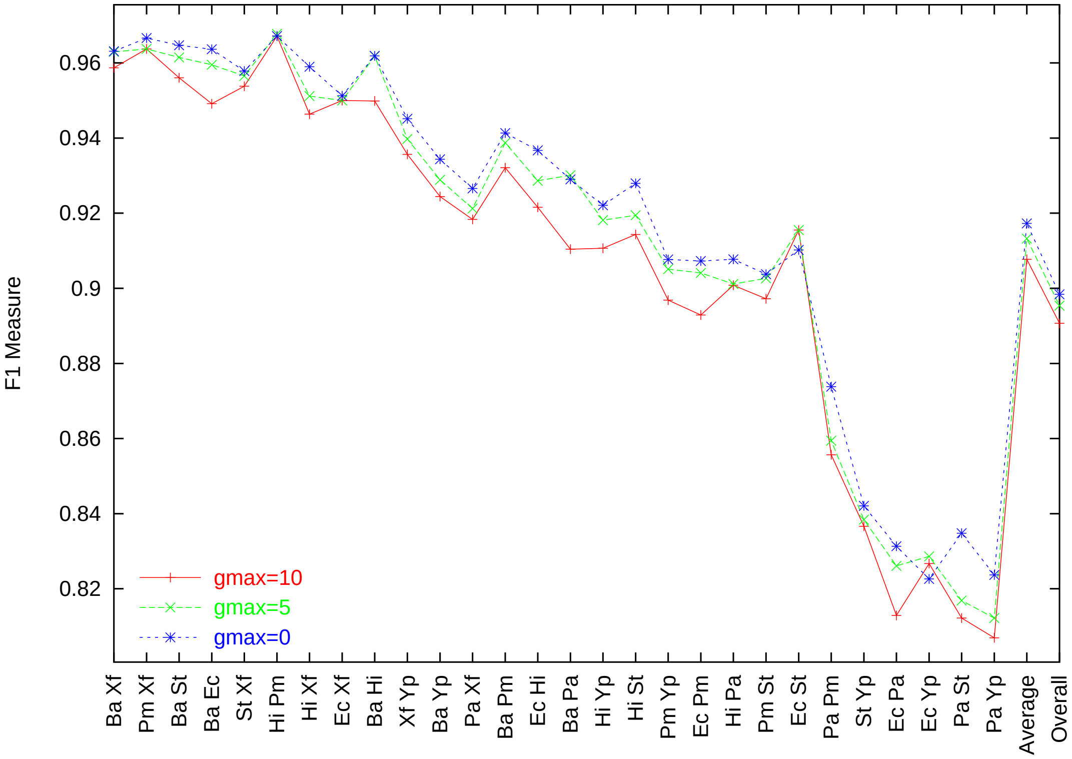


Figure 4.11: Performance comparison using different values of $g_{\max}$, when E-values are used for selecting a matching and ranking clusters: $g_{\max} = 0$ (MG8), $g_{\max} = 5$ (MG6), and $g_{\max} = 10$ (MG7).

Incorporating Sequence Information

We proposed two ways to incorporate sequence information within a spatial framework. In the first case, E-values are used to select a local matching, once a cluster has been obtained from the priority queue. In the second case, E-values are used to break ties in the cluster ranking when more than one cluster has the same cost. Here we evaluate the effectiveness of these two methods. Figure 4.10 shows that our method for using E-values to select a local matching is preferable to our method for selecting a matching based on gene order alone. It is possible, however, that a method that considers only gene order, but recognizes inversions, would perform as well or better than our sequence-based method.

Using E-values to rank clusters with the same cost also results in a large increase in performance (Figure 4.10). This improvement is much larger than that obtained when using E-values only to assign a local matching. This is because the majority of gene clusters have only one associated matching. Even when there is a choice of matching, for this data, gene order is highly conserved within clusters (as illustrated by Figure 4.8), so the order-based matching strategy performs reasonably well. However, there are a number of genome pairs in which there are large numbers of orthologs that do not share any gene neighbors at all. Without using E-values to break ties, these *singletons* are ranked randomly, which is equivalent to picking an arbitrary family member as the ortholog. Not surprisingly, choosing the gene with the most similar sequence as the ortholog yields much better results. These two uses of sequence information are orthogonal, so by using E-values both to select a local matching and to break ties, performance is further increased.

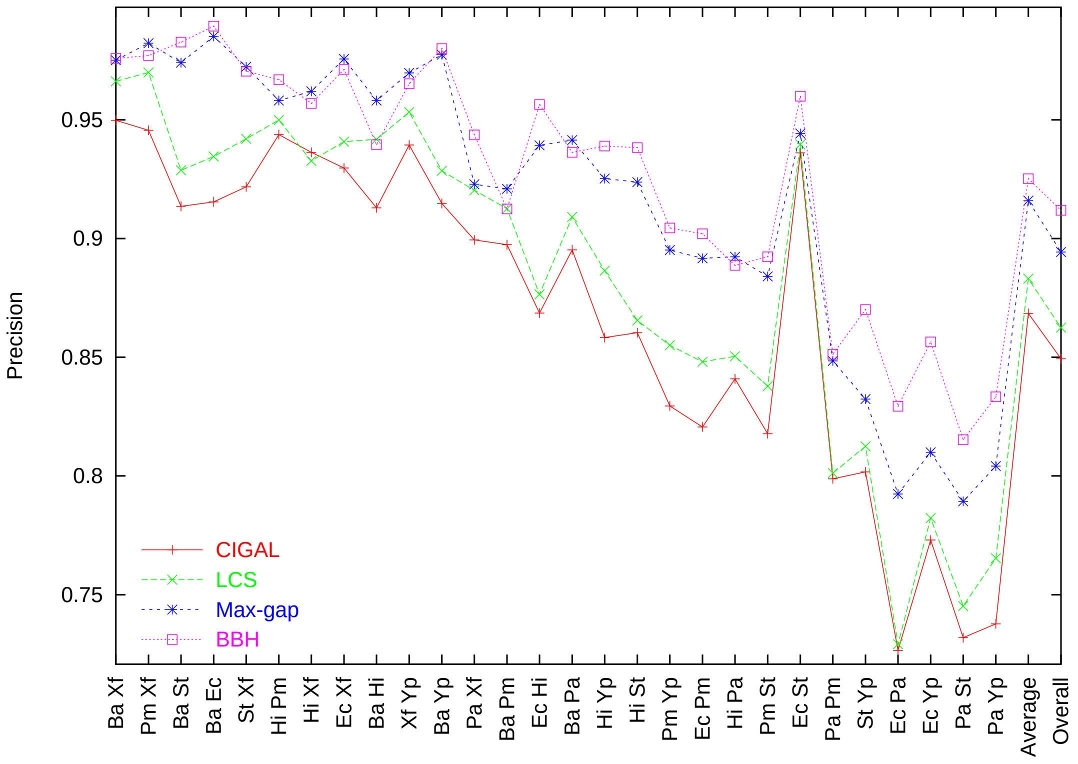
Incorporating sequence into a context-based approach consistently improves performance, over all 28 genome pairs. However, when both methods of incorporating sequence similarity are used, allowing gaps no longer improves performance, as shown in Figure 4.11; in fact, the best performance is obtained when $g_{\max} = 0$. It is not completely clear why larger gap sizes lead to slightly worse predictions. It could be that sequence is just a better predictor than spatial context for this dataset, and so by using a smaller gap we are relying on sequence for a larger portion of the genes. Another possible explanation is that, with larger gaps, there are more “innocent bystanders” that get erroneously pulled into a highly significant, neighboring cluster. When a conserved block is very large and dense, the probability of it occurring by chance is extremely small, and so if there are neighboring genes, even at some distance, they may get included in the cluster, without substantially affecting the probability. A small probability of occurring by chance is a good indicator that a part of the cluster indicates an orthologous block, but it is not a good indicator that the entire cluster represents an orthologous block.

Comparison to Existing Approaches

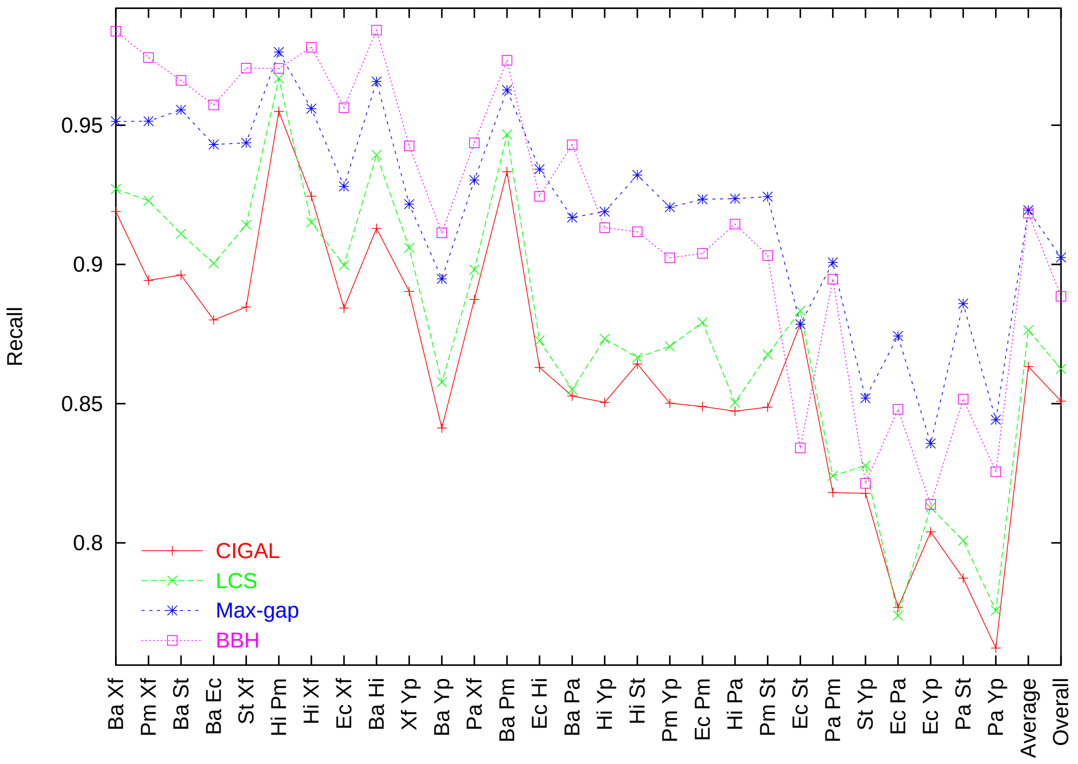
Here we compare our method, based on a combination of sequence and spatial information, with existing approaches based on only spatial context, or only sequence comparison. Figure 4.12 compares our method with CIGAL, LCS, and BBHs. As shown in Figure 4.7, even without using sequence information, our method is a better predictor or orthologs than CIGAL. Allowing gaps and using sequence information, together, results in an even larger improvement over CIGAL (Figure 4.12). Without sequence information, our method performed similarly to LCS, but our combined method achieves substantially higher precision and recall than LCS on all 28 genome pairs. Incorporating sequence into a context-based approach consistently improves performance.

Although BBH is the most common method for assigning orthologs, previous studies of spatial methods have not compared their results with BBHs. We address that omission here, comparing CIGAL, LCS, and our max-gap method with BBHs (Figure 4.12). Surprisingly, both CIGAL and LCS have significantly worse results than BBH. Compared to our method, BBH has higher precision, but slightly lower recall overall. For the easier datasets, BBH tends to do better, particularly on recall. For the harder datasets, however, our method gets consistently higher recall.

The small magnitude of the improvement over BBHs could be due to a number of factors. First, our evaluation metric is based on gene names, which are assigned primarily based on sequence similarity. Indeed, there are many cases of clearly conserved clusters, with identical gene order and content, in which the genes were not assigned the same names. It is highly unlikely that these clusters occurred by chance. Second, our method assumes that all orthologs are assigned to the same gene family. We observed numerous cases where orthologous pairs were assigned to two different families. In these cases, our method can not possibly make the correct prediction. To address this issue, one possibility would be to use a more liberal sequence threshold when identifying homologous genes. However, this strategy would add many extraneous edges to the homology graph. We discuss an alternative solution to this problem in the next section. A third factor may be the close relationship between the γ-bacteria considered in this evaluation. It may be that these eight species are so similar that a simple approach like BBHs works quite well. When comparing more highly diverged species, however, our approach may yield larger improvements. Finally, there are a number of ways to improve our methods, both in how it utilizes spatial and sequence information. These extensions are discussed in the next section.



(a)



(b)

Figure 4.12: A comparison of (a) precision and (b) recall, for BBH, LCS, CIGAL, and max-gap (MG8).

4.7 Discussion and Future Work

In this chapter we presented a new method that predicts orthologs based on a combination of spatial context and sequence information. This method makes two main contributions. The first is an efficient algorithm that, given a bipartite family graph $\mathcal{F}$, identifies all dominant max-gap clusters. As orthologs are assigned, and edges are removed from the graph, our algorithm efficiently updates the set of dominant max-gap clusters. The second contribution is a statistical method that, given two max-gap clusters of different sizes and gaps, estimates which cluster is least likely to have occurred by chance. Our ortholog identification method improves over existing methods based on spatial context, which rely on more conservative cluster definitions, and disregard sequence information. Assessing gene clusters statistically allows us to use a more flexible cluster definition, increasing true positives without increasing false positives. In fact, by identifying conserved blocks that contain gaps, we increase both precision and recall, compared to existing spatial approaches. Furthermore, unlike previous methods, our statistical approach allows us to not only return a set of predicted ortholog pairs, but also to rank those pairs by the strength of the evidence.

On the datasets tested, our combination approach results in slightly lower precision and slightly higher recall than BBHs. However, even with equivalent performance to sequence-based approaches, our approach has the advantage that in addition to identifying orthologous genes, it identifies orthologous regions, which are the required input to many comparative genomics applications.

Aside from ortholog prediction, our framework is useful as a platform for comparing clusters definitions and/or test statistics. Although many different definitions of a conserved block have been used for this problem, it is not yet clear which characteristics of a conserved block are most important. Although it might seem preferable to choose the most liberal possible definition of a conserved block, we demonstrated that this is not necessarily the optimal approach. The appropriate definition of a conserved block will depend closely on the rates and patterns of large-scale chromosomal changes, and may differ from organism to organism. In order to determine which properties are most important, we need an algorithmic framework in which all of these properties can be considered. Since our approach is designed for a very general cluster definition, it can easily be modified to use more constrained definitions: we can set the max-gap to zero (yielding conserved intervals), disregard scrambled clusters and those with duplicates (yielding common substrings), restrict the minimum size of a cluster, or allow only partially scrambled clusters, etc. A Monte Carlo method can be used to estimate cluster significance based on a wide range of test statistics. Hence, our framework is useful for conducting unbiased comparisons of the performance of different cluster definitions and test statistics. For example, we showed that for the γ-proteobacteria considered here, it is more important to consider insertions/deletions then local rearrangements.

Finally, we demonstrated that it is critical to consider sequence information in addition to spatial context. Although spatial context is often useful, there are often large numbers of orthologs that share no neighboring genes. Thus, since LCS and CIGAL disregard sequence information, they do not perform as well as BBHs. It is possible that spatial methods based on other principles might provide more accurate ortholog predictions. On the other hand, it has been demonstrated that CIGAL achieves similar performance on a mouse/human dataset as MSOAR, a state-of-the-art approach that seeks to minimize rearrangement distances [13]. This suggests that other methods based only on spatial data also suffer from the same limitations.

4.7.1 Directions for Future Work

Evaluation

Comparisons of gene names are not ideal for constructing a gold standard. An alternate strategy could be to use a phylogenetic test to assess ortholog predictions. Phylogeny reconstruction is NP complete and hence computationally prohibitive for predicting orthologs in large gene families. In contrast, the use of phylogenetic methods for testing ortholog predictions is less computationally demanding because only a restricted search space must be considered. Since only the phylogenetic position of the predicted orthologs is in question, it is unnecessary to build gene trees for all the genes in a given family. Rather, trees could be constructed for all subsets of four genes (quartets) that include the predicted ortholog pair and two other family members. A prediction is validated if the majority of the quartet trees confirms that the predicted pair is indeed orthologous. A confidence score for each prediction could also be derived from the fraction of quartet trees that support the prediction.

In addition to better evaluation strategies, it would also be interesting to test our method on a more diverse set of species, with more distantly related pairs. A larger, more diverse dataset may help us understand why ortholog prediction is easier in some genomes than in others, and how the characteristics of the genome determine which cluster definition is most appropriate. Finally, we plan to compare the performance of our method with a broader range of competing approaches. In this thesis, the method was compared with two existing methods based on spatial context. Other methods have been developed, but their data and/or code is not publicly available. Creation of a standard, publicly available benchmark will allow more thorough comparison of all the existing methods.

More effective use of sequence information

Our results demonstrate that incorporating sequence information in a spatial approach yields a marked performance improvement. Additional use of sequence information is likely to yield further improvements. For example, our method continues to select clusters based on their spatial characteristics, even when the cost of the cluster is extremely high, *i.e.* there is no evidence that the clusters represent homologous regions. An alternative strategy would be to switch from a combined spatial/sequence approach to a purely sequence-based approach, once the spatial organization no longer provides sufficient evidence to reject the null hypothesis. One possibility is to compare the observed number of clusters with size $\geq h$ and max-gap $\leq g$ with the expected number of clusters. If the difference is small, then the cluster should be discarded. A significance threshold could be selected based on a χ^2 test with one degree of freedom. Only clusters with scores above the significance threshold would be considered for ortholog assignment. To assign the remaining unmatched genes, a purely sequence-based method could be used. Alternatively, when a high precision dataset is desired, no additional orthologs could be assigned.

Sequence information could be better utilized in our our method for choosing an associated matching based on E-values. Currently a greedy heuristic is used. Other possibilities include identifying the maximum weight, maximum cardinality matching: *i.e.* the maximum matching such that the sum of the edge weights is largest. (In this case it is appropriate to use bit scores rather than E-values.) Alternatively, we could select a stable matching. Even better than either of these approaches would be to design an algorithm that considers both gene order and sequence similarity. An algorithm such as Shuffle-Lagan [23] could be used to create a *glocal* alignment of the gene sequences in the cluster: an alignment in which each letter of one sequence is aligned to only one letter of the other sequence, but which allows for rearrangement events such as inversions, translocations, and duplications.

Another modification that would allow us to make better use of sequence data is to omit the transitive closure step when creating the input graph. As discussed in Section 4.3, for ortholog identification, it is not necessary to identify all homologs of a gene, but only a small set that is likely to contain the true orthologs. Requiring gene families defined as equivalence classes is actually detrimental, since to create families we have to either remove strong edges between genes in $\mathcal{H}$, or add weak edges. Removing strong edges may remove orthologs, whereas adding weak edges mostly adds noise. Instead, our method could be modified to work on the homology graph $\mathcal{H}$, rather than the family graph $\mathcal{F}$. This introduces some algorithmic challenges, but they are not insurmountable. My approach is strongly based on the fact that the the homology relation is reflexive (*i.e.* the graph is undirected), but only a few details of the algorithm require transitivity.

More effective use of spatial information

Estimating probabilities is currently the most time-consuming step of our algorithm. Even with a million random iterations, the Monte Carlo method does not estimate small probabilities accurately, and the analytical approach is only approximate since it does not consider the effect of gene families. Faster, more accurate statistics could be obtained by combining the two approaches. The analytical equations could be used to generate a biased distribution of permutations for importance sampling [25]. Although the sample space of all possible gene permutations is very large, only a small fraction of random samples will contain non-trivial gene clusters. Our combinatorial analysis can be used to devise a sampling strategy that selects samples only from the small fraction of permutations for which the probability of a cluster is high.

Another way to use spatial information in the absence of sequence similarity would be to add a post-processing step in which pairs of genes with weak or even no detectable similarity are assigned as orthologs if they appear in a gene cluster, and no other potential ortholog was identified for either gene.

One of the strengths of the max-gap cluster definition is it allows homologous blocks that have sustained local rearrangement to be identified. Nonetheless, homologous blocks tend to be more ordered than gene clusters found in randomly ordered genomes. Requiring identical gene order is too strict, and designing an algorithm to find only partially disordered clusters is challenging. A simpler way to consider order, while still allowing rearrangements, is to incorporate the degree of rearrangement into the test statistic. Several such test statistics have been proposed [144]. An order-based test statistic could be included as a secondary sorting criteria, to rank clusters with identical size and max-gap. Alternatively, a compound test statistic could be designed that considers size, max-gap, and order simultaneously.

Chapter 5

Discussion

In this thesis, I provide statistical tests to assess the significance of gene clusters for a variety of biological questions and search scenarios. I developed the first formal statistical framework for max-gap gene clusters [80], the most widely used cluster definition in genomic analyses. This framework provides statistical tests for two common search scenarios: a reference set scenario in which the goal is to find clusters comprising a set of genes of interest, and a whole genome comparison to identify homologous segments. In addition to assessing significance of gene clusters after they are detected by a search algorithm, this framework facilitates principled selection of parameter values prior to conducting a search for gene clusters.

In the development of statistical tests for the max-gap cluster definition, I observed two troubling issues regarding the use of this definition. First, my statistical results demonstrate that cluster probabilities under the null hypothesis are not monotonic with respect to cluster size, which is commonly used as a test statistic for gene clusters. Although there is a widespread belief that cluster significance grows with the number of homologs in the cluster, it is critical to recognize that for some cluster definitions, larger clusters do not always imply greater significance. In the design of future studies, before selecting a test statistic its distribution under the null hypothesis should be analyzed to ensure that the distribution is monotonic. Second, I observed that the majority of studies based on the max-gap definition use a greedy, bottom-up search strategy that implicitly enforces order constraints, yet these biases are rarely recognized. The use of such heuristics can be particularly dangerous when attempting to draw conclusions about the degree of disorder observed in homologous regions [79].

I also proposed a novel statistical framework for evaluating the significance of clusters spanning three genomic regions, based on an r-window cluster definition and a window sampling search scenario. I designed statistical tests for clusters spanning exactly three regions [133] based on genome models for two typical comparative genomics problems: analysis of conserved linkage within multiple species and identification of large-scale duplications. My statistical tests for three genomic regions are the first to combine evidence from genes shared among all three regions and genes shared between pairs of regions. My results demonstrate that these tests are more sensitive than existing pairwise methods, and have the potential to detect more diverged homologous regions. Recent studies of whole genome duplication have compared a duplicated genome with a related genome that diverged prior to the duplication event. This approach has been shown to detect more paralogous regions than can be identified through genome self-comparison. However, my statistical analysis demonstrates that there may be many additional duplicated blocks that these studies are failing to detect, due to their reliance on pairwise tests. The promise of increased statistical power is intriguing in light of the continuing debate concerning the history and tempo of whole genome duplications in the evolution of species such as human and *Arabidopsis*.

Finally, I demonstrated the importance of statistical analysis of gene clusters by applying my max-gap cluster statistics to a key problem in comparative genomics: ortholog prediction. I developed a new method for ortholog prediction, based on a simple greedy strategy which repeatedly selects the most significant max-gap gene cluster, and assigns orthologs within the cluster. The fundamental idea of this approach is to rank clusters based on statistical significance; this strategy was key to applying this greedy strategy to the max-gap cluster definition. Another important innovation was the design of an efficient algorithm for finding all highly significant max-gap clusters, for *all* values of g. This algorithm extends on previous work that finds only maximal max-gap clusters for one particular choice of g, and hence could miss many highly significant clusters.

My method for otholog prediction improves over other methods based on conserved spatial organization, by allowing a more flexible cluster definition to be used, by employing a more principled ranking criterion, and by relying on sequence information in the absence of any significant spatial signal. In addition, rather than just returning a binary classification of each gene pair as an ortholog or a paralog, the statistical approach makes it possible to assign a confidence score to each pair based on the strength of the associated spatial evidence. Lastly, by disentangling the ranking criterion from the cluster definition, my statistical approach to ranking clusters allows the same basic framework to be applied to an unlimited range of cluster definitions and test statistics, making it an effective framework for comparing the performance of different algorithmic and statistical approaches to detecting homologous chromosomal regions.

5.1 Designing Improved Gene Cluster Definitions

In addition to developing new statistical and algorithmic tools for key problems in spatial comparative genomics, this thesis has led to a number of observations about the current challenges in analyzing the spatial organization of genomes, as well as insights into the most promising directions for new methods in spatial comparative genomics.

Identification of distantly related homologous chromosomal regions has traditionally been broken down into two independent steps. The first is to define the spatial patterns suggestive of common ancestry, then search for “gene clusters,” pairs of regions that exhibit these patterns. The second step is to select a test statistic and design a statistical test to determine the significance of an observed cluster. Ideally a cluster definition would be based on all properties of interest, and search parameters would be selected to ensure that only significant clusters are identified. In practice, a cluster definition often constrains only one property, such as the maximum gap size or cluster length. A significance test, based on an orthogonal property such as cluster size or density, filters the clusters identified by the algorithm to ensure that they are statistically significant. Both steps are critical for ensuring sensitive detection of ancient homologous regions without inclusion of false positives.

Formal characterization of a gene cluster is one of the most challenging tasks in cluster identification. Many definitions have been proposed, but there is little understanding of the trade-offs between them, or consensus on which criteria best reflect biologically important features of gene clusters. Nor has any consensus been reached about how to compare or evaluate different gene cluster definitions. This lack of consensus reflects the difficulty in characterizing what homologous blocks will look like, since in most cases evolutionary histories are not known. Most often, when designing cluster definitions this issue is ignored altogether. Formal definitions of gene clusters are typically geared toward the design of efficient search algorithms, rather than on selecting a definition that reflects the underlying biological processes.

Even when the explicit goal is to select a definition that reflects the underlying biological processes, definitions are generally based upon intuitive notions, often derived from small, well-studied examples (*e.g.* such as the MHC region [55, 154, 167]). However, these regions were identified precisely because of their distinctiveness, and so they may not be appropriate representatives of typical homologous regions. Inferences drawn from larger sets of predicted homologous regions may be biased as well, since only those regions that match existing cluster definitions are detected. Confusing the picture still further, inferences about properties of homologous blocks may be unreliable, due to implicit constraints enforced by search algorithms, as described in Section 2.3.3.

Cluster definitions should reflect the patterns of spatial conservation in the data, but these patterns, in turn, will depend on which rearrangement processes dominate in the lineage of interest. The most common large-scale rearrangement events are inversions, translocations, horizontal gene transfer, duplications, and loss. All of these processes will result in different characteristic patterns behind in the genome. In order to design appropriate cluster definitions, it is important to understand not only which rearrangement processes occur, but how often they occur, and how they influence cluster properties.

Inversions can arise as a result of recombination between inverted repeats, and are seen frequently in both eukaryotic and prokaryotic genomes. In fact, inversions appear to be the most frequent rearrangement events in closely related bacteria [9, 84]. The size and spatial distribution of inversions will affect both cluster size and order. If inversions span many genes, and are located randomly throughout the genome, then although the global organization of two genomes may look very different, gene order will be well-conserved within homologous blocks. If inversions are short, on the other hand, conserved regions will be quite small, and gene order in homologous regions may differ substantially. In bacteria, inversions occur most often in a symmetric fashion around the origin or terminus of replication [84]. As a results, genes located together in the ancestral genome will tend to maintain similar distances to the origin and terminus, but may appear on opposite sides of the genome. For this situation, it may be most appropriate to use a cluster definition in which the location relative to the axis of replication is considered, but the absolute genomic location is not. However, note that this pattern is not predictive of orthology per se. Paralogs that arose through tandem duplication could be separated by an inversion, and thus also end up on opposite sides of the origin of replication.

The rate of inversion depends on genome characteristics such as the number of repeats, as well as characteristics of a species' lifestyle, such as level of selective pressure and effective population size. For example, *Saccharomyces* "sensu stricto" species exhibit protein divergence levels similar to mammals, and yet only a few large inversions have been identified within this group, compared to much higher numbers in mammalian genomes [58]. Even within yeasts, inversion rates vary substantially. *A. gosyppi* and *K. lactis* have fewer inversions, and smaller inversions than *S. cerevisiae* and *C. glabrata*, which are more closely related. *D. hanseii*, has been shown to have an inversion rate more than twice as high as that of related yeasts, whereas the rate in *Y. lipolytica* is at least twice as small [58]. In bacteria, there is some evidence that short inversions are generally more common than longer ones [139], but for the most part, the length distribution of inversions in different lineages is unknown.

Multi-gene insertions can occur as a results of translocations and—in bacteria— horizontal gene transfer (HGT). Both translocations and HGT can lead to rearrangement and fragmentation of clusters within the genome, but neither will cause substantial shuffling of gene order within clusters. However, HGT in particular may confound ortholog identification: clusters inserted by HGT may be more conserved than orthologous clusters, and could lead to errors predicting orthologs. Like inversion rates, translocation rates may be affected by repeat frequency. Translocations can occur when direct repeats lead to deletions, and these deleted fragments are reinserted at another location in the genome. Rates of HGT also differ between species [121]. Some of these differences have been attributed to selection against disruption of short sequences used by the cell for orientation purposes during processes like replication and segregation [78].

The mechanisms and rates of gene duplication and loss will also influence the characteristics of homologous blocks [49]. Gene duplication can occur by retrotransposition, tandem duplication, segmental duplication, and whole genome duplication. The characteristics of gene clusters will depend on which duplication mechanisms dominate in the genomes of interest. Whole genome duplication, for example, is often followed by massive gene loss, and thus results in clusters with large numbers of gaps, but often highly conserved gene order [89, 93, 146]. The effect of gene loss on cluster properties will depend on whether genes are lost gradually, one at a time, or abruptly, in large blocks. If gene loss occurs in large contiguous blocks, such as might occur following whole genome duplication, or a lifestyle change from a free-living organism to a symbiont, then the retained genes will occur in large, dense conserved blocks. If single genes are lost independently, on the other hand, conserved regions may still be large, but not very dense.

It has been shown that both duplication and loss rates vary substantially between species and over evolutionary time [104, 105, 108]. Tandem duplication rates may be affected by the number of repeated elements, since recombination between direct repeats can lead to tandem duplications [84]. The little that is known regarding susceptibility to whole genome duplications, on the other hand, suggests that it is related more to species lifestyle than genome characteristics [106, 107, 155].

Functional constraints could also influence the local rate of rearrangement, and thus the local characteristics of a conserved block. If two genes are in the same operon, then there will be selection against insertions or inversions with endpoints between the genes. Consequently, the genes will maintain the same orientation, and the physical distance between them will be constrained. Hence, gene orientation and physical distances between genes may be very informative for identifying functional clusters in bacterial genomes.

In addition to theories about which genomic and lifestyle factors affect specific types of rearrangements, a few hypotheses have been proposed concerning the factors that results in high or low overall levels of rearrangements. For example, symbiotic or pathogenic species often have high rearrangement rates [58, 84]. This has been attributed to a number of factors, including smaller population sizes, and selective pressure to escape immune recognition. However, these associations tend to be either speculative, or weak.

In summary, little is currently known about the rates at which different evolutionary processes occur. The relative frequency of these processes and the degree to which these frequencies are consistent across lineages, remain open questions. Thus, we cannot yet carry out accurate simulations to investigate what gene clusters would look like under characteristic rearrangement regimes. This does not mean it is impossible to compare the performance of two potential cluster definitions on a particular dataset, however.

I argue that a cluster definition should be selected that is precisely as general as needed to include the set of homologous blocks, but no more general, in order to capture as few chance clusters as possible [79]. Of course, we do not know which are the homologous regions. In the future, we may be able to construct accurate generative models, then use these models to evaluate the discriminatory power of different cluster definitions and statistical tests. An innovation we can implement immediately is to select a cluster definition that maximizes the difference between the number of clusters observed in the genomic data of interest, compared to random data, as discussed in Section 2.3.3.

It is essential that new cluster definitions be designed specifically to discriminate truly homologous regions from background noise (clusters of genes that occur by chance). This requires statistical techniques for quantifying the discriminatory power of different combinations of definitions and test statistics, as well as software tools that, given a dataset of interest, and a suite of possible cluster definitions, selects the most appropriate one. In Appendix B, I present a detailed catalog of cluster properties that can be considered in designing new definitions. Analyses of desirable cluster properties may pave the way for new, possibly more powerful cluster definitions.

5.2 Open Problems

In addition to the open problems discussed in previous chapters, and the need for improved gene cluster definitions, my thesis raises a number of other important problems:

Multi-region clusters: Additional statistical tests for comparison of multiple regions is an important area for future work. Tests for more than three regions are needed, as well as tests for whole genome comparison. Such tests will be particularly useful for detecting evidence of more than one round of WGD, and for designing ortholog prediction methods that consider spatial context in more than two genomes.

Combining sequence, spatial, and phylogenetic evidence for ortholog detection: Ortholog detection based on spatial data is a hot topic, that has received considerable attention in recent years. Most existing methods either assume very conservative cluster definitions, or ignore sequence information entirely. However, spatial approaches have limitations. Ideally, methods would be developed to effectively exploit spatial information while at the same time making optimal use of sequence and phylogenetic data as well. Sequence similarities and spatial context could be analyzed simultaneously within a combined statistical framework. This problem seems to fit naturally within an expectation maximization framework, since if the orthologous blocks were known the orthologous genes could be identified, and vice versa.

Gene families: Exact cluster statistics that take gene families into account remains an important and challenging problem. Virtually all genomic data sets require models that consider many-to-many homology relationships. The model upon which I based my statistical tests in Chapter 2 assumes that each gene has at most one homolog. In Chapter 3 this assumption was relaxed slightly to allow for two copies of a gene that was duplicated via WGD. In Chapter 4, arbitrary sized gene families were assumed, but I approximated the probability of gene clusters in this case by assuming a one-to-one homology mapping, and then adjusting the number of homologous gene pairs upward. This approximation worked as well for ortholog prediction as estimating probabilities using a Monte Carlo approach. Even better estimates may be obtained by an approach combining analytical and Monte Carlo methods, as described in Section 4.7.1.

Statistics for clusters found by whole genome comparison: There are a number of unresolved statistical questions regarding evaluating the significance of clusters identified through whole genome comparison. Whole genome comparisons can lead to questions about the degree of clustering in the genome overall, or about individual clusters. For example, a researcher might want to make a global statement about processes in the evolution of the genomes, such as whether an ancestral genome underwent a whole genome duplication. In this case the focus is not on a single homologous region, but on the level of clustering overall. In other cases, we may want to ascribe meaning to individual clusters, *e.g.* to argue that a particular cluster was a result of a whole genome duplication. In this search scenario, clusters are not independent. The presence of one cluster affects the probability of finding additional clusters. The gaps in a cluster will typically be smaller than the expected gap size, and so the expected gap size of the remaining gaps will be larger than expected in a random genome. For example, if a large conserved operon is detected, then to evaluate the degree of clustering of the next largest cluster we might need to take the existence of the first cluster into account, since it will effectively reduce the size of the genome and change the distribution of the test statistics. With many large clusters, the probability of finding small clusters might be changed significantly. Thus, what can be said about the significance of any individual cluster identified through whole genome comparison is unclear.

Statistical tests for selective pressure on spatial organization: In this thesis I attribute similarities in spatial organization of genes to common ancestry, either through speciation or duplication events. If our goal is merely to detect homologous regions, we need not consider why the regions are conserved, or why some regions are more conserved than others. However, clustering of genes may indicate more than recent shared ancestry. Conservation of spatial organization across large phylogenetic distances often indicates selective pressure on gene order, especially in bacteria. With increasing evolutionary divergence, ongoing rearrangement processes lead to randomization of gene order in the absence of functional selection. In distantly related genomes, conservation of genomic organization suggests functional selection, while in more closely related species similarities in gene order may be due only to shared ancestry. Thus, to identify functional selection on spatial organization, the phylogenetic distance between the species must be incorporated into the null hypothesis. One possible direction would be to take an approach analogous to the approach that is used to detect selective pressure at individual sites along specific lineages [182]. Sequence data could be used to infer a phylogenetic tree topology. With the topology fixed, branch-specific rearrangements rates could be learned that maximize the overall likelihood of the data. This likelihood could be compared to that achieved when allowing rearrangement rates to vary vary at different spatial portions of the genome. If the latter likelihood is significantly higher, then selective pressure on these regions can be inferred. The main challenge would be to devise a statistical model that allows efficient computation of the likelihood of observing a particular spatial organization given the inferred rearrangement rates.

Identifying precise boundaries of homologous regions: The statistical tests presented here reject the null hypothesis of random gene order if there is any evidence of shared ancestry in the regions being compared. In the three-window tests in Chapter 3, this means that a cluster may be significant even if only two of the three regions share a common ancestor, or if two regions share non-overlapping regions of homology with the third. In Chapter 4, we observed that highly significant gene clusters may attract spurious neighboring genes by chance. Large gene families exacerbate this problem, since if a cluster is large, there is a good chance that there will be two genes from the same large family in proximity to the cluster in both genomes. More work must be done to identify such “innocent bystanders.” Given the gene family distribution, and the size and length of a gene cluster, it may be possible to estimate the number of unrelated genes that will be near the cluster in both genomes simply by chance. Then the size of the cluster could be corrected before evaluating it statistically. Alternatively, we might be able to identify outliers by comparing density in the periphery of the cluster with density in the center of the cluster.

Statistical tests that consider cluster density and order: The results in this thesis have shown that for many datasets the max-gap definition is too liberal since gene order is not considered. In Section 2.3.3, in an empirical study of three genomic datasets, I demonstrated that the majority of max-gap gene clusters are nested. In Section 4.6.3, I showed that for identifying orthologs in a set of γ-proteobacteria, a cluster definition that requires identical gene order performed better than a definition that allows rearrangements, but not gaps. It is likely that a definition that allows small differences in gene order would perform even better. Rather than trying to define a search algorithm to find only partially ordered clusters, order could be considered in a test statistic. How to choose such a test, and how to combine it with tests of density, is unclear, however. A first step in this direction has been taken by Sankoff *et al.* [144], who proposed a number of quantitative measures of gene order. However, analyses comparing the discriminative power of these measures in genomic data have not yet been carried out. How to best quantify the degree to which order is conserved remains an open question.

Appendix A

Glossary

Conserved: Derived from a common ancestor and retained in contemporary related species. Conserved features may or may not be under selection.

Chromosome: a single DNA molecule. Typically, bacterial chromosomes are circular, while erotic chromosomes are linear.

Gene: a unit of inheritance that consists of a segment of DNA that, typically, encodes a protein or structural or functional RNA. Alternately spliced genes can encode more than one product.

Gene orientation: gene orientation is dictated by the strand from which the gene is transcribed. Genes in a cluster have the same orientation if they are transcribed from the same strand.

Genome: The total genetic material of an individual or species, consisting of one or more chromosomes.

Homologs: Genes or features that share common ancestry.
Homologous: Related through common ancestry.
Homology: Similarity due to shared ancestry.

MRCA: Most recent common ancestor.

Negative selection: The removal of deleterious mutations from a population; also referred to as purifying selection.

Orthologs: Homologs that arose through speciation. They are descendants of the same gene in their most recent common ancestor.

Paralogs: Homologs that arose through duplication.

Phylogenetic distances: Measures of the degree of separation between two organisms or their genomes, expressed in various terms such as number of accumulated sequences changes, number of years, or number of generations.

Positive selection: The retention of mutations that benefit an organism; also referred to as Darwinian selection.

WGD: whole genome duplication.

Appendix B

Catalog of Cluster Properties

The properties underlying existing cluster definitions are generally not stated, and the dimensions along which they differ have been analyzed in only a cursory manner. As a result, the formal trade-offs between different models have been difficult to understand or compare in a rigorous way. Here we attempt to characterize desirable properties of clusters and cluster definitions, in order to develop a more rigorous understanding of how modeling choices determine the types of clusters we are able to find, and how such choices influence the statistical power of tests of segmental homology. We present a set of properties upon which many existing gene cluster definitions, algorithms, and statistical tests are explicitly or implicitly based [79]. We also propose additional properties that we believe are desirable, but are rarely stated explicitly.

Many of the cluster properties underlying existing definitions derive from the processes that lead to genome rearrangements. As genomes diverge, large-scale rearrangements break apart homologous regions, reducing the size and length of clusters. Gene duplications and losses cause the gene complement of homologous regions to drift apart, so that many genes will not have a homolog in the other region, and gene clusters will appear less dense. Smaller rearrangements will disrupt the gene order and orientation within homologous regions. Thus, clusters are often characterized according to their size, length, density, and the extent to which order and orientation are conserved. We discuss these properties in more detail below, as well as a number of additional properties that are rarely stated explicitly, but that we argue are nonetheless desirable.

Size: Almost all methods to evaluate clusters consider the size of a cluster, *i.e.* the number of homologous gene pairs contained within it. In general it is assumed that the more homologs in a cluster, the more likely it is to indicate common ancestry rather than chance similarities. An appropriate minimum size threshold will depend, however, on the specific cluster definition. For example, a cluster of four homologs in which order is conserved may be less likely to occur by chance, and thus more significant than an unordered cluster of size four.

Length: The length of a cluster, defined with respect to a particular genome, is the total number of genes spanned by the cluster. For example, in Figure 1.2(b), the upper left cluster is of size four, and spans two singletons, so is of total length six. In a whole genome comparison, the number of non-homologous genes spanned by the cluster in each genome may differ. However, if the processes that degrade a cluster are operating uniformly, then the length of the cluster in both genomes should be similar. Similarity of lengths is implicitly sought by the length constraint of r-windows, and explicitly sought in a clustering method proposed by Hampson *et al.* [73].

Density: Although over time gene insertions and losses will cause the gene content of homologous regions to diverge, in most cases we expect that significant similarity in gene content will be preserved. Thus, the majority of existing approaches attempt to find regions that are densely populated with homologs. We define the *global density* of a cluster as its size divided by its length. For a fixed value of r, the minimum global density of an r-window is set by choosing the parameter k. The only way to set a constraint on the global density of a max-gap cluster, on the other hand, is to reduce g, which will also reduce the maximum length of a cluster.

Even when a minimum global density is required, regions of a cluster may not be locally dense: a cluster could be composed of two very dense regions separated by a large region with no homologs. In this case, it might seem more natural to break the cluster into two separate clusters. Density as we have defined it here reflects the average gap size, but does not reflect the *variance* in gap sizes. The gap between adjacent marked genes in an r-window can be as large as $r-k$, whereas max-gap clusters guarantee that the maximum gap will be no more than g. Note that the two definitions have switched roles: the local density is easily controlled by the parameter g for max-gap clusters but there is no way to constrain the local density of r-window clusters without also further constraining the maximum cluster length. This trade-off between global and local density gives a simple illustration of how it can be difficult to design a cluster definition that satisfies our basic intuitions about cluster properties.

Order: For whole genome comparison, a cluster is considered ordered if the homologs in the second genome are in the identical or opposite order of the homologs in the first genome. For example, consider the two genomes shown in Figure 1.2. The clusters $\{\texttt{6,7}\}$ and $\{\texttt{8,9}\}$ are ordered, but $\{\texttt{5,6,7}\}$ and $\{\texttt{1,2,3,4}\}$ are not. Many cluster definitions require a strictly conserved gene order [11, 32, 179]. Over time, however, inversions will cause rearrangements, and thus conserved gene order is often considered too strict a requirement. In order to allow some short inversions, Hampson *et al.* [72] explicitly parameterize the number of order violations that are allowed in a cluster. A number of groups use heuristic, constructive methods that either implicitly enforce certain constraints on gene order, or explicitly bias their method to prefer clusters that form near-diagonals in the dot plot [30, 171, 175, 110]. The remainder, including r-windows and max-gap clusters, completely disregard gene order. However, as we explained in Section 2.3.3, though a number of groups *state* that they ignore gene order, constraints on gene order are often unintended consequences of algorithmic choices.

Orientation: Conserved spatial organization in bacterial genomes often points to functional associations between genes. In particular, clusters of genes in close proximity, with the same orientation, often indicate operons. In whole genome comparison of eukaryotes, similarities in gene orientation can provide additional evidence that two regions share a common ancestor. To the best of our knowledge, however, except for the method of Vision *et al.* [175], in which changes in orientation decrease the cluster score, existing definitions either require all genes in a cluster to have the same orientation, or disregard orientation altogether.

Temporal Coherence: Temporal information can be used to evaluate the significance of a putative homologous region identified through whole genome comparison. If a set of homologous genes all arose through the same speciation or duplication event, then the points in time at which each homolog pair diverged will be similar, and consequently we would expect our estimates of these divergence times group close together. However, all existing methods to find clusters are based solely on spatial information, and divergence times have been used only to estimate the age of a duplicated block identified based on spatial organization [11, 131], but not to assess the statistical significance of a cluster. In theory, combined analysis of temporal and spatial information could be used, for example, to increase our confidence that a region is the result of a single large-scale duplication event. However, due to the large error bounds that must be associated with any sequence-based estimate of divergence times [70, 117, 184], the practicality of such an approach is as yet unclear.

Nestedness: For whole genome comparison, one cluster property that is generally not considered explicitly, but may be assumed implicitly, is nestedness. A cluster of size k is *nested* if for each $h \in 1 \ldots k-1$ it contains a valid cluster of size h. Intuitively it may seem that any reasonable cluster definition should have this property. In fact, clusters with no ordering constraints are not necessarily nested. For example, Bergeron *et al.* [10] state a formal definition of max-gap clusters, and prove that there are maximal max-gap clusters of size k which do not contain any valid sub-cluster of size $2..k{-}1$. For example, when $g = 0$ they present a non-nested max-gap cluster with only four genes. The sequence of genes 1234 on one genome and 3142 on the other form a max-gap cluster of size four which does not contain any max-gap cluster of size two or three. Thus, nested max-gap clusters comprise only a subset of general max-gap clusters found through whole genome comparison.

There are no definitions that explicitly require that clusters be nested; rather, greedy search algorithms implicitly limit the results to nested clusters. Greedy algorithms use a bottom-up approach: each homologous gene pair serves as a cluster seed, and a cluster is extended by looking in its chromosomal neighborhood for another homologous gene pair close to the cluster on both genomes [30, 32, 73, 82]. It can be shown that any greedy search algorithm that constructs max-gap clusters iteratively, *i.e.* by constructing a cluster of size k by adding a gene to a cluster of size $k - 1$, will find *exactly* the set of all maximal nested max-gap clusters, as long as it considers each homologous gene pair as a seed for a potential cluster. In such cases, although order is not explicitly constrained, the search algorithm enforces implicit constraints on gene order: nested clusters can only get disordered to a limited degree. In most cases, however, such constraints are not acknowledged, and perhaps not even recognized.

Disjointness: If two clusters are not disjoint, *i.e.* the intersection of the marked genes they contain is not empty[1], our intuitive notion of a cluster may correspond more closely to the single island of overlapping windows than to the individual clusters. For example, in Figure 1.2, when $r = 5$, and $k = 4$ there are two r-windows: $\{5,6,7,9\}$ and $\{6,7,8,9\}$. Although both clusters contain genes 6, 7, and 9, there is no window of length five that contains all five of the genes. Thus, r-windows are not always disjoint. Indeed, it is surprisingly hard to find a cluster definition that guarantees that all clusters will be disjoint. The majority of definitions lead to overlapping clusters that must be merged or separated in an ad-hoc post-processing step for use by algorithms that require a unique tiling of regions. The only definition for which maximal clusters have been shown to be disjoint is the max-gap cluster [10], but only when homology relationships are one-to-one. When a gene may be matched with multiple genes, or when additional constraints are enforced (in addition to the maximum gap size), disjointness is quickly forfeited. For example, consider the consequences of requiring conserved order when looking for max-gap clusters in Figure 1.2. With a maximum gap of $g = 2$, five maximal g-clusters with conserved order are identified: $\{1,2\}$, $\{2,4\}$, $\{3,4,5,6,9\}$, $\{3,4,5,7,8\}$, and $\{3,4,5,7,9\}$. Although the last three clusters overlap, they cannot be merged without breaking the ordering constraint (due to the inversions of the segments containing genes 6 and 7 and genes 8 and 9).

More generally, a lack of disjointness strongly suggests that the cluster definition is too constrained. In the r-window example, these clusters are not disjoint *precisely* because the definition artificially constrains the length of a cluster. In the second example, the clusters were not disjoint because a definition with a strict ordering constraint was not able to capture the types of processes, such as inversions, that created the cluster.

[1] Note that it is possible, however, for two disjoint clusters to have overlapping spans in one of the genomes, as long as they do not share any homologs.

Isolation: If we observe a cluster with some additional homologous pairs in close proximity to its borders we might feel that the cluster border was arbitrary, and should extend to cover the neighboring island of genes. Thus, we propose that cluster definitions should guarantee that clusters will be *isolated*, that is: the maximum distance between marked genes in a cluster should always be less than the minimum distance between two clusters. A maximum-gap constraint guarantees that clusters will be isolated, but only barely—the gap within a cluster may be as large as g, whereas the gap separating two clusters may be just $g{+}1$.

Symmetry: For whole genome comparison, a desirable property that is rarely considered explicitly is whether the definition is symmetric with respect to genome. In some cases, such as the definition proposed by Calabrese *et al.* [30], a cluster is defined in such a way that whether a set of genes form a valid cluster may depend on whether genome G_1 or genome G_2 is represented by the vertical axis in the dot plot. Put another way, the set of clusters identified will differ depending on which genome is designated as the reference genome. A surprisingly large proportion of constructive definitions are not symmetric. These clustering algorithms require the selection of a reference genome even when there is no clear biological motivation for this choice. Definitions that are symmetric with respect to genome include r-windows and max-gap cluster definitions, as well as algorithms that represent the dot plot as a graph and use a symmetric distance function [128, 175].

* * *

The detailed catalog of cluster properties presented here will be useful for assessing whether definitions satisfy the intuitive notions upon which they are implicitly based, and whether these notions actually correspond to the types of structures present in real genomic data. Analysis of cluster properties can be useful for determining which characteristics actually reflect the types of structures found in real genomes, and thus which will best discriminate truly homologous regions from background noise (clusters of genes that occur by chance). Analyses of desirable cluster properties may also pave the way for new, possibly more powerful cluster definitions.

It is important to note that the importance of a property may depend on the goals of the study. For example, when clusters are being identified as a pre-processing step for reconstructing rearrangement histories, the exact boundaries and sizes of the cluster may be quite important [168]. In other cases, a researcher may wish to test a global hypothesis (such as finding evidence for one or two rounds of whole genome duplication), and may not necessarily care about the significance or boundaries of any specific cluster.

Even if it were known which properties reflect biologically and methodologically relevant features, designing a definition to satisfy those properties may not be straightforward because, in many cases, properties are not independent. Properties may interact in subtle ways—a definition that guarantees one desirable property will often fail to satisfy another. For example, one of the nice properties of the max-gap definition is that clusters are always disjoint. However, as shown above, adding additional constraints on order or length results in clusters that are no longer guaranteed to be disjoint. The subtle and sometimes undesirable interplay of some of these properties makes it difficult to devise a definition that satisfies them all. In fact, many of the most important properties are difficult to satisfy with the same definition. Thus, it remains an open question to what extent a single definition can capture all of these properties simultaneously.

Appendix C

Derivations of Max-Gap Expressions

C.1 Derivation of $d_g(c, u, s)$

For a given, non-zero integer s, $d_g(c, u, s)$ is the number of solutions to the following equation

$$\sum_{i=1}^{c} v_i + \sum_{j=1}^{u} w_j = s,$$

such that $0 \leq v_i \leq g, \forall i \in 1..c$ and $0 \leq w_j, \forall j \in 1..u$. The number of ways in which s can be obtained is the coefficient of x^s in the generating function

$$f(x) = (1 + x + x^2 + ... + x^g)^c \cdot (1 + x + x^2 + ...)^u.$$

Since $f(x)$ is the product of finite and infinite geometric series, it can be written as follows:

$$f(x) = \left(\sum_{i=0}^{g} x^i\right)^c \left(\frac{1}{1-x}\right)^u = \left(\frac{1-x^{g+1}}{1-x}\right)^c \left(\frac{1}{1-x}\right)^u = (1-x^{g+1})^c(1-x)^{-(c+u)}.$$

Expanding by application of the binomial theorem, we obtain:

$$f(x) = \sum_{i=0}^{c}(-1)^i \binom{c}{i} x^{i(g+1)} \sum_{l=0}^{\infty} \binom{c+u+l-1}{l} x^l.$$

In order to get the coefficient of x^s, we must include all terms where $i(g+1) + l = s$, which means that $l = s - i(g+1)$. Therefore,

$$d_g(c, u, s) = \sum_{i=0}^{c}(-1)^i \binom{c}{i} \binom{s - i(g+1) + c + u - 1}{s - i(g+1)}.$$

However, $s - i(g+1) > 0$ only when $i < s/(g+1)$, so the other terms do not contribute to the sum. Furthermore,

$$\binom{s - i(g+1) + c + u - 1}{s - i(g+1)} = \binom{s - i(g+1) + c + u - 1}{c + u - 1},$$

yielding the final expression

$$d_g(c,u,s) = \sum_{i=0}^{\lfloor s/(g+1)\rfloor} (-1)^i \binom{c}{i}\binom{s-i(g+1)+c+u-1}{c+u-1}.$$

C.2 Derivation of $d_g(m-1,1,l-m)$ from $d_g(m-1,0,l-m)$

In Section 2.1.2 we gave an expression $d_g(m-1,0,l-m)$ for the number of ways of arranging m marked genes in a max-gap g-chain of length *exactly* l. We obtain an expression for $d_g(m-1,1,l-m)$, the number of ways of arranging m black genes in a max-gap chain of length *no greater* than l, as follows:

$$\sum_{r=m}^{l} d_g(m–1,0,r–m) = \sum_{r=m}^{l}\sum_{i=0}^{\lfloor (r-m)/(g+1)\rfloor} (-1)^i \binom{m-1}{i}\binom{r-i(g+1)-2}{m-2},$$

The r in the upper bound of the second summation can be replaced by l because when $i > \lfloor (l-m)/(g+1)\rfloor$ the final binomial will be zero, which gives

$$\sum_{r=m}^{l}\sum_{i=0}^{\lfloor (l-m)/(g+1)\rfloor} (-1)^i \binom{m-1}{i}\binom{r-i(g+1)-2}{m-2}.$$

Now that the upper bound of the second summation is no longer dependent on r, the outer summation can be moved inward:

$$\sum_{i=0}^{\lfloor (l-m)/(g+1)\rfloor} (-1)^i \binom{m-1}{i}\sum_{r=m}^{l}\binom{r-i(g+1)-2}{m-2}.$$

Rewriting the bounds of the inner summation gives:

$$\sum_{i=0}^{\lfloor (l-m)/(g+1)\rfloor} (-1)^i \binom{m-1}{i}\sum_{r=m-i(g+1)-2}^{l-i(g+1)-2}\binom{r}{m-2}.$$

Decreasing the lower bound to $r = 0$ does not affect the probability because when $0 \le r < m-2$ the binomial is zero. We apply the upper summation identity (see Appendix C.5) to eliminate the inner summation, which yields

$$\sum_{i=0}^{\lfloor (l-m)/(g+1)\rfloor} (-1)^i \binom{m-1}{i}\binom{l-i(g+1)-1}{m-1},$$

which is exactly $d_g(m-1,1,l-m)$. The derivation of $d_g(m-1,2,l-m)$ from $d_g(m-1,1,l-m)$ is identical.

C.3 A closed-form expression for $d_g(m-1,2,L_m–m–1)$

The following three lemmas are needed to obtain a closed-form expression for $d_g(m–1,g,(m-1)g–1)$. Recall that the maximum possible length of a g-chain of size m is $L_m = m + g(m-1)$.

Lemma C.3.1. *For all l such that $m \leq l \leq L_m$, $d_g(m-1, 0, l-m) = d_g(m-1, 0, L_m{-}l)$.*

Proof. Let $S(m, g, l)$ be the set of g-chains of size m and length l, with no gap greater than g. Clearly, $|S(m, g, l)| = d_g(m-1, 0, l-m)$. Let $\langle g_1, ..., g_{m-1}\rangle$, where $0 \leq g_i \leq g$, denote a member of this set, *i.e.* a g-chain of size m and length $l = m + \sum_{i=1}^{m-1} g_i$, with gap sizes $g_1, ..., g_{m-1}$. Define a function $f(\langle g_1, ..., g_{m-1}\rangle) = \langle y_1, ..., y_{m-1}\rangle$, where $y_i = g - g_i$. We claim f maps $S(m, g, l)$ to $S(m, g, L_m{+}m{-}l)$. To see this, observe that $0 \leq y_i \leq g$, and the length of the chain $\langle y_1, ..., y_{m-1}\rangle$ is

$$m{+}\sum_{i=1}^{m-1} y_i = m{+}\sum_{i=1}^{m-1} g{-}g_i = m{+}(m{-}1)g{-}\sum_{i=1}^{m-1} g_i = m{+}(m{-}1)g{-}(l{-}m) = 2m{+}(m{-}1)g{-}l = L_m{+}m{-}l$$

Since f is a bijection, $|S(m, g, l)| = |S(m, g, L_m{+}m-l)|$, and thus $d_g(m-1, 0, l-m) = d_g(m-1, 0, L_m - l)$. □

Lemma C.3.2. *For all l such that $m \leq l \leq L_m$, $d_g(m-1, 1, l-m) + d_g(m-1, 1, L_m{-}l{-}1) = d_g(m-1, 1, L_m - m)$.*

Proof. By definition,

$$d_g(m-1, 1, l-m) + d_g(m-1, 1, L_m{-}l{-}1) = \sum_{i=m}^{l} d_g(m-1, 0, i-m) \; + \sum_{j=m}^{L_m+m-l-1} d_g(m-1, 0, j-m),$$

Lemma C.3.1 can be used to simplify the second term, yielding

$$\begin{aligned}\sum_{i=m}^{l} d_g(m-1, 0, i-m) &+ \sum_{j=l+1}^{L_m} d_g(m-1, 0, j-m) \\ &= \sum_{j=m}^{L_m} d_g(m-1, 0, j-m) = d_g(m-1, 1, L_m - m)\end{aligned}$$ □

Lemma C.3.3. *If $m \geq \frac{1}{g}{+}1$ and $(L_m{+}m{-}1)$ is even, then $2d_g(m{-}1, 1, \frac{1}{2}(L_m{+}m{-}1){-}m) = d_g(m{-}1, 1, L_m{-}m)$*

Proof.

$$d_g(m{-}1, 1, L_m{-}m) = \sum_{i=m}^{L_m} d_g(m{-}1, 0, i{-}m) = \sum_{i=m}^{\frac{1}{2}(L_m+m-1)} d_g(m{-}1, 0, i{-}m){+}\sum_{j=\frac{1}{2}(L_m+m+1)}^{L_m} d_g(m{-}1, 0, j{-}m)$$

which by Lemma C.3.1 is equal to

$$\begin{aligned}\sum_{i=m}^{\frac{1}{2}(L_m+m-1)} d_g(m-1, 0, i-m) + \sum_{j=\frac{1}{2}(L_m+m-1)}^{m} d_g(m-1, 0, j-m) &= \sum_{i=m}^{\frac{1}{2}(L_m+m-1)} 2d_g(m-1, 0, i-m) \\ &= 2d_g(m-1, 1, \frac{1}{2}(L_m + m - 1) - m)\end{aligned}$$ □

Theorem C.3.4. $d_g(m-1, 2, L_m{-}m{-}1) = \dfrac{L_m - m}{2}(g+1)^{m-1}$

Proof. Either $L_m + m$ is even or it is odd. When it is even $d_g(m-1, 2, L_m{-}m{-}1)$ is equivalent to

$$\sum_{i=m}^{L_m-1} d_g(m-1, 1, i-m) = \sum_{i=m}^{\frac{1}{2}(L_m+m)-1} d_g(m-1, 1, i-m) + \sum_{j=\frac{1}{2}(L_m+m)}^{L_m-1} d_g(m-1, 1, j-m).$$

Rewriting the summation index on the second term yields

$$\begin{aligned} &\sum_{i=m}^{\frac{1}{2}(L_m+m)-1} d_g(m-1, 1, i-m) \; + \sum_{j=\frac{1}{2}(L_m+m)-1}^{m} d_g(m-1, 1, L_m-j-1) \\ &= \sum_{i=m}^{\frac{1}{2}(L_m+m)-1} d_g(m-1, 1, i-m) + d_g(m-1, 1, L_m-i-1). \end{aligned}$$

By Lemma C.3.2, this simplifies to

$$\sum_{i=m}^{\frac{1}{2}(L_m+m)-1} d_g(m-1, 1, L_m-m) = \frac{L_m-m}{2}(g+1)^{m-1},$$

as desired.

Otherwise, if $L_m + m$ is odd, then $d_g(m-1, 2, L_m{-}m{-}1)$ is equivalent to

$$\sum_{i=m}^{\frac{1}{2}(L_m+m-3)} d_g(m-1, 1, i-m) \; + \; d_g(m-1, 1, \frac{1}{2}(L_m+m-1)-m) \; + \sum_{j=\frac{1}{2}(L_m+m+1)}^{L_m-1} d_g(m-1, 1, j-m).$$

The second term can be simplified by Lemma C.3.3, yielding

$$\frac{1}{2}d_g(m-1, 1, L_m-m) \; + \sum_{i=m}^{\frac{1}{2}(L_m+m-3)} d_g(m-1, 1, i-m) \; + \sum_{j=\frac{1}{2}(L_m+m+1)}^{L_m-1} d_g(m-1, 1, j-m).$$

As in the even case, the last two terms can be combined and simplified by Lemma C.3.2:

$$\frac{1}{2}d_g(m-1, 1, L_m-m) + \frac{(L_m-m-1)}{2}d_g(m-1, 1, L_m-m) = \frac{L_m-m}{2}(g+1)^{m-1},$$

as desired. □

C.4 Expected length and gap of a chain of m marked genes

Expected length The expected length, $E[l]$ of a complete chain of m marked genes (with no restriction on the gap sizes, *i.e.* $g = n$), placed randomly in a genome containing n genes is:

$$E[l] = \sum_{l=m}^{n} l \cdot prob(l) = \frac{1}{\binom{n}{m}} \cdot \sum_{l=m}^{n} l \cdot (n-l+1) \cdot \binom{l-2}{m-2} = \frac{1}{\binom{n}{m}} \cdot M = 1 + \frac{(m-1)(n+1)}{m+1}$$

$$\begin{aligned}
M &= \sum_{l=m}^{n}(n-l+1)\cdot(l-1)\binom{l-2}{m-2} \quad + \quad \sum_{l=m}^{n}(n-l+1)\cdot\binom{l-2}{m-2} \\
&= (m-1)\sum_{l=m}^{n}(n-l+1)\binom{l-1}{m-1} \quad + \quad \sum_{l=m}^{n}(n-l+1)\cdot\binom{l-2}{m-2} \\
&= (m-1)(n+1)\sum_{l=m}^{n}\binom{l-1}{m-1} - (m-1)\sum_{l=m}^{n} l\cdot\binom{l-1}{m-1} \quad + \quad (n+1)\sum_{l=m}^{n}\binom{l-2}{m-2} - \sum_{l=m}^{n} l\cdot\binom{l-2}{m-2} \\
&= A - B + C - D \\
&= \binom{n}{m}\cdot\left((m-1)(n+1) - m(m-1)\frac{n+1}{m+1} + \frac{m}{n}(n+1) - (m-1) \ - \ \frac{m}{n}\right) \\
&= \binom{n}{m}\cdot\left(1 + \frac{(m-1)(n+1)}{m+1}\right),
\end{aligned}$$

where A, B, C, and D are defined below, and simplified using the identities given in Appendix C.5.

$$\begin{aligned}
A &= (m-1)(n+1)\sum_{l=m}^{n}\binom{l-1}{m-1} = (m-1)(n+1)\binom{n}{m} \\
B &= (m-1)\sum_{l=m}^{n} l\cdot\binom{l-1}{m-1} = (m-1)\sum_{l=m}^{n} m\cdot\binom{l}{m} = m(m-1)\frac{n+1}{m+1}\binom{n}{m} \\
C &= (n+1)\sum_{l=m}^{n}\binom{l-2}{m-2} = \frac{m}{n}(n+1)\binom{n}{m} \\
D &= \sum_{l=m}^{n} l\cdot\binom{l-2}{m-2} = \sum_{l=m}^{n}(l-1)\cdot\binom{l-2}{m-2} \ + \ \sum_{l=m}^{n}\binom{l-2}{m-2} = (m-1)\sum_{l=m}^{n}\binom{l-1}{m-1} \ + \ \frac{m}{n}\binom{n}{m} \\
&= (m-1)\binom{n}{m} \ + \ \frac{m}{n}\binom{n}{m}.
\end{aligned}$$

Expected gap size The expected gap size, $E(g)$, if m genes are placed randomly in a genome of size n is:

$$\begin{aligned}
E[g] = \frac{E[l]-m}{m-1} &= \frac{1}{m-1} + \frac{(m-1)(n+1)}{(m+1)(m-1)} - \frac{m}{m-1} \\
&= \frac{m+1+(m-1)(n+1)-m(m+1)}{(m+1)(m-1)} = \frac{(m-1)(n-m)}{(m-1)(m+1)} = \frac{n-m}{m+1}
\end{aligned}$$

C.5 Useful combinatorial identities

The following three simple identities are used in the above proofs. For derivations see Graham *et al.* [69].

$$\begin{aligned}
\sum_{l=m}^{n}\binom{l-1}{m-1} = \sum_{x=0}^{n-1}\binom{x}{m-1} &= \binom{n}{m} \text{ by upper summation} \\
\sum_{l=m}^{n}\binom{l-2}{m-2} = \binom{n-1}{m-1} &= \frac{m}{n}\binom{n}{m} \text{ by an absorption identity} \\
\sum_{l=m}^{n}\binom{l}{m} = \binom{n+1}{m+1} &= \frac{n+1}{m+1}\binom{n}{m} \text{ by an absorption identity}
\end{aligned}$$

Bibliography

[1] ALTSCHUL, S. F. Evaluating the statistical significance of multiple distinct local alignments. In *Theoretical and Computational Methods in Genome Research*, Subai, Ed. 1997, pp. 1–14.

[2] AMORES, A., FORCE, A., L. YAN, Y., JOLY, L., AMEMIYA, C., FRITZ, A., HO, R., LANGELAND, J., PRINCE, V., WANG, Y. L., WESTERFIELD, M., EKKER, M., AND POSTLETHWAIT, J. H. Zebrafish hox clusters and vertebrate genome evolution. *Science 282* (1998), 1711–1714.

[3] APWEILER, R., BAIROCH, A., WU, CATHY, H., BARKER, WINONA, C., BOECKMANN, B., FERRO, S., GASTEIGER, E., HUANG, H., LOPEZ, R., MAGRANE, M., MARTIN, MARIA, J., NATALE, DARREN, A., O'DONOVAN, C., REDASCHI, N., AND YEH, LAI-SU, L. Uniprot: the universal protein knowledgebase. *Nucleic Acids Res 32* (Jan 2004), 115–119.

[4] ARABIDOPSIS GENOME INITIATIVE. Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana*. *Nature 408* (2000), 796–815.

[5] BANDYOPADHYAY, S., SHARAN, R., AND IDEKER, T. Systematic identification of functional orthologs based on protein network comparison. *Genome Res 16* (Mar 2006), 428–435.

[6] BANSAL, A. K. An automated comparative analysis of 17 complete microbial genomes. *Bioinformatics 15* (1999), 900–908. http://www.cs.kent.edu/∼arvind/orthos.html.

[7] BANSAL, A. K., BORK, P., AND STUCKEY, P. Automated pair-wise comparisons of complete microbial genomes. *Mathematical Modeling and Scientific Computing 9* (1998), 1–23.

[8] BEAL, M.-P., BERGERON, A., CORTEEL, S., AND RAFFINOT, M. An algorithmic view of gene teams. *Theoretical Computer Science 320* (June 2004), 395–418.

[9] BELDA, E., MOYA, A., AND SILVA, FRANCISCO, J. Genome rearrangement distances and gene order phylogeny in gamma-proteobacteria. *Mol Biol Evol 22* (Jun 2005), 1456–1467.

[10] BERGERON, A., CORTEEL, S., AND RAFFINOT, M. The algorithmic of gene teams. In *Workshop on Algorithmics in Bioinformatics (WABI)* (2002), D. Gusfield and R. Guigo, Eds., vol. 2452 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 464–476.

[11] BLANC, G., HOKAMP, K., AND WOLFE, K. H. A recent polyploidy superimposed on older large-scale duplications in the *Arabidopsis* genome. *Genome Res 13* (Feb 2003), 137–144.

[12] BLANCHETTE, M., KUNISAWA, T., AND SANKOFF, D. Gene order breakpoint evidence in animal mitochondrial phylogeny. *J Mol Evol 49* (1999), 193–203.

[13] BLIN, G., CHATEAU, A., CHAUVE, C., AND GINGRAS, Y. Inferring positional homologs with common intervals of sequences. In *RECOMB Workshop on Comparative Genomics* (2006), G. Bourque and N. El-Mabrouk, Eds., vol. 4205 of *Lecture Notes in Bioinformatics*, Springer Verlag, pp. 24–38.

[14] BLIN, G., CHAUVE, C., AND FERTIN, G. Gene order and phylogenetic reconstruction: application to gamma-Proteobacteria. In *RECOMB Workshop on Comparative Genomics* (Berlin Heidelberg, 2005), A. McLysaght and D. H. Huson, Eds., vol. 3678 of *Lecture Notes in Bioinformatics*, Springer Verlag, pp. 11–20.

[15] BLIN, G., FERTIN, G., AND CHAUVE, C. The breakpoint distance for signed sequences. In *CompBioNets 2004: Algorithms and computational methods for biochemical and evolutionary networks* (London, England, 2004), vol. 3 of *Texts in algorithmics*, Kings Coll Publications, pp. 3–16.

[16] BLIN, G., AND RIZZI, R. Conserved interval distance computation between non-trivial genomes. In *Proc. 11th International Computing and Combinatorics Conference (COCOON)* (2005), vol. 3595 of *Lecture Notes in Computer Science*, pp. 22–31.

[17] BOURGEOIS, F., AND LASSALLE, J.-C. An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Communications of the ACM 14* (1971), 802–804.

[18] BOURQUE, G., AND PEVZNER, P. Genome-scale evolution: Reconstructing gene orders in the ancestral species. *Genome Res 12* (2002), 26–36.

[19] BOURQUE, G., YASEF, Y., AND EL-MABROUK, N. Maximizing synteny blocks to identify ancestral homologs. In *RECOMB Workshop on Comparative Genomics* (Berlin Heidelberg, 2005), A. McLysaght and D. H. Huson, Eds., Lecture Notes in Bioinformatics, Springer Verlag, pp. 21–34.

[20] BOURQUE, G., ZDOBNOV, E., BORK, P., PEVZNER, P., AND TELSER, G. Genome rearrangements in human, mouse, rat and chicken. *Genome Res* (2004).

[21] BREJOVA, B., BROWN, D., AND VINAR, T. Optimal spaced seeds for homologous coding regions. In *Proceedings of Symposium on Combinatorial Pattern Matching (CPM'03)* (Morelia, Mexico, 2003), R. Baeza-Yates, E. Chávez, and M. Crochemore, Eds., vol. 2676 of *Lecture Notes in Computer Science*, Springer, pp. 42–54.

[22] BROWN, D., AND SJOLANDER, K. Functional classification using phylogenomic inference. *PLoS Comput Biol 2* (Jun 2006), 479–483.

[23] BRUDNO, M., MALDE, S., POLIAKOV, A., DO, CHUONG, B., COURONNE, O., DUBCHAK, I., AND BATZOGLOU, S. Glocal alignment: finding rearrangements during alignment. *Bioinformatics 19 Suppl 1* (2003), 54–62.

[24] BRYANT, D. The complexity of calculating exemplar distances. In *Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment, and the Evolution of Gene Families*, D. Sankoff and J. Nadeau, Eds., Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment, and the Evolution of Gene Families. Kluwer, Dordrecht, Netherlands, 2000, pp. 207–212.

[25] BUCKLEW, J. A. *Introduction to rare event simulation*. Springer Verlag, 2004.

[26] BUHLER, J., KEICH, U., AND SUN, Y. Designing seeds for similarity search in genomic DNA. In *Proceedings of the seventh annual international conference on Research in computational molecular biology* (2003), M. Vingron, S. Istrail, P. Pevzner, and M. Waterman, Eds., ACM Press, pp. 67–75.

[27] BURGETZ, I., SHARIFF, S., PANG, A., AND TILLIER, E. Positional homology in bacterial genomes. *Evol Bio online 2* (2006), 42–55.

[28] BYRNE, K., AND WOLFE, K. H. Visualizing syntenic relationships among the hemiascomycetes with the yeast gene order browser. *Nucleic Acids Res. 34* (2006), D452–D455.

[29] BYRNE, K. P., AND WOLFE, K. H. The Yeast Gene Order Browser: combining curated homology and syntenic context reveals gene fate in polyploid species. *Genome Res 15* (Oct 2005), 1456–1461.

[30] CALABRESE, P. P., CHAKRAVARTY, S., AND VISION, T. J. Fast identification and statistical evaluation of segmental homologies in comparative maps. *Bioinformatics 19* (2003), i74–80.

[31] CANNON, STEVEN, B., AND YOUNG, NEVIN, D. OrthoParaMap: distinguishing orthologs from paralogs by integrating comparative genome data and gene phylogenies. *BMC Bioinformatics 4* (Sep 2003), 35.

[32] CANNON, S. B., KOZIK, A., CHAN, B., MICHELMORE, R., AND YOUNG, N. D. DiagHunter and GenoPix2D: programs for genomic comparisons, large-scale homology discovery and visualization. *Genome Biol 4* (2003), R68.

[33] CANNON, S. B., MITRA, A., BAUMGARTEN, A., YOUNG, N. D., AND MAY, G. The roles of segmental and tandem gene duplication in the evolution of large gene families in *Arabidopsis thaliana*. *BMC Plant Biol 4* (Jun 2004), 10.

[34] CAVALCANTI, A. R. O., FERREIRA, R., GU, Z., AND LI, W.-H. Patterns of gene duplication in *Saccharomyces cerevisiae* and *Caenorhabditis elegans*. *J Mol Evol 56* (Jan 2003), 28–37.

[35] CHAUVE, C., FERTIN, G., RIZZI, R., AND VIALETTE, S. Genomes containing duplicates are hard to compare. In *Proc. of 2nd International Workshop on Bioinformatics Research and Applications (IWBRA 2006)* (Berlin, 2006), vol. 3992 of *Lecture Notes in Computer Science*, Springer, pp. 783–790.

[36] CHE, D., LI, G., MAO, F., WU, H., AND XU, Y. Detecting uber-operons in prokaryotic genomes. *Nucleic Acids Res 34* (2006), 2418–2427.

[37] CHEN, X., SU, Z., DAM, P., PALENIK, B., XU, Y., AND JIANG, T. Operon prediction by comparative genomics: an application to the *Synechococcus* sp. WH8102 genome. *Nucleic Acids Res 32* (2004), 2147–2157.

[38] CHEN, X., ZHENG, J., FU, Z., NAN, P., ZHONG, Y., LONARDI, S., AND JIANG., T. Assignment of orthologous genes via genome rearrangement. *IEEE/ACM Transactions on Computational Biology and Bioinformatics 2* (2005), 302–315. in press.

[39] CHIU, JOANNA, C., LEE, ERNEST, K., EGAN, MARY, G., SARKAR, I. N., CORUZZI, GLORIA, M., AND DESALLE, R. OrthologID: automation of genome-scale ortholog identification within a parsimony framework. *Bioinformatics 22* (Mar 2006), 699–707.

[40] Choi, K. P., and Zhang, L. Sensitive analysis and efficient method for identifying optimal spaced seeds. *Journal of Computer and System Sciences 68* (2004), 254–291.

[41] Clamp, M., et al. Ensembl 2002: accommodating comparative genomics. *Nucleic Acids Res 31* (Jan 2003), 38–42.

[42] Coghlan, A., and Wolfe, K. H. Fourfold faster rate of genome rearrangement in nematodes than in *Drosophila*. *Genome Res 12* (2002), 857–867.

[43] Corteel, S., Louchard, G., and Pemante, R. Common intervals in permutations. *Submitted* (2006).

[44] Cosner, M. E., Jansen, R. K., Moret, B. M. E., Raubeson, L. A., Wang, L.-S., Warnow, T., and Wyman, S. An empirical comparison of phylogenetic methods on chloroplast gene order data in *Campanulaceae*. In *Comparative Genomics*, D. Sankoff and J. H. Nadeau, Eds. Kluwer Academic Press, Dordrecht, NL, 2000, pp. 99–121.

[45] Coulier, F., Pontarotti, P., Roubin, R., Hartung, H., Goldfarb, M., and Birnbaum, D. Of worms and men: An evolutionary perspective on the fibroblast growth factor (FGF) and FGF receptor families. *J Mol Evol 44* (1997), 43–56.

[46] Danchin, E. G. J., Abi-Rached, L., Gilles, A., and Pontarotti, P. Conservation of the MHC-like region throughout evolution. *Immunogenetics 55* (Jun 2003), 141–8.

[47] Dehal, P., and Boore, J. L. Two rounds of whole genome duplication in the ancestral vertebrate. *PLoS Biol 3* (Oct 2005), e314.

[48] Didier, G. Common intervals of two sequences. In *Workshop on Algorithmics in Bioinformatics (WABI)* (2003), vol. 2812, Lecture Notes in Computer Science, pp. 17–24.

[49] Durand, D., and Hoberman, R. A. Diagnosing duplications: can it be done? *Trends Genet 22* (Mar 2006), 156–64.

[50] Durand, D., and Sankoff, D. Tests for gene clustering. *J Comput Biol 10* (2003), 453–482.

[51] Ehrlich, J., Sankoff, D., and Nadeau, J. Synteny conservation and chromosome rearrangements during mammalian evolution. *Genetics 147* (1997), 289–296.

[52] El-Mabrouk, N. Reconstructing an ancestral genome using minimum segments duplications and reversals. *J. Comput. Syst. Sci. 65* (2002), 442–464.

[53] El-Mabrouk, N., Nadeau, J. H., and Sankoff, D. Genome halving. In *Combinatorial Pattern Matching* (1998), Springer-Verlag, Ed., pp. 235–250.

[54] El-Mabrouk, N., and Sankoff, D. The reconstruction of doubled genomes. *SIAM Journal of Computing 32* (2003), 754–792.

[55] Endo, T., Imanishi, T., Gojobori, T., and Inoko, H. Evolutionary significance of intragenome duplications on human chromosomes. *Gene 205* (1997), 19–27.

[56] Engelhardt, B. E., Jordan, M. I., Muratore, K. E., and Brenner, S. E. Protein molecular function prediction by Bayesian phylogenomics. *PLoS Comput Biol 1* (Oct 2005), e45.

[57] Ermolaeva, M. D., White, O., and Salzberg, S. Prediction of operons in microbial genomes. *Nucleic Acids Res 5* (Mar 2001), 1216–1221.

[58] Fischer, G., Rocha, E. P. C., Brunet, F., Vergassola, M., and Dujon, B. Highly variable rates of genome rearrangements between hemiascomycetous yeast lineages. *PLoS Genet 2* (Mar 2006), e32.

[59] Fitch, W. M. Distinguishing homologous from analogous proteins. *Syst Zool 19* (Jun 1970), 99–113.

[60] Fitch, W. M. Uses for evolutionary trees. *Philos Trans R Soc Lond B Biol Sci 349* (Jul 1995), 93–102.

[61] Fitch, W. M. Homology: a personal view on some of the problems. *Trends Genet 16* (May 2000), 227–231.

[62] Friedman, R., and Hughes, A. L. Gene duplication and the structure of eukaryotic genomes. *Genome Res 11* (Mar 2001), 373–81.

[63] Fu, Z., Chen, X., Vacic, V., Nan, P., Yong, Y., and Jiang, T. A parsimony approach to genome-wide orthology assignment. In *RECOMB 2006* (Berlin Heidelberg, 2006), vol. 3909 of *Lecture Notes in Bioinformatics*, Springer Verlag, pp. 578–594.

[64] Fujibuchi, W., Ogata, H., Matsuda, H., and Kanehisa, M. Automatic detection of conserved gene clusters in multiple genomes by graph comparison and P-quasi grouping. *Nucleic Acids Res 28* (Oct 2000), 4029–36.

[65] Gibson, T., and Spring, J. Evidence in favour of ancient octaploidy in the vertebrate genome. *Biochem Soc Trans 2* (Feb 2000), 259–264.

[66] Glaz, J., and Balakrishnan, N. *Scan Statistics and Applications*. Birkhauser, Boston, 1999.

[67] Glaz, J., Naus, J., and Wallenstein, S. *Scan Statistics*. Springer Series in Statistics. Springer-Verlag, New York, NY, USA, 2001.

[68] Goodstadt, L., and Ponting, Chris, P. Phylogenetic reconstruction of orthology, paralogy, and conserved synteny for dog and human. *PLoS Comput Biol 2* (Sep 2006), e133.

[69] Graham, Knuth, and Patashnik. *Concrete Mathematics*. Addison-Wesley, 1989.

[70] Graur, D., and Martin, W. Reading the entrails of chickens: molecular timescales of evolution and the illusion of precision. *Trends Genet 20* (Feb 2004), 80–86.

[71] Haas, B. J., Delcher, A. L., Wortman, J. R., and Salzberg, S. L. DAGchainer: a tool for mining segmental genome duplications and synteny. *Bioinformatics 20* (Dec 2004), 3643–6.

[72] Hampson, S., McLysaght, A., Gaut, B., and Baldi, P. LineUp: statistical detection of chromosomal homology with application to plant comparative genomics. *Genome Res 13* (May 2003), 999–1010.

[73] Hampson, S. E., Gaut, B. S., and Baldi, P. Statistical detection of chromosomal homology using shared-gene density alone. *Bioinformatics 21* (Apr 2005), 1339–48.

[74] HANNENHALLI, S., CHAPPEY, C., KOONIN, E. V., AND PEVZNER, P. A. Genome sequence comparison and scenarios for gene rearrangements: A test case. *Genomics 30* (1995), 299–311.

[75] HE, X., AND GOLDWASSER, M. H. Identifying conserved gene clusters in the presence of orthologous groups. In *RECOMB* (2004), ACM Press, pp. 272–280.

[76] HE, X., AND GOLDWASSER, M. H. Identifying conserved gene clusters in the presence of homology families. *J Comput Biol 12* (2005), 638–656.

[77] HEBER, S., AND STOYE, J. Algorithms for finding gene clusters. In *Workshop on Algorithmics in Bioinformatics (WABI)* (2001), vol. 2149 of *Lecture Notes in Computer Science*, pp. 254–265.

[78] HENDRICKSON, H., AND LAWRENCE, JEFFREY, G. Selection for chromosome architecture in bacteria. *J Mol Evol 62* (May 2006), 615–629.

[79] HOBERMAN, R., AND DURAND, D. The incompatible desiderata of gene cluster properties. In *RECOMB Comparative Genomics Workshop* (2005), A. McLysaght and D. H. Huson, Eds., vol. 3678 of *Lecture Notes in Bioinformatics*, Springer Verlag, pp. 73–87.

[80] HOBERMAN, R., SANKOFF, D., AND DURAND, D. The statistical analysis of spatially clustered genes under the maximum gap criterion. *J Comput Biol 12* (Oct 2005), 1081–1100.

[81] HOBERMAN, R., SANKOFF, D., AND DURAND, D. The statistical significance of max-gap clusters. In *RECOMB Workshop on Comparative Genomics* (2005), J. Lagergren, Ed., vol. 3388 of *Lecture Notes in Bioinformatics*, Springer Verlag, pp. 55–71.

[82] HOKAMP, K. *A Bioinformatics Approach to (Intra-)Genome Comparisons*. PhD thesis, University of Dublin, Trinity College, October 2001.

[83] HUGHES, A. L. Phylogenetic tests of the hypothesis of block duplication of homologous genes on human chromosomes 6, 9, and 1. *Mol Biol Evol 15* (1998), 854–70.

[84] HUGHES, D. Evaluating genome dynamics: the constraints on rearrangements within bacterial genomes. *Genome Biol 1* (2000), REVIEWS0006.

[85] HULSEN, T., HUYNEN, MARTIJN, A., DE VLIEG, J., AND GROENEN, P. M. A. Benchmarking ortholog identification methods using functional genomics data. *Genome Biol 7* (2006), R31.

[86] HUYNEN, M. A., AND BORK, P. Measuring genome evolution. *Proc Natl Acad Sci 95* (May 1998), 5849–5856.

[87] HUYNEN, M. A., AND SNEL, B. Gene and context: integrative approaches to genome analysis. *Adv Protein Chem 54* (2000), 345–79.

[88] INTERNATIONAL HUMAN GENOME SEQUENCING CONSORTIUM. Initial sequencing and analysis of the human genome. *Nature 409* (2001), 860–921.

[89] JAILLON, O., ET AL. Genome duplication in the teleost fish *Tetraodon nigroviridis* reveals the early vertebrate proto-karyotype. *Nature 431* (Oct 2004), 946–957.

[90] KASAHARA, M. New insights into the genomic organization and origin of the major histocompatibility complex: role of chromosomal (genome) duplication in the emergence of the adaptive immune system. *Hereditas 127* (1997), 59–65.

[91] KATSANIS, N., FITZGIBBON, J., AND FISHER, E. Paralogy mapping: identification of a region in the human MHC triplicated onto human chromosomes 1 and 9 allows the prediction and isolation of novel PBX and NOTCH loci. *Genomics 35* (1996), 101–108.

[92] KEICH, U., LI, M., MA, B., AND TROMP, J. On spaced seeds for similarity search. *Discrete Applied Math 138* (2004), 253–263.

[93] KELLIS, M., BIRREN, B. W., AND LANDER, E. S. Proof and evolutionary analysis of ancient genome duplication in the yeast *Saccharomyces cerevisiae*. *Nature 428* (Apr 2004), 617–624.

[94] KELLIS, M., PATTERSON, N., BIRREN, B., BERGER, B., AND LANDER, E. S. Methods in comparative genomics: genome correspondence, gene identification and regulatory motif discovery. *J Comput Biol 11* (2004), 319–55.

[95] KOLSTO, A. B. Dynamic bacterial genome organization. *Molecular Microbiology 24* (1997), 241–8.

[96] KU, H.-M., VISION, T., LIU, J., AND TANKSLEY, S. D. Comparing sequenced segments of the tomato and *Arabidopsis* genomes: large-scale duplication followed by selective gene loss creates a network of synteny. *PNAS 97* (2000), 9121–9126.

[97] LAWRENCE, J., AND ROTH, J. R. Selfish operons: horizontal transfer may drive the evolution of gene clusters. *Genetics 143* (1996), 1843–60.

[98] LERAT, E., DAUBIN, V., AND MORAN, NANCY, A. From gene trees to organismal phylogeny in prokaryotes: the case of the gamma-proteobacteria. *PLoS Biol 1* (Oct 2003), E19.

[99] LI, L., STOECKERT, C. J., AND ROOS, D. S. OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Res 13* (Sep 2003), 2178–89.

[100] LI, W.-H., YANG, J., AND GU, X. Expression divergence between duplicate genes. *Trends Genet 21* (Nov 2005), 602–607.

[101] LIPOVICH, L., LYNCH, E. D., LEE, M. K., AND KING, M.-C. A novel sodium bicarbonate cotransporter-like gene in an ancient duplicated region: *SLC4A9* at 5q31. *Genome Biol 2* (2001), 0011.1–0011.13.

[102] LUC, N., RISLER, J., BERGERON, A., AND RAFFINOT, M. Gene teams: a new formalization of gene clusters for comparative genomics. *Comput Biol Chem 27* (2003), 59–67.

[103] LUNDIN, L. G. Evolution of the vertebrate genome as reflected in paralogous chromosomal regions in man and the house mouse. *Genomics 16* (1993), 1–19.

[104] LYNCH, M., AND CONERY, J. S. The evolutionary fate and consequences of duplicate genes. *Science 290* (Nov 2000), 1151–1155.

[105] LYNCH, M., O'HELY, M., WALSH, B., AND FORCE, A. The probability of preservation of a newly arisen gene duplicate. *Genetics 159* (2001), 1789–1804.

[106] MABLE, B. K. Breaking down taxonomic barriers in polyploidy research. *Trends Plant Sci 8* (Dec 2003), 582–590.

[107] MABLE, B. K. Why polyploidy is rarer in animals than in plants: myths and mechanisms. *Biological Journal of the Linnean Society 82* (Aug 2004), 453–466.

[108] MAERE, S., BODT, S. D., RAES, J., CASNEUF, T., MONTAGU, M. V., KUIPER, M., AND VAN DE PEER, Y. Modeling gene and genome duplications in eukaryotes. *Proc Natl Acad Sci U S A 102* (Apr 2005), 5454–5459.

[109] MAU, B., DARLING, A. E., AND PERNA., N. T. Identifying evolutionarily conserved segments among multiple divergent and rearranged genomes. In *RECOMB Workshop on Comparative Genomics* (Bertinoro, Italy, October 2004), Lagergren, Ed., Lecture Notes in Bioinformatics, Springer Verlag, pp. 72–84.

[110] MCLYSAGHT, A., HOKAMP, K., AND WOLFE, K. H. Extensive genomic duplication during early chordate evolution. *Nat Genet 31* (Jun 2002), 200–204.

[111] MIRNY, LEONID, A., AND GELFAND, MIKHAIL, S. Using orthologous and paralogous proteins to identify specificity-determining residues in bacterial transcription factors. *J Mol Biol 321* (Aug 2002), 7–20.

[112] MOUSE GENOME SEQUENCING CONSORTIUM. Initial sequencing and comparative analysis of the mouse genome. *Nature 420* (2002), 520–562.

[113] MURPHY, W. J., PEVZNER, P. A., AND O'BRIEN, S. J. Mammalian phylogenomics comes of age. *Trends Genet 20* (Dec 2004), 631–9.

[114] NADEAU, J., AND SANKOFF, D. Counting on comparative maps. *Trends Genet 14* (1998), 495–501.

[115] NADEAU, J. H., AND SANKOFF, D. The lengths of undiscovered conserved segments in comparative maps. *Mamm Genome 9* (1998), 491–495.

[116] NADEAU, J. H., AND TAYLOR, B. A. Lengths of chromosomal segments conserved since the divergence of man and mouse. *Proc Natl Acad Sci U S A 81* (1984), 814–818.

[117] NEI, M., AND KUMAR, S. *Molecular Evolution and Phylogenetics.* Oxford University Press, 2000.

[118] O'BRIEN, K. P., REMM, M., AND SONNHAMMER, E. L. L. Inparanoid: a comprehensive database of eukaryotic orthologs. *Nucleic Acids Res 33* (Jan 2005), D476–80. Version 4.0, downloaded May 2005.

[119] O'BRIEN, S. J., MENOTTI-RAYMOND, M., MURPHY, W. J., NASH, W. G., WIENBERG, J., STANYON, R., COPELAND, N. G., JENKINS, N. A., WOMACK, J. E., AND GRAVES, J. A. M. The promise of comparative genomics in mammals. *Science 286* (Oct 1999), 458–62, 479–81.

[120] O'BRIEN, S. J., WIENBERG, J., AND LYONS, L. A. Comparative genomics: lessons from cats. *Trends Genet 10* (Oct 1997), 393–399.

[121] OCHMAN, H., LAWRENCE, J. G., AND GROISMAN, E. A. Lateral gene transfer and the nature of bacterial innovation. *Nature 405* (May 2000), 299–304.

[122] OGATA, H., FUJIBUCHI, W., GOTO, S., AND KANEHISA, M. A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters. *Nucleic Acids Res 28* (Oct 2000), 4021–8.

[123] OLINSKI, R. P., LUNDIN, L. G., AND HALLBÖÖK, F. Conserved synteny between the Ciona genome and human paralogons identifies large duplication events in the molecular evolution of the insulin-relaxin gene family. *Mol Biol Evol 23* (Jan 2006), 10–22.

[124] OVERBEEK, R., FONSTEIN, M., D'SOUZA, M., PUSCH, G. D., AND MALTSEV, N. The use of gene clusters to infer functional coupling. *Proc Natl Acad Sci U S A 96* (Mar 1999), 2896–2901.

[125] PASEK, S., BERGERON, A., RISLER, J.-L., LOUIS, A., OLLIVIER, E., AND RAFFINOT, M. Identification of genomic features using domain teams. Tech. rep., Laboratoire Génome et Informatique, Evry, France, 2004.

[126] PASEK, S., BERGERON, A., RISLER, J.-L., LOUIS, A., OLLIVIER, E., AND RAFFINOT, M. Identification of genomic features using microsyntenies of domains : domain teams. *Genome Res 15* (2005), 867–874. In press.

[127] PEBUSQUE, M., COULIER, F., BIRNBAUM, D., AND PONTAROTTI, P. Ancient large-scale genome duplications: phylogenetic and linkage analyses shed light on chordate genome evolution. *Mol Biol Evol 15* (1998), 1145–1159.

[128] PEVZNER, P., AND TESLER, G. Genome rearrangements in mammalian evolution: lessons from human and mouse genomes. *Genome Res 13* (Jan 2003), 37–45.

[129] PEVZNER, P. A. *Computational Molecular Biology: An Algorithmic Approach*. MIT Press, Cambridge, MA, 2000.

[130] PRICE, M. N., HUANG, K. H., ALM, E. J., AND ARKIN, A. P. A novel method for accurate operon predictions in all sequenced prokaryotes. *Nucleic Acids Res 33* (2005), 880–92.

[131] RAES, J., VANDEPOELE, K., SIMILLION, C., SAEYS, Y., AND VAN DE PEER, Y. Investigating ancient duplication events in the *Arabidopsis* genome. *J Struct Funct Genomics 3* (2003), 117–29.

[132] RAGHUPATHY, N., AND DURAND, D. Individual gene cluster statistics in noisy maps. In *RECOMB Workshop on Comparative Genomics* (2005), vol. 3678 of *Lecture Notes in Bioinformatics*, Springer Verlag, pp. 106–120.

[133] RAGHUPATHY, N., HOBERMAN, R., AND DURAND, D. Two plus two does not equal three: Statistical tests for multiple genome comparison. In *Proceedings of the 5th Asia-Pacific Bioinformatics Conference* (2007), Series on Advances in Bioinformatics and Computational Biology, Imperial College Press. In press.

[134] REMM, M., STORM, C. E., AND SONNHAMMER, E. L. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *J Mol Biol 314* (Dec 2001), 1041–1052.

[135] ROGOZIN, I. B., MAKAROVA, K. S., WOLF, Y. I., AND KOONIN, E. V. Computational approaches for the analysis of gene neighbourhoods in prokaryotic genomes. *Brief Bioinform 5* (Jun 2004), 131–49.

[136] RUVINSKY, I., AND SILVER, L. M. Newly indentified paralogous groups on mouse chromosomes 5 and 11 reveal the age of a T-box cluster duplication. *Genomics 40* (1997), 262–266.

[137] SALGADO, H., GAMA-CASTRO, S., MARTNEZ-ANTONIO, A., DAZ-PEREDO, E., SNCHEZ-SOLANO, F., PERALTA-GIL, M., GARCIA-ALONSO, D., JIMNEZ-JACINTO, V., SANTOS-ZAVALETA, A., BONAVIDES-MARTNEZ, C., AND COLLADO-VIDES, J. RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in *Escherichia coli* K-12. *Nucleic Acids Res 32* (Jan 2004), D303–6.

[138] SANKOFF, D. Genome rearrangement with gene families. *Bioinformatics 15* (1999), 909–917.

[139] SANKOFF, D. Short inversions and conserved gene clusters. *Bioinformatics 18* (2002), 1305–1308.

[140] SANKOFF, D., BRYANT, D., DENEAULT, M., LANG, B. F., AND BURGER, G. Early eukaryote evolution based on mitochondrial gene order breakpoints. *J Comput Biol 3–4* (2000), 521–535.

[141] SANKOFF, D., DENEAULT, M., BRYANT, D., LEMIEUX, C., AND TURMEL, M. Chloroplast gene order and the divergence of plants and algae from the normalized number of induced breakpoints. In *Comparative Genomics*, D. Sankoff and J. H. Nadeau, Eds. Kluwer Academic Press, Dordrecht, NL, 2000, pp. 89–98.

[142] SANKOFF, D., AND EL-MABROUK, N. Genome rearrangement. In *Current Topics in Computational Biology* (2002), T. Jiang, T. Smith, Y. Xu, and M. Zhang, Eds., MIT Press, pp. 135–155.

[143] SANKOFF, D., FERRETTI, V., AND NADEAU, J. H. Conserved segment identification. *J Comput Biol 4* (1997), 559–565.

[144] SANKOFF, D., AND HAQUE, L. Power boosts for cluster tests. In *RECOMB Comparative Genomics Workshop* (2005), A. McLysaght and D. H. Huson, Eds., vol. 3678 of *Lecture Notes in Bioinformatics*, Springer Verlag, pp. 121–130.

[145] SANKOFF, D., AND NADEAU, J. H. Chromosome rearrangements in evolution: From gene order to genome sequence and back. *Proc Natl Acad Sci U S A 100* (Sep 2003), 11188–11189.

[146] SCANNELL, D. R., BYRNE, K. P., GORDON, J. L., WONG, S., AND WOLFE, K. H. Multiple rounds of speciation associated with reciprocal gene loss in polyploid yeasts. *Nature 440* (Mar 2006), 341–345.

[147] SEARLS, D. B. Pharmacophylogenomics: genes, evolution and drug targets. *Nat Rev Drug Discov 2* (Aug 2003), 613–623.

[148] SEMPLE, C., AND WOLFE, K. H. Gene duplication and gene conversion in the *Caenorhabditis elegans* genome. *J Mol Evol 48* (1999), 555–64.

[149] SEOIGHE, C., AND WOLFE, K. Extent of genomic rearrangement after genome duplication in yeast. *Proc Natl Acad Sci U S A 95* (1998), 4447–4452.

[150] SEOIGHE, C., AND WOLFE, K. H. Updated map of duplicated regions in the yeast genome. *Gene 238* (1999), 253–261.

[151] SIMILLION, C., VANDEPOELE, K., MONTAGU, M. V., ZABEAU, M., AND VAN DE PEER, Y. The hidden duplication past of *Arabidopsis thaliana*. *Proc Natl Acad Sci U S A 99* (2002), 13627–32.

[152] SIMILLION, C., VANDEPOELE, K., AND VAN DE PEER, Y. Recent developments in computational approaches for uncovering genomic homology. *Bioessays 26* (Nov 2004), 1225–35.

[153] SKOVGAARD, M., JENSEN, L. J., BRUNAK, S., USSERY, D., AND KROGH, A. On the total number of genes and their length distribution in complete microbial genomes. *Trends Genet 17* (Aug 2001), 425–428.

[154] SMITH, N. G. C., KNIGHT, R., AND HURST, L. D. Vertebrate genome evolution: a slow shuffle or a big bang. *BioEssays 21* (1999), 697–703.

[155] SOLTIS, P. S., AND SOLTIS, D. E. The role of genetic and genomic attributes in the success of polyploids. *Proc Natl Acad Sci U S A 97* (Jun 2000), 7051–7057.

[156] SONG, N., DAVIS, G. B., AND DURAND, D. Homology identification for multi-domain proteins. *PNAS* (2005). Submitted.

[157] SPRING, J. Genome duplication strikes back. *Nature Genetics 31* (2002), 128–129.

[158] STORM, C. E. V., AND SONNHAMMER, E. L. L. Automated ortholog inference from phylogenetic trees and calculation of orthology reliability. *Bioinformatics 18* (Jan 2002), 92–99.

[159] SUYAMA, M., AND BORK, P. Evolution of prokaryotic gene order: genome rearrangements in closely related species. *Trends Genet 1* (Jan 2001), 10–13.

[160] SWENSON, K., PATTENGALE, N. D., AND MORET, B. M. E. A framework for orthology assignment from gene rearrangement data. In *RECOMB Workshop on Comparative Genomics* (2005), vol. 3678 of *Lecture Notes in Bioinformatics*, RECOMB, Springer Verlag, pp. 153–166.

[161] SWENSON, K. M., MARRON, M., EARNEST-DEYOUNG, J. V., AND MORET, B. M. E. Approximating the true evolutionary distance between two genomes. In *Proc. 7th Workshop on Algorithm Engineering and Experiments* (2005), SIAM Press, pp. 121–129.

[162] TAMAMES, J. Evolution of gene order conservation in prokaryotes. *Genome Biol 6* (2001), 0020.1–0020.11.

[163] TAMAMES, J., CASARI, G., OUZOUNIS, C., AND VALENCIA, A. Conserved clusters of functionally related genes in two bacterial genomes. *J Mol Evol 44:* (1997), 66–73.

[164] TAMAMES, J., GONZALEZ-MORENO, M., VALENCIA, A., AND VICENTE, M. Bringing gene order into bacterial shape. *Trends Genet 3* (Mar 2001), 124–126.

[165] TATUSOV, R. L., FEDOROVA, N. D., JACKSON, J. D., JACOBS, A. R., KIRYUTIN, B., KOONIN, E. V., KRYLOV, D. M., MAZUMDER, R., MEKHEDOV, S. L., NIKOLSKAYA, A. N., RAO, B. S., SMIRNOV, S., SVERDLOV, A. V., VASUDEVAN, S., WOLF, Y. I., YIN, J. J., AND NATALE, D. A. The COG database: an updated version includes eukaryotes. *BMC Bioinformatics 4* (Sep 2003), 41.

[166] TATUSOV, R. L., KOONIN, E. V., AND LIPMAN, D. J. A genomic perspective on protein families. *Science 278* (Oct 1997), 631–637.

[167] TRACHTULEC, Z., AND FOREJT, J. Synteny of orthologous genes conserved in mammals, snake, fly, nematode, and fission yeast. *Mamm Genome 3* (Mar 2001), 227–231.

[168] TRINH, P., MCLYSAGHT, A., AND SANKOFF, D. Genomic features in the breakpoint regions between syntenic blocks. *Bioinformatics 20 Suppl 1* (Aug 2004), I318–I325.

[169] UNO, T., AND YAGIURA, M. Fast algorithms to enumerate all common intervals of two permutations. *Algorithmica 26* (2000), 290–309.

[170] USPENSKY, J. V. *Introduction to Mathematical Probability*. McGraw-Hill, New York, 1937, pp. 23–24.

[171] VANDEPOELE, K., SAEYS, Y., SIMILLION, C., RAES, J., AND VAN DE PEER, Y. The automatic detection of homologous regions (ADHoRe) and its application to microcolinearity between *Arabidopsis* and rice. *Genome Res 12* (2002), 1792–801.

[172] VANDEPOELE, K., SIMILLION, C., AND VAN DE PEER, Y. Detecting the undetectable: uncovering duplicated segments in *Arabidopsis* by comparison with rice. *Trends Genet 18* (2002), 604–6.

[173] VANDEPOELE, K., SIMILLION, C., AND VAN DE PEER, Y. Evidence that rice and other cereals are ancient aneuploids. *Plant Cell 15* (Sep 2003), 2192–2202.

[174] VENTER, J. C., ET AL. The sequence of the human genome. *Science 291* (2001), 1304–1351.

[175] VISION, T. J., BROWN, D. G., AND TANKSLEY, S. D. The origins of genomic duplications in *Arabidopsis*. *Science 290* (2000), 2114–2117.

[176] WESTON, J., ELISSEEFF, A., ZHOU, D., LESLIE, C. S., AND NOBLE, W. S. Protein ranking: from local to global structure in the protein similarity network. *PNAS 101* (2004), 6559–6563.

[177] WESTOVER, B. P., BUHLER, J. D., SONNENBURG, J. L., AND GORDON, J. I. Operon prediction without a training set. *Bioinformatics 21* (Apr 2005), 880–8.

[178] WHEELER, D. L., ET AL. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res 33* (Jan 2005), D39–45.

[179] WOLF, Y. I., ROGOZIN, I. B., KONDRASHOV, A. S., AND KOONIN, E. V. Genome alignment, evolution of prokaryotic genome organization, and prediction of gene function using genomic context. *Genome Res 11* (Mar 2001), 356–72.

[180] WOLFE, K. H., AND SHIELDS, D. C. Molecular evidence for an ancient duplication of the entire yeast genome. *Nature 387* (1997), 708–713.

[181] YANAI, I., MELLOR, J. C., AND DELISI, C. Identifying functional links between genes using conserved chromosomal proximity. *Trends Genet 18* (Apr 2002), 176–9.

[182] YANG, Z., AND NIELSEN, R. Codon-substitution models for detecting molecular adaptation at individual sites along specific lineages. *Mol Biol Evol 19*, 6 (Jun 2002), 908–917.

[183] YU, J., ET AL. The Genomes of *Oryza sativa*: a history of duplications. *PLoS Biol 3* (Feb 2005).

[184] ZHANG, L., VISION, T. J., AND GAUT, B. S. Patterns of nucleotide substitution among simultaneously duplicated gene pairs in Arabidopsis thaliana. *Mol Biol Evol 19* (Sep 2002), 1464–1473.

[185] Zheng, Xiangqun, H., Lu, F., Wang, Z.-Y., Zhong, F., Hoover, J., and Mural, R. Using shared genomic synteny and shared protein functions to enhance the identification of orthologous gene pairs. *Bioinformatics 21* (Mar 2005), 703–710.

[186] Zheng, Y., Szustakowski, J. D., Fortnow, L., Roberts, R. J., and Kasif, S. Computational identification of operons in microbial genomes. *Genome Res 12* (Aug 2002), 1221–1230.

[187] Zmasek, C. M., and Eddy, S. R. RIO: analyzing proteomes by automated phylogenomics using resampled inference of orthologs. *BMC Bioinformatics 3* (May 2002), 14.