AutoOrthoGen

Multiple Genome Alignment and Comparison

Orion Buske, Yogesh Saletore, Kris Weber

CSE 428 Spring 2009

Martin Tompa

Computer Science and Engineering

University of Washington, Seattle, WA

Abstract

AutoOrthoGen is a novel software solution to finding sets of orthologous genes in genomes. The approach is to augment Mauve's genomic alignment with seeds of pairs of orthologous genes from Blast comparisons of proteins in each genome, and then cluster these seeds based on synteny provided by Mauve's LCBs. The software solution uses a database backend to simplify and accelerate seeds. AutoOrthoGen has been tested against four species of *Legionella pneuomophila* found in the Alignable Tight Genomic Clusters (ATGC) database and against the "Evolver" data sets artificially created. The results indicate that the AutoOrthoGen solution is highly successful at identifying orthologous sets of genes, and finding unique, common, and missing genes. This may be of substantial benefit to researchers in microbiology, as they seek to use previous knowledge of related genomes to better understand new ones.

Introduction

Recent advances in genomics have allowed scientists to sequence the genomes of thousands of organisms. This was once heralded as the solution to solving evolutionary trees and determining the phylogeny of various organisms. However, finding similarities between two genomes is no easy task, and the difficulty grows exponentially as more and more genomes are added. Bacterial genomes are much smaller than the sophisticated genomes of Eukaryotic organisms, but still pose a daunting task for scientists to compare.

The difficulty in aligning the genomes is a result of the multiple mutations that can occur on the nucleotide level. These mutations are what drive natural selection and evolution and result in the creation of new species and new organisms. Simple insertions and deletions can result in the addition or removal of a single base pair in the sequence, changing the structure of the genome. Sequences from other organisms can be inserted into the existing genome through a lateral transfer event. Sequences from the current genome can also be copied back into the same genome, resulting in sequences that are paralogous to those in the genome.

The coding sequences of the genome are regions that can be translated into proteins, which are essential to all life. Proteins are amino acid polymers, and each protein is defined by a gene, or a DNA sequence in the genome. As with the non-coding regions of the genome, genes also can undergo mutations, including lateral transfers and the creation of paralogous genes. Genes can be used as phylogenetic markers to help in the comparison of entire sets of genomes, by finding orthologous genes, or orthologs. Scientists have also studied the structure and function of thousands of proteins. The National Center for Biotechnology Information (NCBI) maintains millions of genome sequences, many with added annotations identifying the amino acid chain for each protein that the genome codes for.

The purpose of AutoOrthoGen (AOG) is to use protein comparisons to augment existing genome alignments. Existing tools can provide some foundation for the basic alignment of two genomes. This information can be augmented with protein-level comparisons, using synteny as a guide. Synteny refers to the relative preservation of the order of genes during evolution, and implies that two orthologous segments can be used to find additional adjacent orthologous segments.

In addition to the problem of developing a better algorithm for finding orthologous genes, there does not exist a well-defined method to gauge the effectiveness of the algorithm. Several methods were defined for this purpose: running the algorithm on a small database of clearly defined test cases, comparing its results to those of a "strawman" algorithm (which maximizes sensitivity at the expense of positive predictive value), comparing its results to an existing orthologous gene-finding algorithms, and testing its accuracy in finding known orthologous genes on genomes produced through simulated evolution. To measure the accuracy of the solution to the specific problem of finding orthologous genes, adapted versions of standard statistical measures were used.

Related Work

Mauve is a multiple genome alignment software that relies on aligning locally collinear blocks (LCBs), which are segments of the genome that can be aligned with each other. Mauve avoids the problem of paralogous segments by focusing on segments in each genome that do not have a paralogous segment in the current genome but do have orthologous segments in other genomes. These multi-MUMs are used to implement a phylogenetic tree, and then partitioned into LCBs that serve as anchors across the genome. The Mauve viewer uses the LCBs to align the genomes in a visual display. By ignoring paralogous segments altogether, Mauve successfully aligns much of the genomes. However, there are instances when Mauve will group some orthologous genes into an LCB, but not include adjacent genes that are also orthologous.

ClustalW is another tool that uses dynamic programming, in addition to other computations, to align two or more genomes. It is especially useful to align gene sequences and protein amino acid sequences, and the robust algorithm can handle simple insertions and deletions. However, paralogous segments make it difficult to use ClustalW to align entire genomes.

BLAST is a tool that compares sequences on the nucleotide or protein level, and computes the relative similarities between the two, computing the S-score and the e-value. The S-score is a measure of how similar the two sequences are, and the e-value is the probability that there exists another sequence with a higher S-score. BLAST can be used to compare protein amino acid sequences to determine the similarity between proteins in two different organisms, which can aid in the calculation of phylogenetic tree structures.

Alignable Tight Genomic Clusters (ATCG) is a database that maintains lists of orthologous genes for many genomes. They uses BLAST to determine similarities between genome sequences, and then using a sliding window for synteny to find orthologous genes. The database is limited but can serve as a basis to test against. In addition, the Evolver team has produced a simulation of evolution of a genome. This provides an artificially generated, and therefore known basis to use for comparison and testing.

AutoOrthoGen Solution

The program

AutoOrthoGen takes a set of genbank (.gbk) files corresponding to closely related prokaryotes as input. As output, it generates nine files which detail different aspects of the ortholog family predictions:

- blockOut.log: contains a list of final state of all blocks upon program completion.
- familyOut.atgc: contains an ATGC-formatted list of the predicted ortholog families produced by AutoOrthoGen. This list may contain conflicted families (families with more than one gene per genome).
- commonOut.log: contains a list of all genes grouped in an ortholog family spanning all the input genomes.
- spanAllOut.atgc: contains the same genes as commonOut.log, but entries are ATGC-formatted and grouped by ortholog family.
- orphanLongOut.log: contains a list of all genes not in any predicted ortholog family but that match other genes too well to be considered unique. For each gene, contains a list of all genes it matched better than an e-value threshold, sorted by increasing e-value.
- orphanShortOut.log: contains the same genes as orphanLongOut.log, but only lists the best match on each genome.
- uniqueOut.log: contains a list of all genes not in any predicted ortholog family and are not listed as an "orphan"
- conflictOut.log: contains a list of the predicted ortholog families that contain more than one gene on any one genome.
- summaryOut.log: contains a summary of the predicted ortholog families, including number of conflicted, common, and total families predicted. These results are then broken down by genome, displaying the number of unique, common, missing, orphaned, orthologous, and total genes in that genome.

Acquire necessary data

From the input genbank files, Mauve is used to generate a nucleotide-sequence-based alignment of these genomes, outputting a set of locally collinear blocks (LCBs) over subsets of these genomes. BLAST is used to evaluate the amino acid sequence similarity of each annotated protein in these genomes with every other one. Because BLAST (specifically *blastp*) takes FASTA protein (.faa) files as input, the genbank files are converted to FASTA format with the *gbk2faa.py* script. The *blastallvall.py* script then blasts every gene in the FASTA files against every other gene in them and outputs all results that match with an e-value less than a given threshold.

Load data into tables

Before analysis begins, relevant data are loaded into three tables in a MySQL database. The schema for these tables, and one table that is generated from the previous three, is shown in Table 1. The Mauve alignment file is parsed into the *lcbs* table, which contains a list of all of the LCBs that mauve found and what they cover. Since LCBs cover different regions on multiple genomes, there is one entry for each genome-LCB pair. The set of these pairs that are created by

a single Mauve LCB are linked together by a common *lcb_set* field. The genbank files are loaded into the *genes* table, which contains all of the genes on each genome. The *ortholog_set* field in this table is set as the program runs and specifies an identifier for the gene's predicted ortholog family. The all-vs-all BLAST results outputted by blastallvall.py are parsed into the *blastall* table, which lists all blast hits returned and their corresponding e-value. From these three tables, the *genelcbs* table is populated, which contains an entry for every gene and LCB pair that exhibit an *overlaps* relationship. Thus, if a gene overlaps an LCB on one or both sides, there is an entry matching those features. It is thus possible for a gene to have multiple entries in this table, as a gene can span across multiple LCBs

Table 1: SQL Data Tables: SQL tables and fields used to store genome and gene data.

Table Name	Database fields
lcbs	genome, [start, end], strand, lcb_set
genes	accession, gi, genome, position, [start, end], strand, ortholog_set
blastall	query, result, evalue
genelcbs	Accession, gi, genome, position, strand, lcb_set

Algorithm overview

After the data is uploaded into the database, the main algorithm of the program is applied to this dataset in order to identify ortholog families. In summary, the program consists of a number of iterations of a general *seed and grow* algorithm that is performed on all possible pairs of input genomes simultaneously. Each iteration is defined by the method of generating the seeds used to start that round. Each seed is just a pair of genes, one on each genome in the current pair, that are predicted to be orthologous because of exceptional support by synteny and/or sequence similarity. Each seed is the start of a new ortholog block, the basic element that will be expanded in the growth phase. Each block spans genes on a pair of genomes and represents, at the high level, orthologous gene pairs that are in close proximity on both genomes. In the growth phase, these blocks are expanded by trying to incorporate nearby pairs of similar genes, as these pairs are now syntenically supported by their proximity with the block. In both the seed and growth stages, only genes which have not already been paired with a gene in the other genome are considered. This pair-wise consistency is a weak version of the expectation that an ortholog family contains just one gene on each genome. An additional phase, *weak family completion*, was added to strengthen this consistency both after the seed and after the growth phases, in which additional implied edges were added to increase consistency.

Seeds

The seed phase is implemented in two stages. First, a broad set of pairs is generated by a seed algorithm based solely on the data in the database. Second, this returned set of pairs is filtered. The remaining pairs are then predicted to be orthologous.

Two distinct seeding algorithms were used, and both are illustrated in Figure 1. The first considers a pair of genes that are in the same Mauve LCB and match each other with an e-value below *parallelLCBThresh* (default 1e-5). This takes into account both the synteny implied by the Mauve LCB and the similarity of the BLAST e-value. The second algorithm ignores synteny and instead focuses on getting very similar proteins by predicting as orthologs all gene pairs with

reciprocal best BLAST hits with both e-values below *blastRBHThresh* (default 1e-30). The first seeding algorithm was split over three seed stages, each with a less-restrictive threshold (1e-65, 1e-35, and finally 1e-5), and the second RBH seeding algorithm was used last as shown below. This progressive relaxation of constraints was employed in order to ensure that the most likely ortholog pairs get classified first.

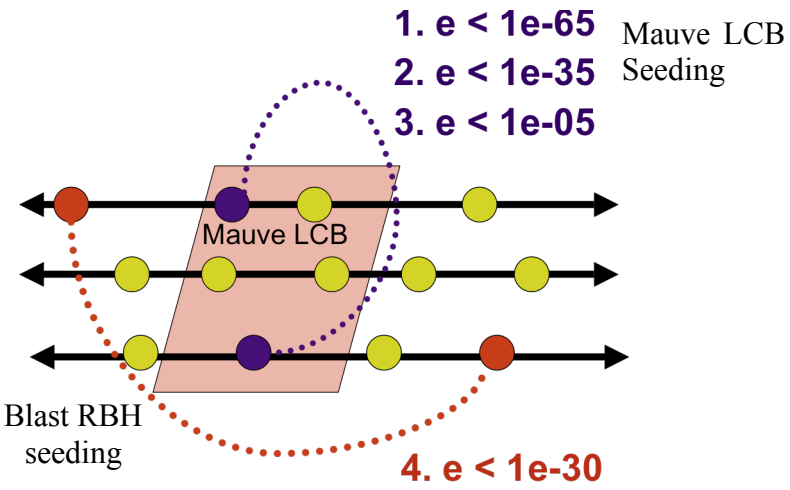


Figure 1: AutoOrthoGen Seeding algorithms, showing the three stages of the LCB and Blast synteny seeding, and then the Blast reciprocal best hit seeding.

The filtering stage takes the predicted ortholog pairs and keeps only those pairs that do not cause a gene on one genome to be paired with more than one gene per genome. First, all pairs are removed that contain a gene on one genome already paired with a gene on the other genome. Second, optionally, the set of seed pairs is analyzed to identify situations in which multiple seeds contain the same gene. If such a situation exists and one seed pair is a significantly more similar than any other (difference in e-values is greater than *evalueSeparationThresh*), the best seed is added and all conflicting seeds are removed. If the conflicting seed pairs are similar in quality then all seeds with the duplicate gene are ignored. These conflicting pairs may be added through block growth or at later stages in the algorithm. After this stage, the seed pairs that remain are predicted to be orthologous and each becomes the start of an ortholog block.

Growth

The ortholog blocks are then grown in both directions, giving a synteny bonus to nearby genes because of their proximity to the predicted orthologs in the block. Further, this bonus degrades with increasing distance from the block by dividing *nearLCBThresh* (default 1e-20) by *adjacencyDecay* (default 100) for each gene out. In order to prevent worse gene pairs being classified as orthologs before better pairs, this growth expands iteratively. First, all blocks on all pairs of genomes try to expand by one gene. At this time, if two blocks are adjacent on both genomes, and have compatible orientations, they are merged together. Then, all blocks try to expand by up to two genes, etc, up to the specified *maxDistance* (default 5). If a block is able to incorporate a new gene pair at some distance, the block immediately restarts iteratively growing from a distance of one up to the current distance.

Block orientation is set to either None, Syn, or Anti, and is representative of the relative orientations of the orthologous genes on the two genomes, as shown in Figure 2. If the orientation is None, it is not yet known what this orientation is and blocks expand in all possible configurations. For Syn and Anti orientations, the growth directions are limited, as shown in the diagram below. Finally, as with most growth, genes are only added if they are not already paired to another gene on the other genome.

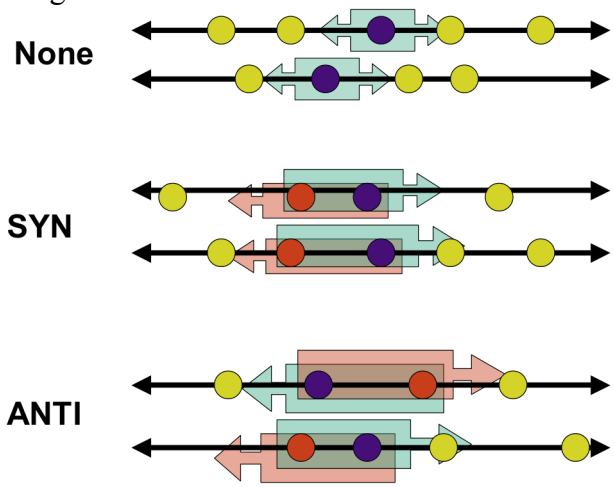


Figure 2: Block orientations: can be none, syn or anti with respect to each other.

Weak family completion

After seeding and after growing, an additional stage was added to synchronize the pair-wise predictions over all genomes to approximate set-wise consistency. It is possible to generate ortholog family predictions that are genome pair-wise consistent, but set-wise inconsistent. Such an example of a strong consistency is shown in Figure 3. In order to help abate this problem, weak conflicts (sets of ortholog pair edges that do not form cliques because of missing edges) are resolved by adding the missing edges necessary to complete the clique. Edges are only added if it does not, in itself, create a conflict. This preemptive step proved useful in avoiding strong conflicts before they occur, rather than trying to disentangle them after the fact. Weak conflict completion is performed by default, but can be turned off with a user flag.

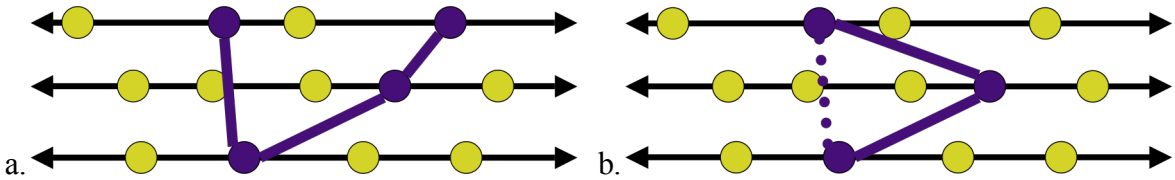


Figure 3: A pair-wise consistent, set-wise inconsistent "strong" conflict (a) and a "weak" conflict with the missing edge that will be added dashed (b).

Testing

An algorithm that claims to find orthologs and categorize genes must be reliable and accurate, or it may do more harm than good to the researchers who depend on its results. With this in mind, several methods were developed for testing the AOG algorithm, including tests for

expected behavior ("does it do exactly what we say it does?") , and tests for its efficacy ("does this behavior produce the desired results?"). These methods are described below.

Test Database:

Artificial test cases were defined for finding ortholog sets across four genomes. The method was to simply draw pictures of the test cases (see Appendix I) and then to insert corresponding data (genes, genomes, LCB's, and blast scores) into a test database. AOG was run on the test data, printing out all ortholog sets found after each stage of execution, to see if it produced the expected orthologs at the expected stages. This was a useful tool while debugging, and it now provides evidence that the algorithm behaves as specified in the algorithm description. Specifically:

- It correctly finds both lcb seeds and reciprocal best hit seeds.
- It grows to find orthologs on either side within a given maximum distance.
- It (initially) ignores ortholog seeds that are not one-to-one (i.e. seeds that match one gene from genome X to more than one gene on genome Y).
- When possible (when there are nearby orthologs to use as reference), it separates a "clump" of genes with identical blast scores into separate orthologs by synteny.
- For "clumps" that cannot be resolved via synteny, it groups the genes into a single, conflicted, ortholog set.
- It properly handles genes that span more than one LCB.
- It properly identifies strong and weak conflicts.

Of course, this set of test cases is necessarily incomplete. However, it provides reasonable assurance that the algorithm is behaving as expected, and it is easily extended to include additional test cases as necessary.

"Strawman" Algorithm:

A very simple algorithm was developed that finds ortholog sets by considering any gene pair with a blast score better than 1e-5 to be orthologous, regardless of synteny or conflicts with other pairs. It then groups these pairs into multi-genome sets via tree search (just like AutoOrthoGen does). Thus, it is expected to be highly sensitive to orthologs – missing very few or none of them – but also to produce a high number of false positives and much "clumping" of genes into large groups.

The "Strawman" serves as a very crude competitior for the AutoOrthoGen algorithm, and can be used as a control. Its other purpose is to demonstrate that genes with a high probability of being unique are not being falsely added into orthologous sets. In other words, since the Strawman defines orthology so loosely, if any gene survives its clumping process as "unique," then the AutoOrthoGen algorithm should also label it as unique. Similarly, if any ortholog set is labeled as "no-conflict" by the Strawman, then it should also be labeled as such by AutoOrthoGen. The results of the Strawman testing are discussed in the "Testing Results" section below.

Comparison to an Existing Algorithm:

An ideal test for an ortholog-finding algorithm would be comparison of its results to those of a perfect algorithm. Unfortunately, no such algorithm exists. Therefore AOG was compared to the

best algorithm that could be found – the "ATGC" (Alignable Tight Genomic Clusters) algorithm (http://atgc.lbl.gov/atgc/). This algorithm, like AOG, uses both gene similarity and synteny to find ortholog pairs, and then links them together into ortholog sets, as does AOG. It, like AOG, uses BLASTP to determine geneT similarity, but excludes all pairs that are not reciprocal best hits (RBHs), and includes all pairs that are, regardless of e-value. It then filters out any RBH that does not have at least four other RBH's in its immediate vicinity (defined as a slidable window over the genome pair that is seven genes wide). A downside to this algorithm (both for comparison purposes and in general) is that it outputs only those ortholog sets that span all the given genomes. Also, the algorithm is not available as a tool to run on genomes of one's choice. Instead, its results for selected genomes are available in an on-line database.

In spite of these shortcomings, it is a good algorithm for AOG testing purposes because it compares closely related species, unlike some other algorithms that are available, and because it considers both synteny and similarity. The results of the ATGC comparison testing are discussed in the "Testing Results" section below.

Testing on Genomes Created By Simulated Evolution:

Just as good as a "perfect" algorithm would be access to unlimited datasets with previously identified orthologs – for example, groups of closely related genomes with manually curated ortholog sets. A search of the internet failed to discover even one such dataset. However, such datasets are generated by the "Evolver" program developed by students at the University of Washington in the 2009 Computational Biology Capstone. This program simulates evolution and publishes sets of related genomes along with lists of their ortholog sets. These ortholog sets should be reliable because the Evolver program keeps track of gene ancestry as it goes. Thus the orthologs it reports are guaranteed, at least in theory, to be correct. (In practice, there are some possible pitfalls which are discussed in the "testing results" section.)

Ideally, AOG would be tested against multiple datasets generated from the Evolver program. However, the Evolver program was not producing usable GenBank files until recently, and so far has produced only one usable dataset of four genomes. (Before yesterday the GenBank files were always either unreadable by Mauve, or they included non-unique gene identifiers. Both precluded their use for testing AOG.) The results of the AOG test on this one dataset are discussed in the "testing results" section.

Adapting Standard Statistical Measures to Ortholog Finding:

To evaluate the results of the tests listed above it was necessary to develop a method for measuring the effectiveness of an ortholog-finding algorithm. The basic problem was to compare a list of "true" ortholog sets to a list of "predicted" ortholog sets and give numerical measure to the sensitivity and positive predictive value of the latter. However, the standard statistical method of counting true positives (TP), false positives (FP), and false negatives (FN) was confounded by the complexity of comparing sets to sets instead of discrete items to discrete items.

For example, if a true ortholog set consists of genes {A, B, C, D}, and the algorithm predicts two ortholog sets of {A, B} and {C, D}, how many true positives and false negatives are counted? Clearly, this result is not as good as finding the complete set, but better than missing it

completely. (There is a converse problem when, for example, the true ortholog sets are {A, B} and {C, D} but the algorithm lumps them together as {A, B, C, D}.)

Thus, the method of measurement must consider both the genes that are found and the way that they are grouped. Furthermore, sensitivity is meant to measure the ratio of the correctly predicted (TP) to the actual (TP + FN), and positive predictive value is meant to measure the ratio of the correctly predicted (TP) to the total predicted (TP + FP). Therefore, it is also important that the developed method preserves a consistent relationship between (TP + FN) and the actual ortholog sets and between (TP + FP) and the predicted ortholog sets.

The method that was developed counts TP's, FP's, and FN's as follows:

1. For every true ortholog family, if x of its genes occur together in some predicted ortholog family, (x-1) is added to TP. (If another y of its genes occur together in a second predicted ortholog family, another (y-1) is added to TP, etc.)
2. For every true ortholog family, if its genes are split across s different predicted ortholog families, (s-1) is added to FN. (s includes genes missing altogether from any predicted ortholog family. These can be thought of as being predicted to be in a family of one.)
3. For every predicted ortholog family, if its genes are joined from j different true ortholog families, (j-1) is added to FP. (j includes genes missing altogether from any true ortholog family.)

This method results in the following relationships between TP, FP, FN, true sets, and predicted sets:

1. For every true ortholog family with g members, the contribution to TP + FN is (g-1).
2. For every predicted ortholog family with g members, the contribution to TP + FP is (g-1).
3. Therefore TP + FN is always equal to the total number of genes in true ortholog families minus the number of such families, and TP + FP is always equal to the total number of genes in predicted ortholog families minus the number of such families.

Thus the method meets the goals of giving a meaningful definition to TP, FP, and FN in the context of comparing ortholog sets, while maintaining a consistent relationship between those counts and the true and predicted sets.

Results

Details on the datasets used:

1. The "Legionella" test set included the following four strains of Legionella:
   Legionella pneumophila subsp. pneumophila str. Philadelphia 1 (NC_002942)
   Legionella pneumophila str. Paris (NC_006368)
   Legionella pneumophila str. Lens (NC_006369)
   Legionella pneumophila str. Corby (NC_009494)

2. The "Strep" test set includes the following four strains of Streptococcus:
   Streptococcus agalactiae 2603V/R (NC_004116)
   Streptococcus agalactiae A909 (NC_007432)

Streptococcus agalactiae NEM316 (NC_ 004368)
Streptococcus pneumoniae TIGR4 (NC_003028)

3. The "Evolver" test set included four genomes derived by simulated evolution from Acaryochloris marina MBIC11017 plasmid pREB1, applying a mutation rate of 5%.

Running both AutoOrthoGen and Strawman on the ATGC Results for Legionella pneumophila:

ATGC results are used as the "true" ortholog sets, and for both Strawman and AOG only the ortholog sets that span all 4 genomes are considered (because ATGC does not output ortholog sets that do not).

Table 2: Results from Strawman evaluation on Legionella pneumophila

True orthologs:	9697 genes in 2424 families
Total genes in predicted ortholog sets:	11289 genes in
True positives:	7228
False negatives:	45
False positives:	1555
Sensitivity (TP / (TP + FN)):	0.994
False Omission Rate (FN / (TP + FN)):	0.006
Positive Predictive Value (TP / (TP + FP)):	0.823
False Discovery Rate (FP / (TP + FP)):	0.177

Table 3: Results from Strawman evaluation on Legionella pneumophila

True orthologs:	9697 genes in 2424 families
Predicted orthologs:	9881 genes in 2460 families
True positives:	7242
False negatives:	31
False positives:	179
Sensitivity (TP / (TP + FN)):	0.996
False Omission Rate (FN / (TP + FN)):	0
Positive Predictive Value (TP / (TP + FP)):	0.976
False Discovery Rate (FP / (TP + FP)):	0.024

Running both AutoOrthoGen and Strawman on Evolver data:

Table 4: Results from Strawman evaluation on Evolver data

True orthologs:	1498 genes in 391 families
Predicted orthologs:	1507 genes in 306 families
True positives:	1091
False negatives:	16
False positives:	110
Sensitivity (TP / (TP + FN)):	0.986
False Omission Rate (FN / (TP + FN)):	0.014
Positive Predictive Value (TP / (TP + FP)):	0.908
False Discovery Rate (FP / (TP + FP)):	0.092

Table 5: Results from AutoOrthoGen evaluation on Evolver data

True orthologs:	1498 genes in 391 families
Predicted orthologs:	1486 genes in 392 families
True positives:	1087
False negatives:	20
False positives:	7
Sensitivity (TP / (TP + FN)):	0.982
False Omission Rate (FN / (TP + FN)):	0.018
Positive Predictive Value (TP / (TP + FP)):	0.994
False Discovery Rate (FP / (TP + FP)):	0.006

Genomic Analysis Results:

Considering the accumulated evidence for AOG's efficacy, the data it produces can be reported here with some reasonable confidence in its validity. AOG was applied to two genome sets – the Legionella set and the Streptococcus set described at the top of this section. An example of AOG's output for a single genome is shown in Table 6: Gene info for: NC_006368 (Legionella pneumophila str. Paris). The full summaries produced by AOG for both genome sets are shown in Appendix II and III.

Table 6: Gene info for: NC_006368 (Legionella pneumophila str. Paris)

Total genes:	3027
Total unique genes:	103
Total orphaned homologs:	67
Total genes in families:	2857
Total 'common' genes (number of genes with orthologs on all other genomes):	2469
Total 'missing' genes (number of ortholog families that include all genomes except this one):	26

Discussion

Legionella pneumophila:

As expected, Strawman shows a very high sensitivity and a relatively low positive predictive value. As hoped, AOG appears to provide much better information than Strawman with respect to false positives. Also as hoped, AOG does not miss any of the unique genes found by Strawman. Surprisingly AOG also shows slightly better sensitivity than Strawman. This appears to be explained by a slight bug in the implementation of the Strawman algorithm: it restricts orthology to those cases where the e-value is better than 1e-5 in both directions (i.e. both GeneA blast GeneB < 1e-5 and GeneB blast GeneA < 1e-5). This explains how it could miss some orthologs found by AOG, since AOG requires only that the e-value be less than 1e-5 in one direction, if both genes are in the same LCB.

AOG also performed very well here with respect to ATGC, with a sensitivity score of 0.996 and a predictive value of 0.976. Also, it should be noted that some of the "false positives" found by AOG may in fact be real orthologs that were missed by ATGC, since AOG uses a more sophisticated method to determine synteny and does not automatically throw out gene pairs that are not reciprocal best hits, as ATGC does.

Evolver:

Again it can be seen that AOG produces many fewer false positives than Strawman, while nearly matching it for sensitivity. Even more importantly, as discussed in the section on testing methods, the Evolver ortholog set is very likely to be accurate, given the method of its production. Therefore, this test gives very strong evidence for the efficacy of the AOG algorithm.

Still, caution must be exercised in interpreting these results. It is after all, only a single set of data. These results will need to be replicated on multiple sets, before any firm conclusions may be drawn. In addition, the Evolver program is very new and has not yet been subjected to peer review. It is possible that it is not as good of a test as one would hope. For example, it may produce less differentiation among species than is found in nature, thus making it an inadequate challenge for AOG.

On the other hand, it is also worth pointing out that Evolver may over-count false negatives, because it begins its simulation from a real organism, and it does not remove paralogs first (or account for them in any way). Therefore, there may be orthologs present in the evolved genomes of which Evolver is unaware.

Genome Analysis

In addition to finding ortholog sets, AOG classifies and counts genes that are "unique" (there are no orthologs in this genome set), "common" (there is at least one ortholog on every other genome in the set), "missing" (there is at least one ortholog on every genome except this one), and "orphan" (there are no orthologs in this genome set, but the gene is similar to at least one other gene in the set, i.e. has a BLASTP score below a certain threshold). The category counts produced by AOG for both Legionella and Streptococcus are consistent with

expectations. Unique genes comprise only 4% of the Legionella genes but 10% of the Strep genes, which is unsurprising because the Legionella genomes are all closely related, while the Strep group includes three agalactiae strains and one pneumoniae. (The pneumoniae genome has 26% unique genes, skewing the average.) Even more dramatic, the pneumoniae has 28% missing genes while the other Streptococcus genomes average just 0.6%.

Future Work

After the first seed and blocks are created, each block is grown using the results from the next seed. However, this is prone to error especially in situations when there are new seeds on both sides of the block set all with the same relative blast e-value scores. On the other hand, genes themselves have a directionality associated with them. This directionality could be used to determine if the blocks are in syn or anti orientation, and then the blocks could be grown using this directionality.

Currently, the algorithm allows for adding additional seeds based on a new threshold. However, the same could be applied for the maximum growth distance and other parameters. Adding functionality to add new seeds based on a new threshold, new maximum growth distance, and other parameters would not only provide additional flexibility, but could potentially produce even better results.

The most difficult part of the process is to convert the pair-wise blast results into some set form. The current solution is to use pair-wise seeds and then form sets based on those seeds and then add additional seeds to the sets. However, this solution produces the weak and strong conflicts defined above. Another solution would be to form sets based on the blast results, and then seed and grow using orthologous sets defined by LCBs and blast.

The latest data from the Evolver project includes an analysis of AOG's performance on genome sets derived from different percentages of mutation. Unsurprisingly, this data shows that AOG's accuracy declines as the percentage of mutation increases. Future work should include an attempt to measure the evolutionary distance (for example, through global alignment) between each pair of genomes in a set and use that distance to customize the parameters used by AOG to find orthologs for that genome pair. For example, increased evolutionary distance would suggest using a lower e-value threshold for both seeding and growing.

Conclusions

The AutoOrthoGen algorithm is successful at identifying orthologous sets of genes and unique genes among a family of genomes. It uses both gene similarity and synteny to find orthologous genes sets in multiple prokaryotic genomes. It uses a "tiered" approach (removing the most likely orthologous genes in the earlier rounds) which gives preference to gene-pairs with the greatest evidence for orthology. To measure similarity, it uses the well-established BLASTP tool, and for synteny it exploits the Mauve alignment program, which allows it to find synteny even in the presence of radical genomic rearrangement.

The result is a solution with a very high sensitivity and positive predictive value of 99%, and very low false discovery rates and false omission rates of less than 2%. AutoOrthoGen is a highly accurate solution for finding orthologous genes and for identifying unique, common, and

missing genes. This may be of substantial benefit to researchers in microbiology, as they seek to use previous knowledge of related genomes to better understand new ones.

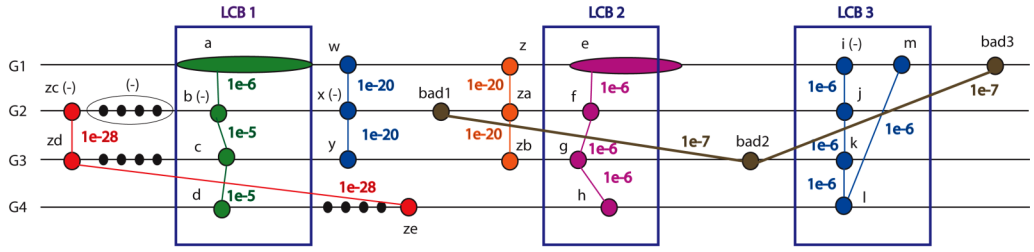
References

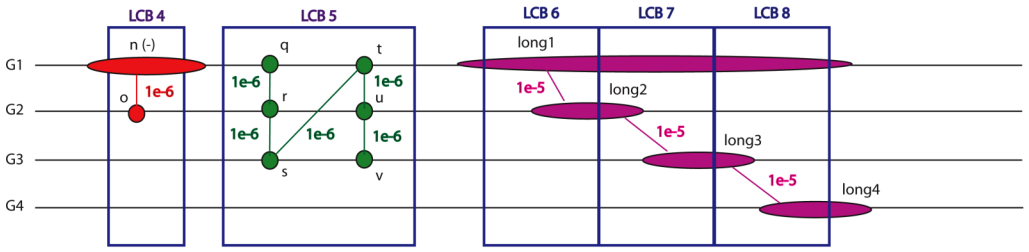
[1] Aaron C.E. Darling, Bob Mau, Frederick R. Blatter, Nicole T. Perna. "Mauve: multiple alignment of conserved genomic sequence with rearrangements" *Genome Research*. 2004; 14(7):1394-1403

[2] Herve Tettelin, Vega Masignani, Michael J. Cieslewicz, et al. "Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae: Implications for the microbial 'pan-genome'."*Proceedings of the National Academy of Science USA* 2005; 102:13950-13955.

[3] Novichkov PS, Ratnere I, Wolf YI, Koonin EV, Dubchak I. "ATGC: a database of orthologous genes from closely related prokaryotic genomes and a research platform for microevolution of prokaryotes" *Nucleic Acids Res.* 2009; 37(Database issue): D448-54

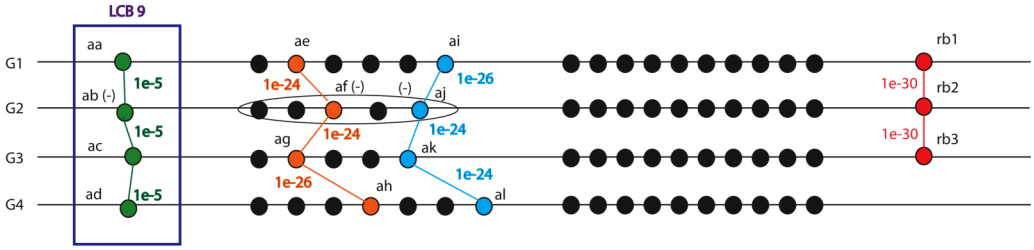
Appendix I: Test Data

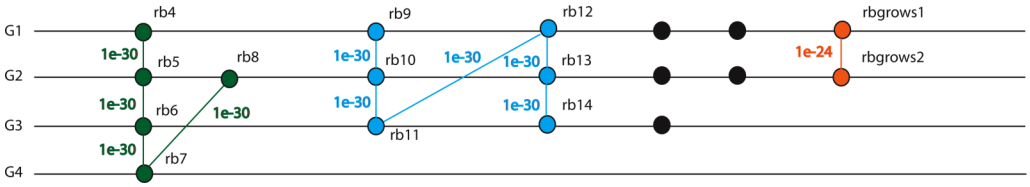




Requires the following settings:
parallelLCBThresh = 1e-5
nearLCBThresh = 1e-20
adjacencyDecay = 100
blastRBHThresh = 1e-30
maxDistance = 5

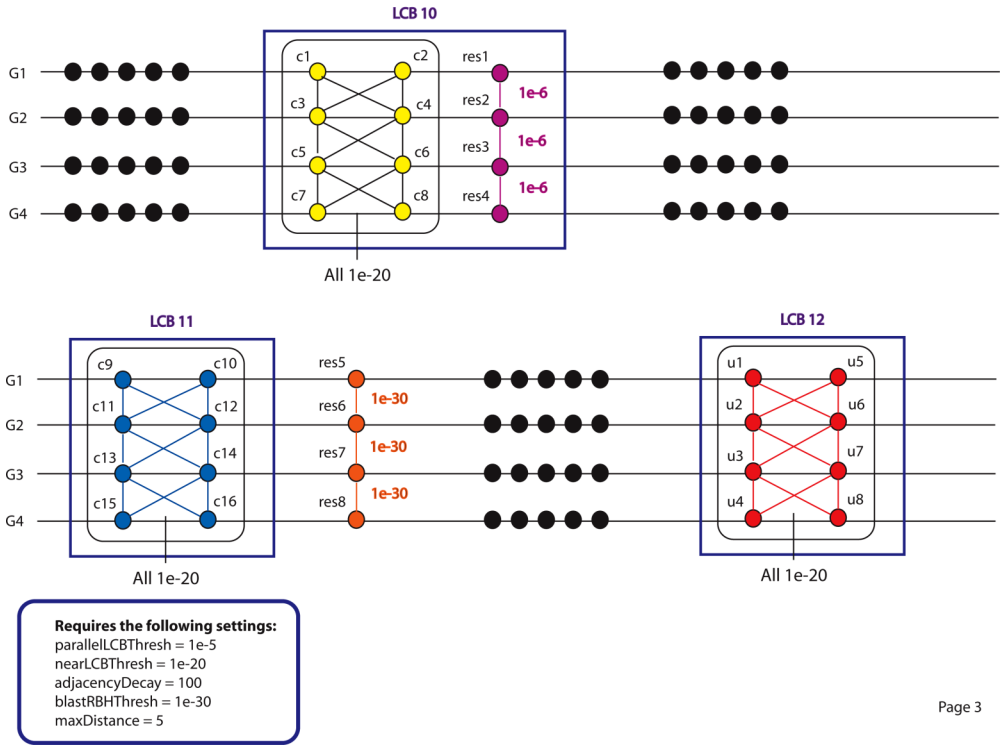
Page 1





Requires the following settings:
parallelLCBThresh = 1e-5
nearLCBThresh = 1e-20
adjacencyDecay = 100
blastRBHThresh = 1e-30
maxDistance = 5

Page 2



After expanding the LCB-based seed, we should have:

Strong conflicts:
[i, j, k, l, m]
[q, r, s, t, u, v]

Weak conflicts:
[a, b, c, d]
[aa, ab, ac, ad]
[ae, af, ag, ah]
[ai, aj, ak, al]
[c1, c3, c5, c7]
[c2, c4, c6, c8]
[e, f, g, h]
[long1, long2, long3, long4]
[res1, res2, res3, res4]
[w, x, y]
[z, za, zb]
[zc, zd, ze]

No conflicts:
[n, o]

After expanding the RBH Seed, we should add:

Strong conflicts:
[rb10, rb11, rb12, rb13, rb14, rb9]
[rb4, rb5, rb6, rb7, rb8]
[u1, u2, u3, u4, u5, u6, u7, u8]

Weak conflicts:
[c10, c12, c14, c16]
[c11, c13, c15, c9]
[rb1, rb2, rb3]
[res5, res6, res7, res8]

No conflicts:
[rbgrows1, rbgrows2]

Appendix II: Legionella SummaryOut.log

Debug Info:

Number of empty ortholog lists: 1 (non-zero indicates problem)

Summary Info on Ortholog Families:

Total number of families: 2972
Total weak conflict families: 0
Total strong conflict families: 30
Total no-conflict families: 2942
Total families that span all 4 genomes (includes conflict families): 2460

Summary Info on Genes:

Totals over all genomes:
Total genes: 12053
Total unique genes: 519
Total orphaned homologs: 338
Total genes in families: 11196
Total 'common' genes (number of genes with orthologs on all other genomes): 9881
Total 'missing' genes (number of ortholog families that include all but one genome): 274

Gene info for: NC_002942
Total genes: 2942
Total unique genes: 118
Total orphaned homologs: 98
Total genes in families: 2726
Total 'common' genes (number of genes with orthologs on all other genomes): 2470
Total 'missing' genes (number of ortholog families that include all genomes except this one): 105

Gene info for: NC_006368
Total genes: 3027
Total unique genes: 103
Total orphaned homologs: 67
Total genes in families: 2857
Total 'common' genes (number of genes with orthologs on all other genomes): 2469
Total 'missing' genes (number of ortholog families that include all genomes except this one): 26

Gene info for: NC_006369
Total genes: 2878
Total unique genes: 98
Total orphaned homologs: 53
Total genes in families: 2727
Total 'common' genes (number of genes with orthologs on all

other genomes): 2466
Total 'missing' genes (number of ortholog families that include all genomes except this one): 102

Gene info for: NC_009494
Total genes: 3206
Total unique genes: 200
Total orphaned homologs: 120
Total genes in families: 2886
Total 'common' genes (number of genes with orthologs on all other genomes): 2476
Total 'missing' genes (number of ortholog families that include all genomes except this one): 41

Appendix III: Streptococcus SummaryOut.log

Debug Info:

Number of empty ortholog lists: 0 (non-zero indicates problem)

Summary Info on Ortholog Families:

Total number of families: 1930
Total weak conflict families: 0
Total strong conflict families: 13
Total no-conflict families: 1917
Total families that span all 4 genomes (includes conflict families): 1111

Summary Info on Genes:

Totals over all genomes:
Total genes: 8317
Total unique genes: 865
Total orphaned homologs: 726
Total genes in families: 6726
Total 'common' genes (number of genes with orthologs on all other genomes): 4461
Total 'missing' genes (number of ortholog families that include all but one genome): 622

Gene info for: NC_003028 (Streptococcus pneumoniae)
Total genes: 2104
Total unique genes: 550
Total orphaned homologs: 366
Total genes in families: 1188
Total 'common' genes (number of genes with orthologs on all other genomes): 1115
Total 'missing' genes (number of ortholog families that include all genomes except this one): 583

Gene info for: NC_004116
Total genes: 2124
Total unique genes: 175

Total orphaned homologs: 92
Total genes in families: 1857
Total 'common' genes (number of genes with orthologs on all other genomes): 1117
Total 'missing' genes (number of ortholog families that include all genomes except this one): 14

Gene info for: NC_004368
Total genes: 2094
Total unique genes: 55
Total orphaned homologs: 200
Total genes in families: 1839
Total 'common' genes (number of genes with orthologs on all other genomes): 1116
Total 'missing' genes (number of ortholog families that include all genomes except this one): 1

Gene info for: NC_007432
Total genes: 1995
Total unique genes: 85
Total orphaned homologs: 68
Total genes in families: 1842
Total 'common' genes (number of genes with orthologs on all other genomes): 1113
Total 'missing' genes (number of ortholog families that include all genomes except this one): 24