Fall 2007

Software tools for comparing genomic sequence

Morel Henley
University of New Hampshire, Durham

SOFTWARE TOOLS FOR COMPARING GENOMIC SEQUENCE

BY

MOREL HENLEY
BS, University of New Hampshire, 2005

THESIS

Submitted to the University of New Hampshire
In Partial Fulfillment of
The Requirements for the Degree of

Master of Science
In
Computer Science

September, 2007

This thesis has been examined and approved.

Thesis Director, R. Daniel Bergeron,
Professor of Computer Science

Co-Thesis Director, Vaughn S. Cooper,
Assistant Professor of Microbiology

Philip J. Hatcher,
Professor of Computer Science

William K. Thomas,
Associate Professor of Biochemistry

July 20, 2007
Date

ACKNOWLEDGEMENTS

I would like to thank all members of my committee: Dan Bergeron, Vaughn Cooper, Phil Hatcher, and Kelley Thomas. The weekly meetings for the past year with Dan, Vaughn, and Phil have been essential in this collaborative project. Dan has been instrumental in his interest and insights into the project. None of this would have been possible without Vaughn's questions, interest, and knowledge about the *Burkholderia* genomes that were the driving force of the OPUS project. I would like to thank Phil for all his work and combined efforts on the project which allowed us to work off each other's research and build better tools as a result. I would also like to thank Kelley for his genomics knowledge and help with the DSNP project.

I would also like to thank everyone involved in the collaborative DSNP project. Especially, Abraham Tucker who has been not only the driving force in the project but also a genomics resource.

The UNH President's Excellence Fund provided me with funding for my time and effort on this research project. I would also like to thank everyone in the Thomas Lab for all their support. I also would like to acknowledge Charles Bancroft for all his preliminary work on Magenta and allowing me to build from his software.

Last but not least, I would like to thank my family and friends who have given me a tremendous amount of support. And a special thanks to Mike, for always being supportive for the majority of my college experience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

SOFTWARE TOOLS FOR COMPARING GENOMIC SEQUENCE

by

Morel Henley

University of New Hampshire, September, 2007

We describe three software tools related to research in comparative genomics, a growing research area that explores the variation within and between organisms. We developed a set of tools that explore sequence similarity and differences in genomes. Two of these tools are specifically aimed at examining DNA sequence data from two or more genomes:

- The Magenta's OPUS tool compares genomic sequences to identify shared or unique segments between closely related species. This tool looks for functional similarities and differences in genomic data by classifying sequences into groups based on genomic categories: *Orthologs, Paralogs,* and *Unique Sequence.*
- The DSNP tool looks at the nucleotide level to find single nucleotide polymorphisms (SNPs) within an individual. This program is a collection of existing and custom built tools to discover and analyze SNPs within the *Daphnia pulex* genome.
- The third tool supports a user evaluation of two different visualization techniques for comparing nucleotide or protein sequences.

CHAPTER 1

INTRODUCTION

Bioinformatics and Computational Biology use various interdisciplinary skills including mathematics and computer science to analyze biological information. One of the main areas of research for these fields is genomic studies. A genome encodes the genetic information of a particular cell or organism (Hartwell 2004). A very fundamental part of a genome is its DNA nucleotide sequence, a linear strand expressing the pattern of bases, **A**denine, **T**hymine, **G**uanine, and **C**ytosine, that codes for proteins and other information. Since DNA encodes a lot of what determines the phenotype and function of an organism, the study of its sequence can provide much insight. Genomic analysis is a fast growing field of study. The DNA sequence of organisms produces large amounts of data; with a growing number of genomes being sequenced, understanding what these genomes represent is an important research area in genomics.

Comparative genomics studies the similarities and differences among the genomes of closely related organisms and within a single organism. With a side-by-side comparison of multiple genomes, geneticists hope to determine the locations in genomes that cause functional variations. This thesis describes three different software tools that contribute to various aspects of comparative genomics exploration. *Magenta's OPUS* compares genomes from closely related organisms to find DNA sequences that are either

shared by the organisms or are unique in one organism. The *DSNP* tool detects single nucleotide polymorphisms (SNPs) in closely related genomes. We also describe an experiment that evaluates two different visualization techniques that are commonly used for comparing sequence data from two different genomes.

1.1 Background

Related nucleotide sequences can be grouped into various categories. Sequences that are shared due to common ancestry are classified as *homologs*. Homologs can be broken down into *orthologs* and *paralogs* based on the way they formed. Orthologs result from a speciation event; that is, a sequence corresponding to the two homologous sequences existed in an ancestor of both species (Hartwell 2004). Paralogs are the result of a sequence duplication event within a species (Hartwell 2004). Therefore, multiple matches of a sequence within a single genome are usually considered paralogous sequence.

Regions in genomes that are similar suggest a similar function. Likewise, differences or uniqueness of sequence may denote distinctive functions. The amount of variation can imply evolutionary relationships of the organisms, as well. Mutations can also aid in predicting evolutionary or phylogenic estimates of the species in comparison. A phylogeny is a representation of how a set of organisms have evolved with respect to the others. Diversity within an organism can also identify potential allelic or characteristic variations such as eye color or hair color.

1.2 Magenta's OPUS, A Genomic Variation Detection Tool

We developed a method for identifying and analyzing the various types of locations in a genome. Magenta's OPUS classifies different regions within and between

genomes into three fundamental categories: orthologs, paralogs, and unique sequences. This provides a flexible, integrated, bioinformatics tool that compares both raw and annotated genome sequence to identify a wide range of genetic variation. Its flexibility is derived from a database of BLAST hits that can be reused for a wide range of queries. We have developed simple algorithms to identify orthologs, paralogs, and unique sequences (OPUS) between multiple pairs of genomes. The utility of the OPUS toolkit is illustrated through a study of genomes from the *Burkholderia cepacia* complex. We hope that the data produced from this method will lead to interesting discoveries. This tool is discussed further in Chapter 2.

1.3 Evaluation Study of Data Visualizations for Genome Comparison

Visualizing the common sequence between species can be difficult. We focus on evaluating the effectiveness of current visualization methods for genomic comparisons. This heuristic evaluation looks at two accepted graphical methods for comparing nucleotide sequences. Scatter plots and parallel coordinate-like visuals have been used in genomics for identifying similarities in genetic code. Our evaluation focuses on determining the aspects of the two visualizations that are successful and those that need enhancements. Mauve (Darling, Mau et al. 2004) has a visualization tool that uses a technique resembling a parallel coordinate method for connecting orthologs. With a large volume of data, this visualization becomes cluttered and hard to view. We developed a scatter plot tool that uses concepts based on dot plots to display data consistent with that shown by Mauve. In our evaluation, we determined the advantages and disadvantages of each visualization. We have identified potential improvements for each, including applying an algorithm to help sort the data. We also wanted to introduce

the dot plot as an optional view in the Mauve software. The dot plot would provide the user with different information and would enhance the tool. More about the modifications and additions are found in Chapter 3.

1.4 DSNP: Detection of Single Nucleotide Polymorphisms (SNPs) in the *Daphnia pulex* Genome

We have been working on detecting SNPs or Single Nucleotide Polymorphisms within a eukaryotic organism. SNPs are the allelic variations that cause different phenotypes or characteristics within a species. They are responsible for variations such as eye or hair color. In order to identify single nucleotide polymorphisms (SNPs) within the *Daphnia pulex* genome, we developed a pipeline of analyses that uses the comparative assembly of whole genome shotgun reads (sequences used for shotgun sequencing) against reference scaffolds (portions of the genome that have been assembled) to conservatively estimate sites of true polymorphism. Our initial analyses have focused on The Chosen One (TCO), the strain selected for the *Daphnia* Genome Project. This process is discussed in Chapter 4.

CHAPTER 2

MAGENTA'S OPUS

2.1 Abstract

We describe a flexible, integrated, bioinformatics tool called *Magenta's OPUS* that compares both raw and annotated genome sequence to identify a wide range of genetic variation. Its flexibility is derived from a database of BLAST hits that can be reused for a wide range of queries. We have developed simple algorithms to identify potential Orthologous, Paralogous, and Unique Sequences (OPUS) between pairs of genomes. Different techniques such as Reciprocal BLAST and a method we refer to as the Lerat (Lerat, Daubin et al. 2003) method were applied to find homologous sequences. We also describe subcategories for sequences including *semi-paralogs* (a term we use to describe matching sequences between genomes that have a one-to-many relationship), *in-paralogs* (duplications after speciation), and *out-paralogs* (duplications before speciation). We also describe post-processing analysis called *OPUS Notes* to parse and annotate the data further. *OPUS Notes* is an OPUS extension postprocessor that annotates identified genes and gene pairs with COG (Tatusov, Fedorova et al. 2003) functional categories and estimates of the per-base average frequencies of synonymous and non-synonymous nucleotide substitutions (K_s and K_a). The utility of the *OPUS* toolkit is illustrated through a study of genomes from the *Burkholderia cepacia* complex.

2.2 Introduction

OPUS is a bioinformatics tool that compares both completely and partially sequenced genomes of closely related organisms to identify genomic variation. Its function is to sort homologous regions between pairs of genomes into regions that resemble *Orthologs* and *Paralogs,* as well as identifying non-homologous regions or *Unique Sequences.*

Our method for identifying and analyzing each of these types of genetic regions does not require that genes be defined or annotated *a priori.* This is often useful because the number of unfinished, incompletely annotated sequences outnumbers those that are closed and annotated (e.g., see http://genomesonline.org). Thus, without prior knowledge, we can discover important sources of genetic and evolutionary novelty before the genome is fully assembled. Although being able to extract information without prior knowledge of the genes is beneficial, our method can work with predicted gene information as well for a more focused analysis. The *OPUS Notes* utility developed by Philip Hatcher uses the *OPUS* output and predicted gene data to extract additional information on sequence evolution. We present analysis derived from both raw sequence and from predicted gene files.

The *OPUS* tool kit builds upon a number of existing software packages. We use *Magenta* (Bancroft 2006) derived from *Mauve* (Darling, Mau et al. 2004) as a base point to start our analysis. *Magenta* uses BLAST (Altschul, Gish et al. 1990), the standard Basic Local Alignment Search Tool, to find sequence similarity among the genomes. It stores these results in a database that can be reused for a variety of queries on the data. The *OPUS Notes* utility aligns selected matches in our database using CLUSTALW

(Thompson, Higgins et al. 1994), estimates the types and frequency of substitutions that differentiate sequences using portions of *GenomeHistory* (Conant and Wagner 2002), and accesses COG information (Tatusov, Fedorova et al. 2003) provided by the Integrated Microbial Genomes (IMG) website of the DOE Joint Genome Institute (JGI).

OPUS, like many methods, works to identify sets of similar sequences and explore various patterns within them. Sequence similarity is the central currency in the field of comparative genomics because highly similar sequences are likely derived from a common ancestor and may perform similar functions. Similarity due to shared descent is known as *homology*. Homologous sequences, in turn, can be subdivided into two major groups: *orthologs*, which differentiate following a splitting or speciation event of the host organism, and *paralogs*, which arise due to gene duplication events within a genome. Distinguishing homologs as either orthologs or paralogs can be a surprisingly challenging task because genes may simultaneously be both orthologs and paralogs depending on the scale of comparison. We have implemented algorithms for predicting categorizations of sequence similarity and difference.

Another major goal of comparing genomes is finding sequence unique or lost in a certain species (an absence of homology), which could imply a distinctive function or explain ecological novelty. Therefore, we designed our tool to try to identify regions from a given comparison between two genomes as potentially belonging to either of three major genomic categories: orthologs, paralogs, and unique sequences. We focus only on regions of close similarity and follow simple algorithms. The data produced from this analysis may shed light on the evolution of genes and genomes, and also provide inferences about the origin of functional novelty for each strain.

2.3 Background

2.3.1 Study organisms

Bacteria of the genus *Burkholderia* are ideal study subjects for our system. Microbial genomes are generally small in size, ranging from 500Kb to 10Mb, but also have very little noncoding sequence. Even with the small genomes (compared to eukaryotes), the amount of data generated by each analysis is still quite large.

Burkholderia are one of the most variable and least understood groups of potential pathogens, which currently foils most surveillance and therapeutics. Formerly classified as pseudomonads, *Burkholderia* are both functionally diverse and broadly distributed. They typically grow in water and around plant roots, but have also been found growing on silicon wafers and in nasal spray (Mahenthiralingam, Urban et al. 2005). Most species are not pathogenic for healthy livestock or humans, but two *Burkholderia* cause lethal human disease and are potential biological weapons (http://www.bt.cdc.gov/Agent). Other strains have shown considerable promise for bioremediation, as plant probiotics, and as pesticides (Parke and Gurian-Sherman 2001).

One species complex of *Burkholderia*, known as the *Burkholderia cepacia* complex (Bcc) is especially worthy of analysis by *Magenta/OPUS*. The eleven species comprising the Bcc are virtually identical in certain core genes that are used to delineate species (the Bcc are >99% identical in their 16S ribosomal RNA coding sequence), which suggests a very recent split (Cooper pers. comm.). However, isolates within these species may differ greatly in their overall genome content, their pathogenic potential, and their functionality (Cooper pers. comm.). Notably, some Bcc members frequently cause lethal pulmonary infections in persons with cystic fibrosis (CF) and in other

immunocompromised patients, while other close relatives are apparently benign. The mechanisms and forces that account for this rapid diversification are unclear (Cooper pers. comm.).

Burkholderia genomes are unusually large and variable for bacteria, composed of two to five circular chromosomes totaling five to nine megabases (Mahenthiralingam, Urban et al. 2005). More than forty complete genome sequences are now available within *Burkholderia* for our analyses, with representatives across a range of evolutionary scales. Thus, we can compare multiple genomes within the same species, genomes from among extremely closely related species, and between these clusters and more distant *Burkholderiacae* out-groups. For this analysis, we have chosen *Ralstonia eutropha* JMP134, which is an immediate relative of *Burkholderia* species, is well-studied, and has broad potential for bioremediation (Cooper pers. comm.). Our out-group genome provides a comparison for the types and magnitude of sequence similarity within and between the *Burkholderia* species.

Our goal is to identify, categorize and quantify the mutational processes that gave rise to variations among Bcc strains. We focus on four genomes from the same Bcc species, *B. cenocepacia*: *B. cenocepacia* AU1054, *B. cenocepacia* HI2424, *B. cenocepacia J2315* and *B. cenocepacia* PC184 (Figure 1). AU1054 and HI2424 are isolates of the same strain type (PHDC) and therefore are most closely related. HI2424 was isolated from the soil, while the other three were isolated from patients with CF. PC184 is the type isolate of the Midwest strain of *B. cenocepacia*. J2315 is the type isolate of the ET12 epidemic strain, which predominates in the UK and Canada. These major strain types (PHDC, ET12, and Midwest) account for a significant fraction of new

human infections, but appear to differ in their associated prognoses and differ in their ability to infect laboratory host organisms (Mahenthiralingam, Urban et al. 2005; Cooper pers. comm.). The underlying genetic determinants of these differences are largely unknown.

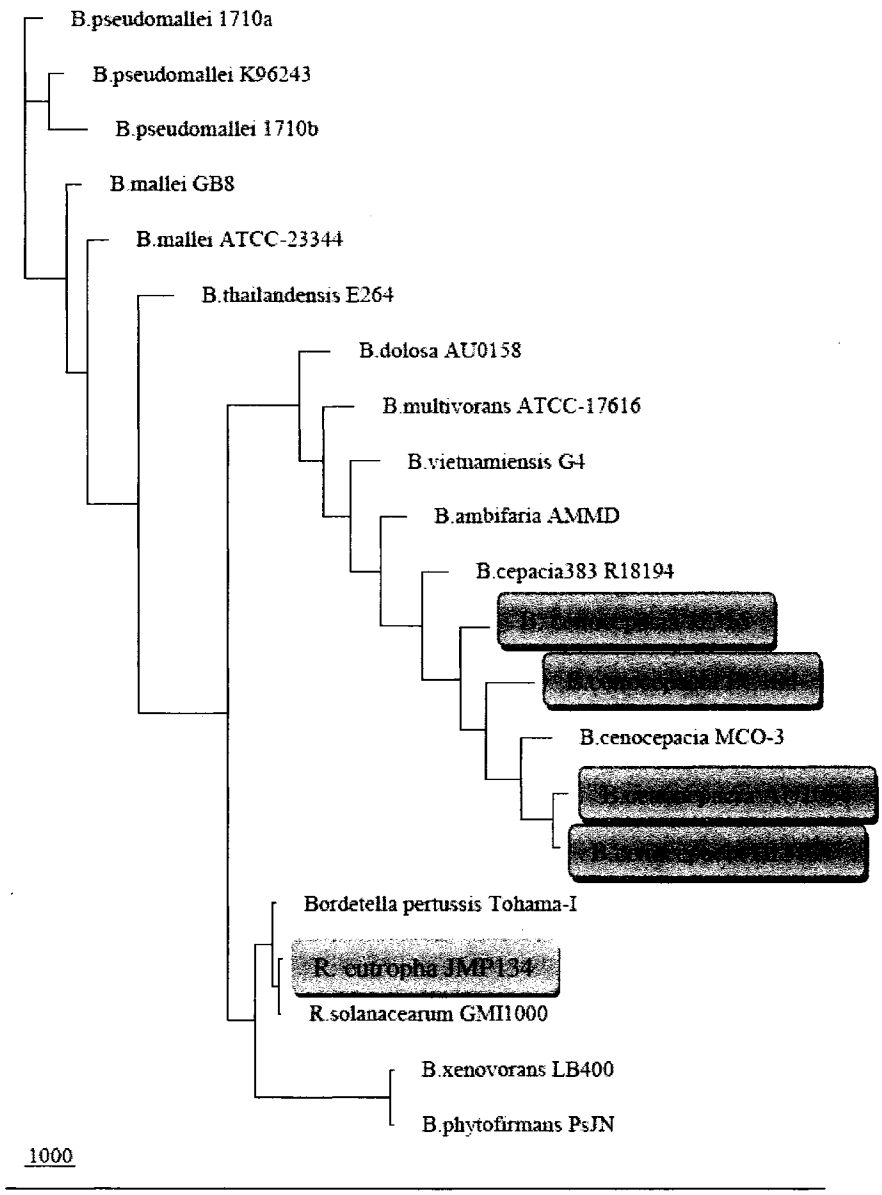


Figure 1: A Phylogenetic Tree of Burkholderia Species. This tree was produced using gene family presence/absence data (Hatcher, unpublished work). The data was generated by PARS and passed to *TreeView* to generate the tree. The blue blocks indicate the *B. cenocepacia* species we used in our study. The green box is the *Ralstonia* out-group genome chosen for comparison.

2.3.1.1 Data

We acquired genome sequences from sequencing center websites including: JGI (http://img.jgi.doe.gov), the Sanger Institute (http://www.sanger.ac.uk/), and the Broad Institute (http://www.broad.mit.edu/). Although our program uses both genomic sequence data and predicted gene information, the analyses presented here are mostly derived from predicted gene files.

2.3.1.2 Lateral Gene Transfer

One potential problem when working with bacterial genomes is that prokaryotes tend to exchange sequence information with other bacteria in their environment. The merging of genomic material into the new genome from this exchange is referred to as lateral gene transfer (LGT) or horizontal gene transfer (HGT) (Hartwell 2004; Lerat, Daubin et al. 2005). LGT incorporates an additional complication when evaluating bacteria because this phenomenon can be masked in many of the classification categories.

2.3.2 Genomic and Biological Background

2.3.2.1 Homology

Homology means evolutionary similarity, or similarity arising from common ancestry. However, homology should not be confused with sequence similarity. Sequence similarity can arise from different mutation processes in the genomes, making two sequences appear similar even though they developed from different ancestors. Also, small sequences can appear similar by random chance even though they are not actually homologous. These cases are few and therefore generally sequence similarity indicates homology with the exception of bacteria, who can also acquire common sequence from lateral gene transfer (LGT).

Revealing homology is valuable in genomic research because it suggests areas of functional similarity and distinctiveness, as well as providing direct evidence for the evolutionary relationships of organisms. Usually, homology is associated with shared or similar function especially when referring to genes, although this is not always the case, either.

Homology is typically separated into two basic categories: *orthology* and *paralogy*. However, the boundaries between these categories are fuzzy, especially when referencing sequence comparisons alone where phylogenetic tree construction and other biological analyses are required. The complications arise because ancestors can have duplicate gene copies that descended through speciation events.

2.3.2.2 Orthologs

Orthologs are similar sequences derived from speciation events. Similar sequences that are present only once in each genome under comparison are likely orthologs (we describe the other explanation, lateral gene transfer, above). Orthologs may also be similar to sequences in the same genome, but these are easily confused with paralogs. True orthologs can be used to help determine recent evolutionary relationships of organisms by evaluating mutations within these sequences. Orthologs also illustrate the operative genes that are common between the organisms. When trying to separate duplications in ancestors (orthologs) from duplication after speciation (paralogs), geneticists predict that the segments that have fewer deviations from each other is the most parsimonious evolutionary explanation of their mutations and therefore consider the segments true orthologs.

2.3.2.3 Paralogs

Paralogs are homologous sequences whose similarity results from gene duplication events rather than speciation events. Similar sequences that are found in multiple locations in a genome are putative paralogs. For example, if segments 'b1' and 'b2' in genome B are similar, then we say segments 'b1' and 'b2' are paralogous. If segment 'a' in genome A matches segments 'b1' and 'b2' in genome B, then we say segments 'b1' and 'b2' are paralogous to segment 'a' since we know that 'b1' and 'b2' are both similar to 'a'. This inequality in sequence copy number between genomes may either result from a duplication event in the target genome or from gene loss in the query genome. (Determining which of these scenarios occurred requires a secondary analysis with one or more allied, phylogenetically related genomes.) Either of these events provides useful inference into what functions the organisms have gained or lost over time.

Sonnhammer and Koonin have proposed nomenclature for different types of paralogs. *In-paralogs* are lineage-specific paralogous sequences resulting from duplication events since the last speciation event (Sonnhammer and Koonin 2002). *Out-paralogs* are paralogous sequences resulting from duplications that preceded the speciation event separating genomes (Figure 2). Both of these definitions require a model of the overall phylogeny of the genomes, which can be challenging with incompletely sequenced genomes or direct sequencing from the environment (metagenomics).

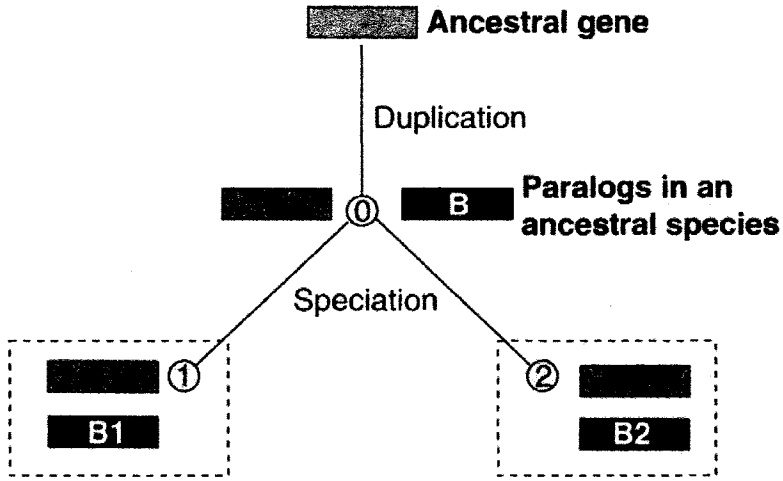


Figure 2: Koonin's (2001) pictorial representation of speciation and duplication events. In this image, the ancestral gene was duplicated before the speciation event, an example of an out-paralog (Koonin 2001).

We introduce a new term, *semi-paralog*, to refer to regions that are paralogous within one genome but that are orthologous to a single sequence in the other genome (Figure 3). Semi-paralogs could also be larger sets of homologous genes differing in their copy number between genomes. We hypothesize that these regions have been duplicated because they are functionally significant, implying an enhanced capacity in the genome bearing multiple copies. Further, since most duplicated sequence is eliminated over time (Lynch, O'Hely et al. 2001), persistence of a duplicate is exceptional and implies selective value.

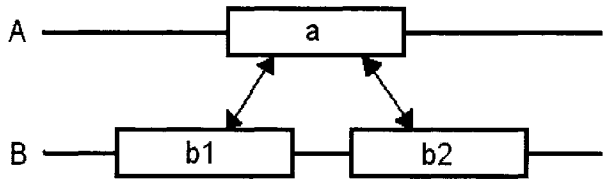


Figure 3: Example of a semi-paralog

2.3.2.4 Unique Sequence

Regions that lack sequence similarity (homology) between species are *unique sequence* and are caused by a loss in one of the genomes, or, in the case of prokaryotes, lateral gene transfer (LGT). Unique sequence can also be a result of simply rapid evolution. We are interested in the frequency and types of unique sequences because those that are due to LGT can be used to illustrate the rate of LGT among bacterial species (Lerat, Daubin et al. 2005).

2.4 Related work

2.4.1 OPUS-like programs

There have been numerous tools developed to identify similarities and differences in genome sequence data, including the identification of unique sequences (Pontius J.U 2003), homology (Schwartz, Zhang et al. 2000; Markowitz, Korzeniewski et al. 2006), orthologs (Remm, Storm et al. 2001; Bansal and Meyer 2002; Overbeek, Larsen et al. 2003; TIGR 2005; Thomson, Howard et al. 2006), and paralogs (Overbeek, Larsen et al. 2003; Haas, Delcher et al. 2004). Most of these have focused on comparing sequence data represented as protein data (i.e., genes)(Remm, Storm et al. 2001; Bansal and Meyer 2002; Overbeek, Larsen et al. 2003; Pontius J.U 2003; Haas, Delcher et al. 2004; TIGR 2005; Markowitz, Korzeniewski et al. 2006; Thomson, Howard et al. 2006).

Phylogenetic Profiler is typical of most of the homology detecting tools. It is part of the Integrated Microbial Genomes (IMG) toolkit provided by the DOE Joint Genome Institute and includes a large and powerful suite of tools for genome comparisons (Markowitz, Korzeniewski et al. 2006). *Phylogenetic Profiler* identifies predicted genes of a genome of interest that are homologous (or that lack homology) with other genomes.

This operation is based on a database of pre-computed (uni-directional) similarities between the genes of the source and target organism(s). However, this method is limited by its focus on coding sequence since it excludes regions of non-coding sequence from the analysis. In addition, *Phylogenetic Profiler* does not differentiate its output list of homologs into orthologs and paralogs, and only identifies any additional paralog copies in the target genome, not the complete collection of paralogous sequences.

2.4.2 Alignment and Assembly Programs

There are two basic types of alignment programs: 'local' alignment and 'global' alignment. BLAST is a very well known local alignment program (Altschul, Gish et al. 1990). The local alignment programs like BLAST use an algorithm for matching sequences and then extending them. Local alignments are good at identifying orthologous and paralogous sequence and are unaffected by rearrangements between the genomes (Dewey and Pachter 2006). However, they have a tendency to identify artificial similarity since small pieces can appear similar. 'Hierarchical' alignments use the output of pairwise local alignments to build multiple genome alignments (Dewey and Pachter 2006). These programs typically only identify orthologous alignments. *Mauve* (Darling, Mau et al. 2004) is an example of a common hierarchical program (see below). Many of these programs have visualizations to view the orthologous comparisons between genomes (Darling, Mau et al. 2004; Dubchak and Ryaboy 2006).

Global alignment programs are fast but less sensitive in identifying sequence similarity (Dewey and Pachter 2006). *MUMmer* is a common assembly program (Kurtz, Phillippy et al. 2004). Its algorithm uses suffix trees to find matches (Delcher, Kasif et al. 1999). This program is very fast but is less sensitive to the detection of matching

sequence. *MUMmer*, up until version 3.0, only found matches with a single copy of exact matches (eliminating paralogs and identifying them as incorrect orthologs by arbitrarily choosing one of the positions to place the segment). *Nucmer* and *Promer* were then added for identifying multiple exact matches. The algorithm used by these programs is nearly equivalent to that used for the BLAST application (Kurtz, Phillippy et al. 2004).

2.5 Tools Used by OPUS

2.5.1 BLAST

The *OPUS* toolkit and some other gene comparison tools use BLAST (Altschul, Gish et al. 1990) as the principal tool for determining sequence similarity. BLAST, basic local alignment search tool, compares either nucleotide or protein sequences with a defined reference database and returns sequences that match a threshold level of similarity. In this paper we focus exclusively on the *blastn* program, which compares nucleotide sequences, but we have also utilized *tblastx*, which compares nucleotide sequences translated in all six reading frames with all frames of the database. Since 'local' alignments have the ability of identifying both orthologous and paralogous sequence, it provides a good building tool for our *OPUS* toolkit.

2.5.2 Mauve/Magenta/OPUS

Mauve (Darling, Mau et al. 2004) aligns multiple genome segments or complete genomes and generates a visualization of the alignment (Figure 4). It is tolerant of major genomic changes like rearrangements or inversions, and it highlights these events by connecting blocks of orthologous sequences shared by the genomes. These blocks may nevertheless include paralogs shared by the aligned genomes. *Mauve* identifies these matching sections of nucleotide sequence as Multiple Maximally Unique Matches (multi-

MUMs). It then groups these multi-MUMs into closely related regions called Locally Collinear Blocks (LCBs) (Darling, Mau et al. 2004). These blocks are displayed on the screen with connecting lines to identify the shared sequence. We have modified and extended the *Mauve* software package by replacing the core method of sequence alignment based on multi-MUMs with BLAST (Bancroft 2006). This version of *Mauve*, referred to as *Magenta*, uses a reciprocal BLAST method to identify the homologs between genomes. Reciprocal BLAST means that we blast genome A as the query against genome B, the target, and genome B as the query against genome A, the target. Each comparison's results are then stored in a database table. The database incorporation provides fast access and reuse of the results. Here we report extensions to *Magenta* that add additional options for identifying and extracting regions of interest, such as *orthologs*, *paralogs*, and *unique sequences* (OPUS).

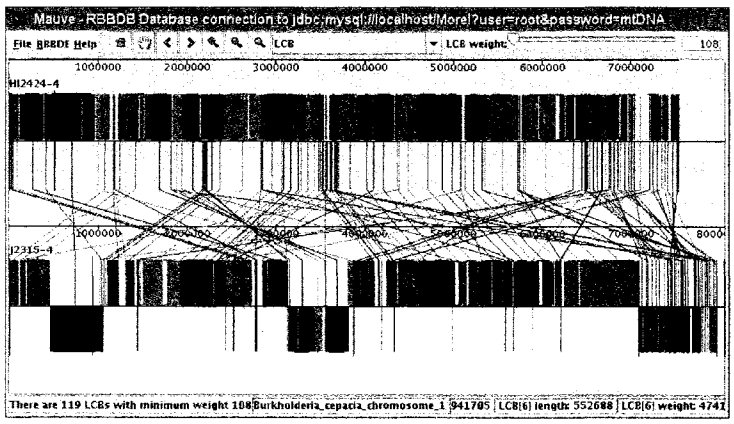


Figure 4: Mauve Visualization

2.5.3 OPUS Notes

OPUS Notes is a Perl program written by Hatcher that processes gene pairs identified by the *OPUS* post-processing of the *Magenta* database (such as semi-paralogs). It utilizes data previously downloaded from the Integrated Microbial Genomes (IMG) website of the DOE Joint Genome Institute (JGI). In particular it accesses the COG

(Tatusov, Fedorova et al. 2003) amino acid sequence and nucleotide sequence data for the genes of the genomes being studied. COG is an abbreviation for Clusters of Orthologous Groups (Tatusov, Fedorova et al. 2003), which are gene sets with similar function, also referred to as gene families with similar functions.

Each gene in a pair to be processed is identified by its genome name and its IMG gene object identifier. The script uses this information to find the COG for each gene in the pair and adds the COG identification information to the FASTA header for its amino acid sequence.

The script then computes K_a and K_s for each gene pair using the approach implemented by *GenomeHistory* (Conant and Wagner 2002). K_a and K_s are the estimated per-base average frequencies of non-synonymous (amino-acid altering) and synonymous (silent) nucleotide substitutions (Hartwell 2004). The ratio of synonymous and non-synonymous substitutions is used to estimate a rate of evolution in a sequence. First, *ClustalW* (Thompson, Higgins et al. 1994) is used to do a protein alignment. The protein alignment is then utilized by a tool distributed as part of GenomeHistory (algndna_new) to perform a nucleotide alignment. Finally, K_a and K_s are estimated by another tool distributed with GenomeHistory (like_pair_dist) that uses a maximum likelihood algorithm.

2.6 Our Approach

Our primary goal in this project was to categorize and evaluate similarities and differences in DNA sequence. We approach this problem by first trying to subdivide the sequences into the biological groups: homologs, orthologs, paralogs, and unique sequence. However, the lines between these groups are not always clear. Instead of pre-

defining a specific rule for distinguishing orthologs from paralogs, OPUS identifies a set of classifications describing various relationships for homologous sequences (Figure 5). As a post-processing step, a researcher can select the sets that are appropriate for the task.

Another goal of our project was to be able to run our program on genome-wide sequences as well as partial and unannotated genomes. Since a large portion of genomes are in the process of being sequenced the ability to do some analysis before their completion is beneficial. Since noncoding regulatory regions may also contribute to function of a genome, looking at variations among these regions is beneficial, as well.

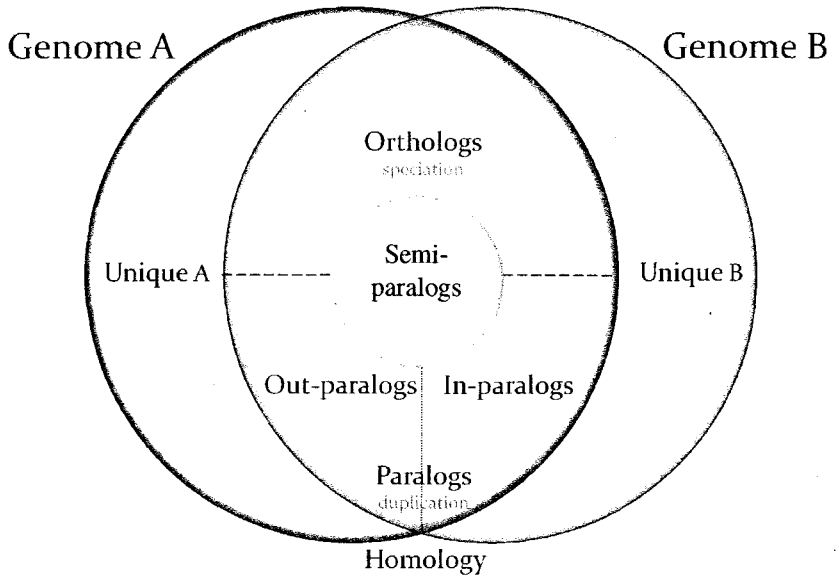


Figure 5: Pictorial representation of our sequence separation categories. It identifies in a two way comparison the unique sequences of each genome, as well as the complication of sequence similarity groups (homology). We have identified in this diagram the orthologs and paralog separation in which there is not a clear separation. We also identify paralogs broken down into in-paralogs and out-paralogs (again with unclear separation). We also identify semi-paralogs which overlap various categories.

2.6.1 Overall Process

Our process for extracting information from the genomes has only a few basic steps. The first step includes taking two nucleotide whole genome sequences (or

identified gene sequences) in FASTA format and blasting one against the other. For a given pair of sequences A and B, similarity is determined with a set of BLAST criterion for the pair of BLASTs. This generates a set of BLAST hits with A as the target, B as the subject and another set of BLAST hits where B is the target, and A as the subject. Both of these results are stored in a database. Second, a query to the database, with another set of minimum match criteria, is generated to gather the list of matching sequence that satisfies the criterion (homology extraction). The third step involves parsing these results into OPUS categories. Finally, we perform post-processing analysis to sort these results further to address various biological questions (Figure 6).

2.6.2 Orthologs

Orthologs are derived from speciation events. However, there is no way of telling if a speciation event occurred from sequence data. To be conservative, we only classify segments as orthologs if they have exactly one copy in the other genome. Post-processing analysis (e.g., other category parsing techniques such as Reciprocal Best BLAST (RBB) (Hirsh and Fraser 2001) and phylogenetic reconstruction) can be performed to identify other possible orthologs.

2.6.3 Paralogs

Paralogs arise from a duplication event. Again, it is hard to infer the true evolutionary history of these sequences. Since we are very conservative with our orthologs (single copy) detection, we initially are very lenient with our paralogs (multi-copy) list. Therefore, the multi-copy category is any sequence that hits multiple places in the other genome.

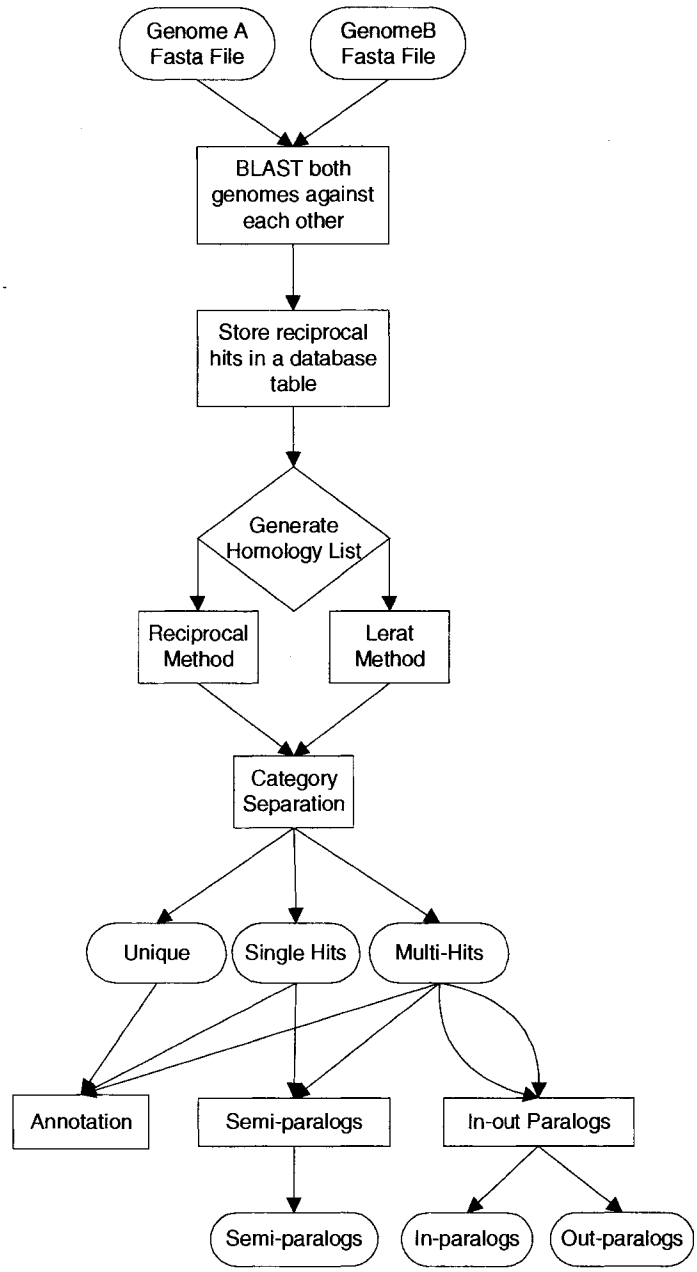


Figure 6: OPUS categorization process

2.6.4 Unique Sequence

We identify unique sequence as sequence that are not found in the other organism (no homology, a BLAST hit meeting the set criteria, was found). These are potentially a source of functional novelty in the genomes. Because, in our analysis "uniqueness" is

only in the context of pairwise comparisons (or in a series of pairwise comparisons), the sequence is usually found to match a different genome in GenBank.

2.7 Expanded Process

2.7.1 The database

The database addition to the Mauve program provides a good platform for subsequent analysis. The database is populated with the bi-directional BLAST results for all matching regions. A query to the database is formulated based on the type of homology the user specifies.

2.7.2 Homology Extraction

There are two types of homology generation we provide with the additions to Magenta. One of the methods generates match criteria based on reciprocal matches. The other method is based on a scheme described by Lerat et al. for identifying gene pairs (Lerat, Daubin et al. 2003). Each method constructs a set of homologous sequences based on the alignments generated by BLAST.

2.7.2.1 Reciprocal BLAST Method

Sometimes, the BLAST algorithm produces artificial sequence similarity such as small hits or long hits that have a lot of mismatches. In order to minimize this phenomenon, it is common practice to use reciprocal BLAST or reciprocally best BLAST (for ortholog detection) hits as a form of rejecting possible false hits. We also want to take advantage of the database storage and therefore, we formulate a reciprocal BLAST match query that locates the matches that hit reciprocally. That is, if genome A is blasted against genome B, the match must hit from A to B and B to A.

Some have suggested that the best sequence hits may not in fact be the closest homologs and reporting all hits yields a better analysis (Koski and Golding 2001). Although, this seems to occur more frequently for distant genome comparisons, we still implemented the program with this concern in mind. Therefore, our method differs from the Reciprocal Best BLAST (RBB) technique (Hirsh and Fraser 2001) because these matches do not have to be the *best* hit in both directions, but rather must hit in both directions and pass a user specified similarity threshold. Later, we discuss a search method that uses a technique very close to reciprocal best BLAST to find long sententious blocks of homology in both genomes.

One complication in this method involves alignments that are of dissimilar sequence lengths or that are partially overlapping. We address this problem by extracting the intersecting hit regions from the database to produce a list of homologs for each of the subsequent comparisons. This allows us to find regions of similarity even when there are length differences, which is sometimes overlooked in other reciprocal BLAST methods.

2.7.2.2 Lerat Method

Another method we use to find homologs is what we refer to as the Lerat method (Lerat, Daubin et al. 2003). This method initially was used to find gene families. Gene families are a collection of genes that have a similar function. However, this method seems to provide a good list of homologs, as well.

The theory behind this method is instead of using the e-value or bit score alone as a similarity threshold, we use a self-hit ratio. A self hit is one in which a sequence is blasted against itself. A self-hit ratio is one in which the bit score from a comparative sequence is compared to the bit score of the self BLAST (equation below where 'a' is a

sequence from genome A and 'b' is a sequence from genome B). A BLAST hit is deemed good enough to be input into a homology list if the ratio is above a given limit.

$$Self\ Hit\ Ratio = \frac{Bit\ score\ (a \rightarrow b)}{Bit\ score\ (a \rightarrow a)}$$

Our implementation of this method is actually slightly different than that of the original method used in the paper. We again want to take advantage of the database. In the original method, a gene is blasted against itself and then blasted against the other gene set to find the bit score ratio. We, however, wanted to maintain the program's ability to work with genome-wide comparisons as well as gene sequences intact. Therefore, we constructed a method that takes advantage of each of these requirements. Our implementation also starts by performing the initial genome comparisons. It extracts the (single direction) hits from the database and stores them in a new database table. For each of these matches, the program extracts the sequence from the query genome and does two secondary blasts to get its self hit and the hit against the other sequence. We have to blast the sequence back against the other genome because bit score is based on the query and subject sequence lengths. Both of these bit scores are stored with the entry for that hit. We are then able to extract from this table the entries that meet the Lerat bit score ratio cut-off specified by the user.

One artifact of this method is that some fairly small sequence matches will still pass the Lerat bit-score ratio. This makes for long run times (since there are a lot of small matches that have to run through two extra blasts). We implemented an alignment length cut off to address this. If the alignment hit of the initial BLAST is less than 100bp, we do not consider it a legitimate alignment and therefore do not run the two subsequent blasts.

2.7.3 Sequence Categorization

In the next stage, we sort sequences from each genome into three groups: segments that do not have a reciprocal hit to them (unique sequence), segments that have one reciprocal hit to them (orthologs), and locations that have two or more reciprocal hits to them (paralogs).

2.7.3.1 Single and Multi-hit Separation

A list of single hits and multiple hits are generated in a single pass for each genome. From the list of homologs of one organism, if the sequence has hits to multiple places in the other genome, it is said to have paralogous (multi-copy) sequence in the other genome. If the sequence only has one hit to the other genome, it is considered to be orthologous (single hit) sequence with the other genome. This step in our process is not reciprocal, that is, the sequences that are identified as single hits in the other genome can hit the initial genome more than once. This process is done for both genomes, generating two categories of sequences for each genome.

2.7.3.2 Unique Sequence Separation

A list of unique sequences is generated with another pass through the homologs. This time we only look for regions outside the homolog set. That is, this pass keeps track of where the last homolog ended in the genomic sequence and the next one starts. If the distance between them is greater than a given size, the sequence is put into the unique group. Sometimes a sequence in one genome hits a sequence in the other genome with the required similarity, but there is not a strong enough reciprocal BLAST hit. We run a check on the list of unique sequences by blasting the sequence to the other genome

sequence in the comparison and separate the ones with a single directional hit into a *one-way* hit bin.

2.7.3.3 Long Syntenic Blocks

In order to partition a step further, we implemented a method similar to the Reciprocal Best BLAST (RBB) Method (Hirsh and Fraser 2001), since it is very widely used and recognized. Again, we want to take advantage of the database storage implemented in Magenta. As such, we use the reciprocal BLAST hits query generated by the method described above to parse out similarity with a variation on the 'best' algorithm. Our algorithm finds long syntenic blocks of sequence that is similar between genomes. This is a nice algorithm to gain a sense for the conserved segments, or ancestral copies of sequence, between the genomes.

Our algorithm starts with the reciprocal BLAST homology list. Then, as it is parsing the data into single and multi-hit regions, it also determines whether the piece has a better match for that region elsewhere. Since bit score and e-value are based on the lengths of the hits, sometimes, if a smaller hit region is longer overall but has a smaller overlap region (in comparison to the alignment length of the current segment), this hit will have a better e-value or bit score than the original piece when, in actuality, the original piece should give the best hit region. The initial RBB algorithm only identifies a hit as a match if it is the "best" in both directions. However, we store all the hits in the database and in order to find the best hit, we would have to apply another blast which would defeat the purpose of storing the hits. Therefore, we set criteria for finding the "best" hits instead. An overlapping hit or smaller hit inside must cover at least 50% of the alignment sequence to be considered as a match. This allows smaller sequences to be

placed in the multi-hit regions and larger regions to be assigned to this category. Also, if the difference between the hit and its next best is less than an epsilon value (user input), the two hits are classified as multiple hits. As a result, any hit that is less than 50% of a larger sequence or hits that have better hits (greater than the epsilon difference) are grouped as multi-hits. Data that does not have other hits that align well enough to more than half of the sequence are classified in the 'best' group. We noticed that these segments were sequences with the largest matches to regions, or long syntenic blocks of homology.

One caveat to this method however is that it finds long blocks of continuous sequences that can overlap other pieces of long blocks by enough nucleotides to be a significant number (larger than a typical gene size). Therefore, these blocks are not able to be classified as potential orthologs as they would have been with the reciprocal best BLAST method.

2.8 Post-analysis

There are various post-processing analyses that can be applied to the data generated above. In particular, we perform annotation steps, semi-paralog identification, and in/out-paralogs classification.

2.8.1 Sequence annotation

One of our goals is to annotate segments of sequences that are not fully assembled and potentially identify genes that might have appeared in the organism as a result of lateral gene transfer. We decided to tackle this by determining the other types of sequences that might hit a particular sequence in one of our categories. Our algorithm takes the nucleotide sequence from the positions identified as one of the categories and

BLASTs them against the non-redundant NCBI nucleotide database. The next steps format the results and then use the NCBI genome identifications to look up any gene's annotations from the GenBank database within the aligned positions of the genomes and output the description to the BLAST hit list. Typically, we find many small hits are in this list. We give the option for the user to decide how to filter these results by length, percent identity, or maximum number of hits.

These results can help reveal the functions associated with sequence categories and what the functional variation is between the genomes. These results also identify sequences from other organisms with high similarity to the categorized regions. These organisms might represent possible donors that provided genes to the genome under study or identify potential gene function. For example, with our dataset, we tested and confirmed the hypothesis that some of the unique sequences were homologs of genes in other bacteria that live in the same environment as *Burkholderia* (Cooper pers. comm.).

2.8.2 Semi-paralogs

Semi-paralogs are regions that are multi-copy within one genome, but mono-copy to a single sequence in the other genome. Figure 3 shows an example of a semi-paralog; genome A has a segment 'a' that hits multiple places, 'b1' and 'b2', in genome B, while 'b1' and 'b2' both have reciprocal hits back to 'a'.

These can be generated by analyzing coding sequences (gene files) of two genomes with the process described above. We use the lists of mono-copy gene pairs and multi-copy gene pairs of each genome to identify the semi-paralogs. We compare the multiple copies of genome A with the single copies of genome B to obtain the semi-paralogs between genome A and B. We do the reverse to acquire the semi-paralogs

between genome B and A. These semi-paralogs are then annotated with the COG information for determining function.

2.8.3 In-out paralogs

In-paralogs are paralogs that happened after the last speciation event while out-paralogs happen before the last speciation event. Figure 7 shows duplication for each of these processes. Figure 7a shows an out-paralog where genomes are duplicated in the first step and then duplicated while Figure 7b shows an out-paralog where the segment was duplicated in the second step. Set analysis can provide information about the extent of shared paralogy by comparing the set of in-paralogs to the set of out-paralogs. We can identify the set of paralogs for each genome by running the OPUS process on each genome against itself in order to obtain a list of paralogs within a genome. We can then match the paralogs in the multiple copy data extracted from running the pairwise OPUS process. This gives us an idea of how many copies are results of duplications within a genome and which are the duplications between closely related genomes.

In-paralogs are duplicates that are in one genome (A) but not in the other (B). Out-paralogs are duplicates that have the same number in both AxA and AxB comparisons (Figure 7). There are three basic rules we follow when identifying a gene as either of the two paralog categories:

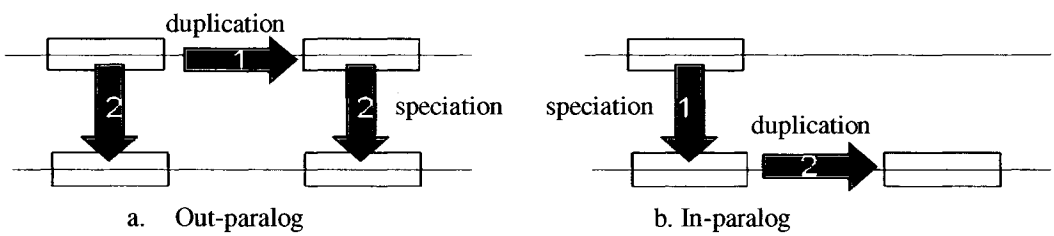


Figure 7: Illustration of in-paralogs and out-paralogs. Out-paralogs are duplications before a speciation event (left). In-paralogs happen after a speciation event (right).

- Rule 1: when a gene is identified as paralogous in both comparisons (AxA and AxB) then the gene is placed in the "out" paralog category
- Rule 2: any gene in the AxA comparison not in AxB comparison is placed in the "in" paralog category
- Rule 3: any gene in the AxB comparison not in the AxA comparison is discarded.

The implementation of this method starts with paralog pair files (output files from OPUS) which are lists of paralogs displayed in pairs (of gene names) of alignments. There are two paralog pair files: one for genome comparison AxA and one for AxB. We then compare the two sorted files that contain two columns of gene pairs (g1 and g2, where 1 and 2 refer to column numbers). These files are sorted for faster searching which allows the program to incrementally stepping through each of the sorted gene pair files. There are three conditions:

- Case 1: When g1 in AxA matches g1 in AxB then g1 goes into the "out" paralog category. This is done with an equality test of the current g1 in AxA and g1 in AxB (satisfies rule 1).
- Case 2: When g1 in AxA does not match anything in AxB, then g1 goes in the "in" box. We determine that g1 does not match anything in AxB by comparing the current g1 in both files. If g1 in AxA is less than g1 in AxB then this is considered a mismatch (satisfies rule 2). In this case, the next pair in AxA will also be compared against g1 in AxB.
- Case 3: When g1 in AxB does not hit anything in the AxA comparison, we discard this match; we assume that in a BxB vs. BxA comparison this match

will be put into an "in" box for genome B (although this may not be the case because our method is not reciprocal). We determine that g1 in AxB does not hit anything in AxA by comparing the current g1 in both files. If g1 in AxA is greater than g1 in AxB, then g1 in AxB is discarded (satisfies rule 3).

We describe a few examples of how this method works. If, for example, there are five duplicates in AxA and only three in AxB, three would be out-paralogs and two would be in-paralogs in A. However, if there is a case that there are two duplicates in AxA and three in AxB this would be reported only as two out-paralogs for A.

These gene numbers can then be annotated with fully annotated gene names to acquire the functions of each. These genes could also be annotated with COG functional groups using *OPUS Notes*.

2.9 Results

We report an overview of the type of output our tool can generate and briefly interpret some of the biological implications. A more detailed biological analysis of these comparisons will be published elsewhere.

2.9.1 Genome-wide Comparisons

We ran the raw, complete genome sequence for *B. cenocepacia* strains AU1054, HI2424, J2315, PC184, and *Ralstonia eutropha* (the out-group) through our methods. Table 1 shows a small portion of the output from a basic comparison between two closely related strains of *B. cenocepacia,* AU1054 and HI2424. It gives a basic overview of the amount of sequence (base pairs) in each of the categories as well as the number of regions or sequence groups in each category. It also provides an overview of the maximum, minimum, and average sequence sizes of regions in each group. Notice that a

large number of base pairs of the sequence are hitting multiple places (two or more hits) in the other genome. Homology images of the *Burkholderia cenocepacia* comparisons can be seen in Figure 8. Each dot in the scatter plot represents a 1000bp match found by our reciprocal BLAST method. *B. cenocepacia* strains AU1054 and HI2424 are most closely related and there are longer blocks of continuous sequence in these two comparisons while the other comparisons are more broken.

Another display of our comparisons can be seen in Figure 9. This figure represents the amount of sequence in base pairs that is associated with one of our OPUS categories. Figure 9a demonstrates that the number of sequences unique to a genome increases greatly with phylogenetic distance. AU1054 and HI2424 are members of the same lineage of *B. cenocepacia*, PC184 and J2315 are also more distant strains of *B. cenocepacia*, whereas *Ralstonia eutropha* JMP134 is a much more distant out-group (Figure 1). Collectively, all of the *B. cenocepacia* strains share roughly the same number and type of unique sequence relative to *R. eutropha*. This pool of genes can serve as candidates to understand the unique functionality of *B. cenocepacia* as compared to *Ralstonia*.

	Total Sizes in bp	Regions	Average Size	Max	Min
B. cenocepacia AU1054					
Unique Regions	3110	2	1555	2846	264
Only 1 hit in paired genome	52755	6	8792	22179	302
2 or more hits in paired genome	7069023	39	181257	1191585	1196
B. cenocepacia HI2424					
Unique Regions	362070	28	12931	83348	200
Only 1 hit in paired genome	44352	26	1705	22179	101
2 or more hits in paired genome	7241305	38	190560	1191594	131

Table 1: Overview of the output produced by OPUS with genome-wide comparisons. This is the comparison of *B. cenocepacia* AU1054 and *B. cenocepacia* HI2424.

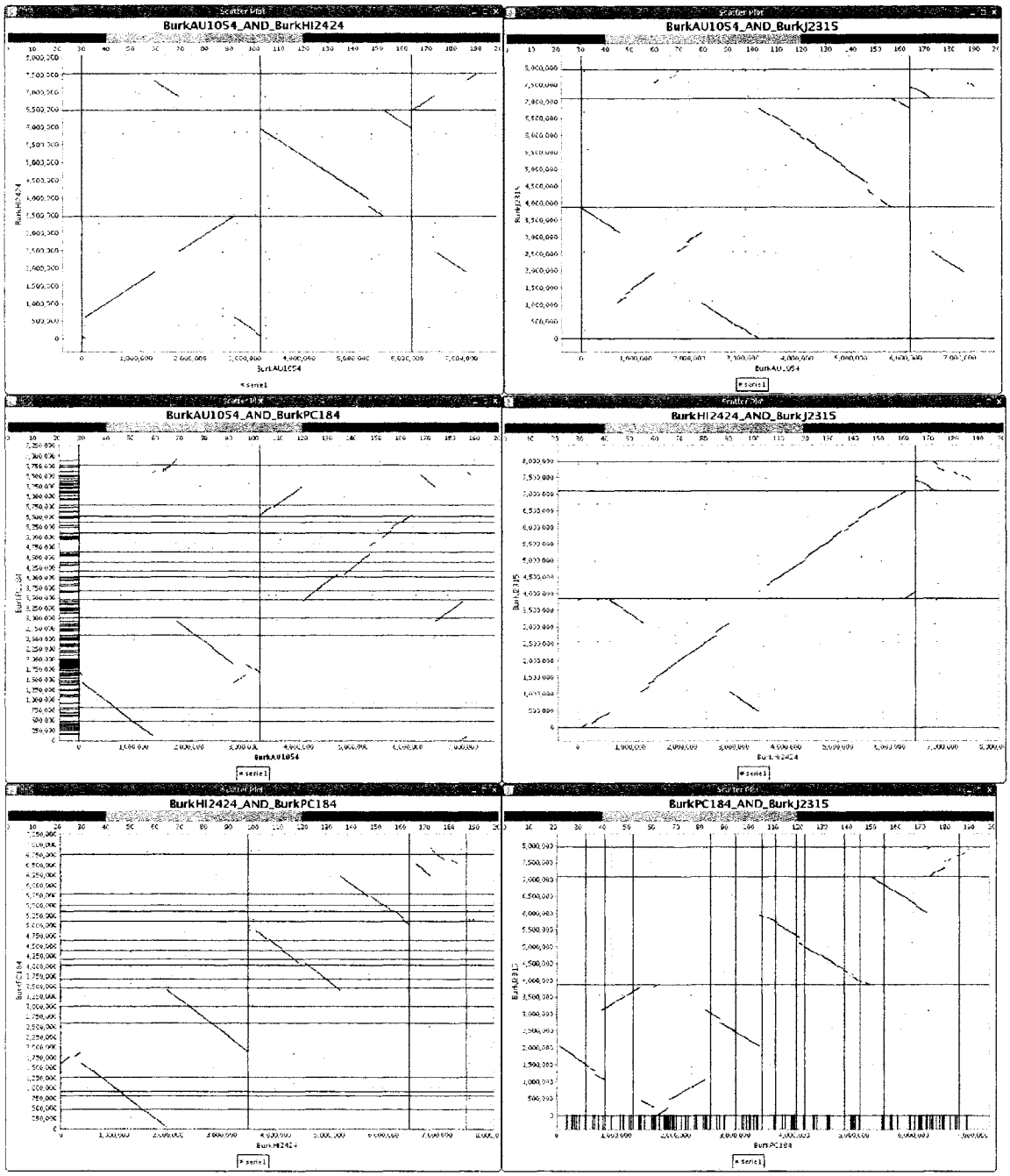


Figure 8: Scatter plot comparisons of homology in four *B. cenocepacia* genomes. The scatter plots are set up so that each axis is a genome sequence and the dots represent the BLAST hits for every 1000bp. Note PC184 is not closed. The horizontal and vertical black lines represent chromosome and contig breaks (PC184) in the sequence. The colors are used to depict the log of the e-value of the BLAST hit. The black dots are to distinguish ends of hits. Also notice large portions of PC184 in the comparisons are in the opposite direction in the comparison which suggests that the genome sequences were simply reversed.

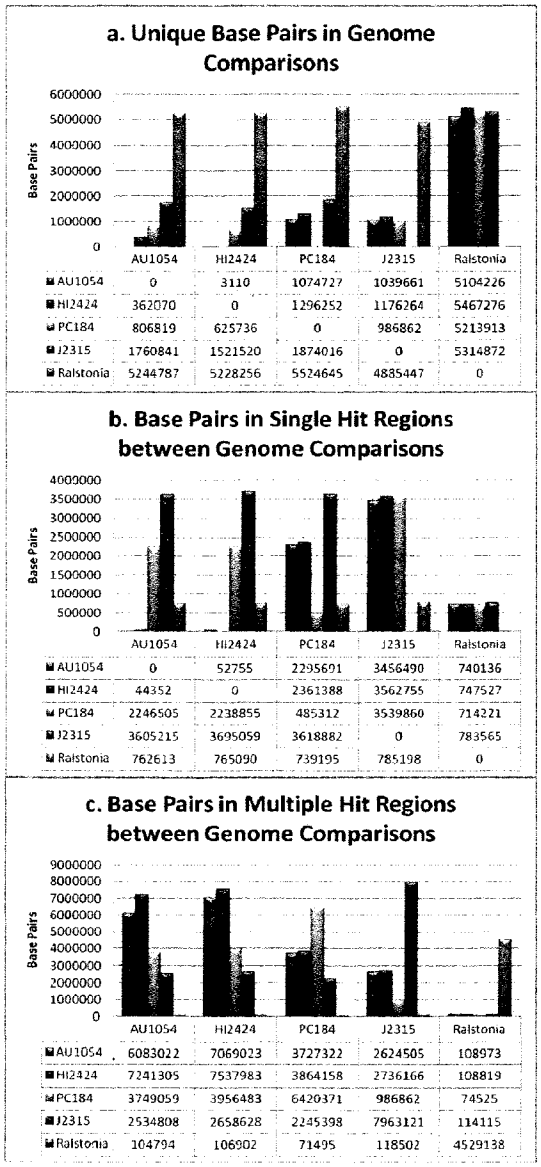


Figure 9: Comparison of the amount of sequence in each sequence group. Each graph shows the pair-wise genome-wide comparison of the four selected *Burkholderia* genomes with the out-group *Ralstonia* genome. (a) Unique sequence represented in base pairs associated with the sequences identified for this group. (b) Single hit sequences (i.e. only one exact match) represented as a sum of the base pairs associated with this group. Specifically, total lengths of the regions that have a single hit to the other genome (i.e. in AU1054 the length of all the regions that hit HI2424 in only one place is 44352 bps.) (c) Multiple hit sequence (i.e. more than one match from the query genome to the target genome, suggesting paralogy) represented as a sum of the base pairs associated with this group. Specifically, total lengths of the regions that hit more than one place in the other genome (i.e. 7241305 bps hit from AU1054 to multiple places in HI2424).

2.9.2 Gene comparisons

In the prior section, we began by comparing raw, unannotated sequences of each genome; here we focus on sequences within predicted genes within the genome (CDs, coding regions) in FASTA format.

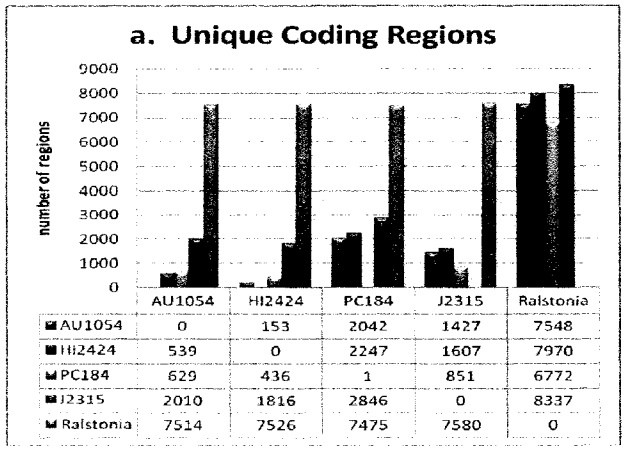
The qualitative output of these queries is very similar to those reported in genome-wide comparisons (compare Tables 1 and 2), but the additional knowledge of annotated sequence can lead to greater insight by providing protein coding region information that could be associated with a known function and groups. Notice that there are more regions in each of the categories when comparing gene files to genome-wide comparisons. This is because the files are broken into genes instead of full sequence. The regions in the gene file comparison generally refer to a single gene or part of a gene that is in a specific category. Also notice that the sequence lengths themselves (max, min, and average columns) are smaller in size. Figure 10 demonstrates the variation of regions within the gene comparisons. Essentially, this is a categorization of the number of genes in each of the OPUS categories; since generally each region hit is gene sized. However, this is not always the case because portions of the genes can be identified as well. It still gives the user an overview of the number of genes in each category.

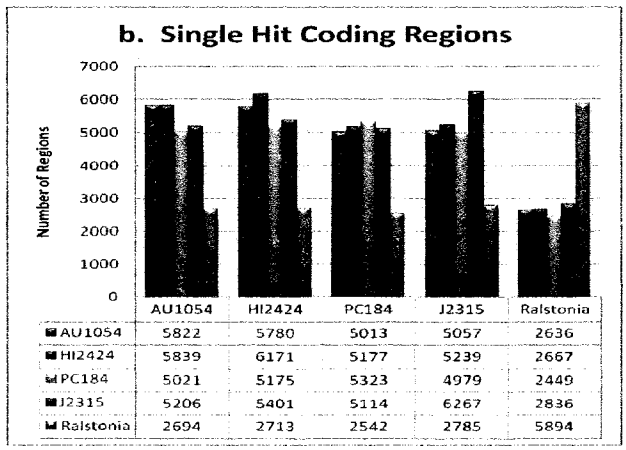
Figure 10a and 10c report the classification of homologs into orthologs (single hits) or paralogs (multiple hits). These gene-level comparisons report a much larger number of orthologs than paralogs, which is the opposite of that reported in Table 1. This discrepancy is likely a product of the varied motifs found in the larger multi-gene segments used in Table 1, which trigger multiple hits and suggest paralogy, as opposed to the single-gene queries of Figure 10, which are more likely to hit only once (orthology).

Another view of the categorized gene region distribution is shown in Figure 11 These graphs represent the same data presented in Figure 10: in this case there is a radial axis for each genome as a target and each polygon represents a genome as the subject. It shows, in another way, the close relationship of the *B. cenocepacia* genomes and the distant relationship of *Ralstonia*.

	Total Sizes in bp	Regions	Average Size	Max	Min
B. cenocepacia AU1054 genes					
Unique Regions	34065	153	222	1493	100
Only 1 hit in paired genome	5396030	5780	933	13530	104
2 or more hits in paired genome	980486	749	1309	11469	113
B. cenocepacia HI2424 genes					
Unique Regions	389177	539	722	4277	100
Only 1 hit in paired genome	5435626	5839	930	13530	101
2 or more hits in paired genome	950879	722	1317	12717	113

Table 2: Overview of the output produced by OPUS on predicted gene files. This is a comparison of *B. cenocepacia* AU1054 and *B. cenocepacia* HI2424 gene files.





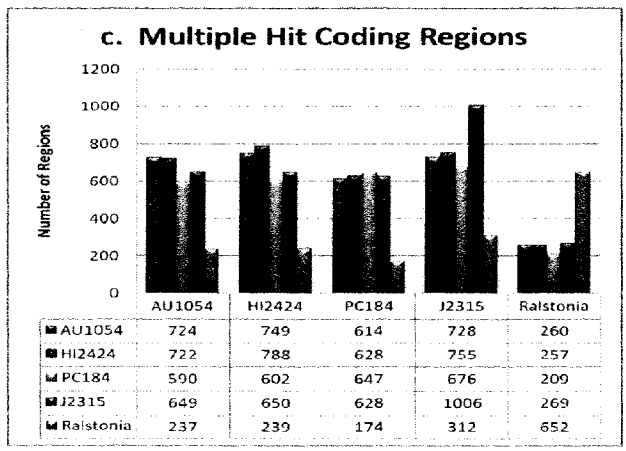
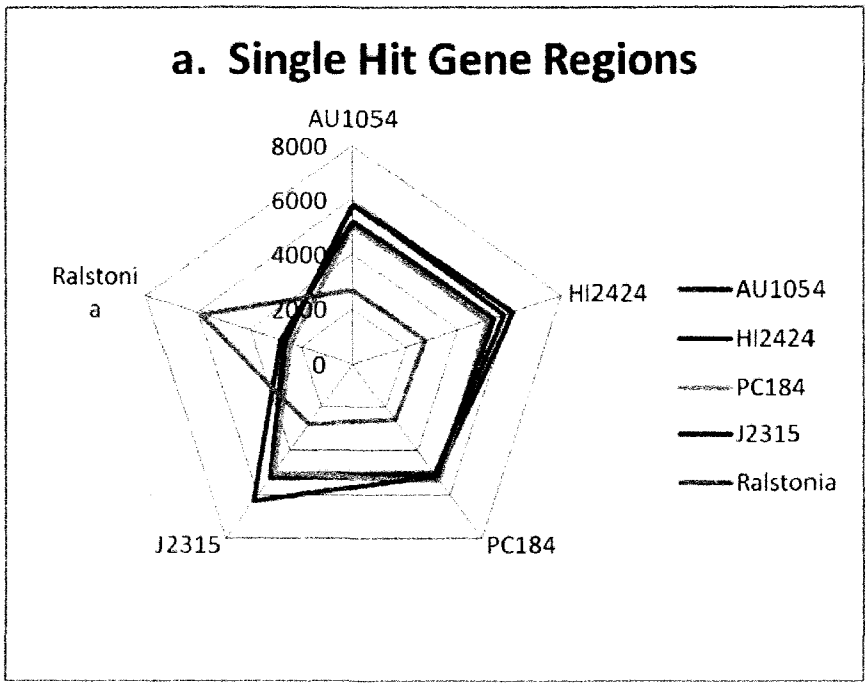


Figure 10: Comparison of the number of regions in each sequence group. Each graph shows the pair-wise gene comparisons of the four selected *Burkholderia* genomes and the out-group *Ralstonia* genome. Since this figure shows gene comparisons, generally a region refers to a gene. However, there can be regions that are only a partial segment in a gene. (a) Unique regions within the comparison. (b) Single hit regions in the comparisons. (c) Multiple hit regions in the comparison.



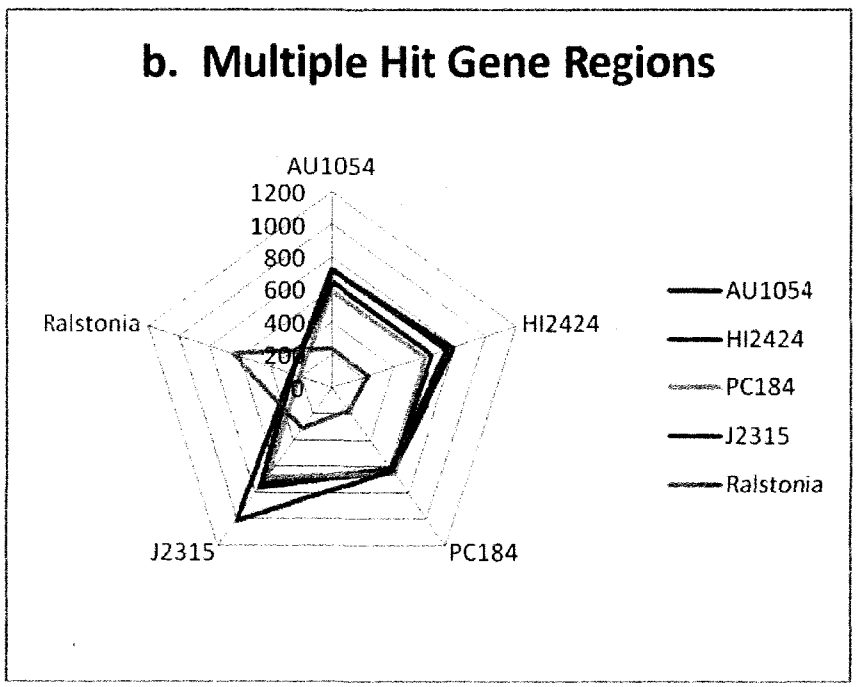


Figure 11: Another view of the single and multiple hit gene regions in the five genome comparisons. Notice that *Ralstonia* has less single and multiple hit regions than the four *B. cenocepacia* genomes.

2.9.3 Homology Methods

Table 3 compares the output of our two homology detection methods: reciprocal BLAST and Lerat. The reciprocal BLAST method makes sure a sequence matches from genome A to genome B and from B to A; the Lerat method uses a self-hit ratio to determine sequence similarity. The outputs are very similar using comparisons between *B. cenocepacia* strains AU1054 and HI2424, although the Lerat method identifies fewer multiple hit sequences within the genome comparisons and more single-hit sequences. We suspect this is because the Lerat method throws out more sequences because they do not pass the self-hit ratio. Each of these methods can be used to generate results based on the type of information the researcher prefers. Perhaps it might be possible to gain better insight into the differences between the two methods by running them on known datasets. Table 3 shows an example of the variation of the outputs that might be of interest.

The researcher might also like to vary the Lerat ratio in order to increase unique sequence or single hit regions, or perhaps decrease multiple hit regions in the data. Figure 12 demonstrates the effect of varying the Lerat ratio cutoff on the two way comparison between *B. cenocepacia* AU1054 and HI2424. In this comparison, as the cutoff value increases, both the unique sequence and single hit sequences increase while the number of multiple hit sequences decrease. We expect this to generally be the case because increasing the Lerat percent means that fewer matches will pass the criteria; the optimal percent cutoff should be determined for a particular comparison by varying the parameters based on the user's goals as described in section 2.10.2.2.

	Lerat at 0.30	Reciprocal
B. cenocepacia AU1054 genes	**Total Sizes in bp**	
Unique Regions	34065	34065
Only 1 hit in paired genome	5469539	5396030
2 or more hits in paired genome	906977	980486
	Regions	
Unique Regions	153	153
Only 1 hit in paired genome	5840	5780
2 or more hits in paired genome	689	749
B. cenocepacia HI2424 genes	**Total Sizes in bp**	
Unique Regions	544	539
Only 1 hit in paired genome	5898	5839
2 or more hits in paired genome	660	722
	Regions	
Unique Regions	393504	389177
Only 1 hit in paired genome	5510527	5435626
2 or more hits in paired genome	872179	950879

Table 3: Comparison of Reciprocal BLAST and Lerat methods for finding homology within the genomes. This is a comparison of predicted genes in the genomes *B. cenocepacia* AU1054 and *B. cenocepacia* HI2424. The Lerat method was run with a self-hit ratio of 0.30. These comparisons are fairly close in comparison. Notice, however, that the Lerat method identifies more single hit regions than the reciprocal BLAST method. Also, it identifies less multiple hits in the neighboring genome.

2.9.4 Annotation

We have written scripts to provide additional functional descriptions of the sequence matches. For example, we re-blast unique sequences against the global GenBank database to determine their likely origin and function. We also can report the G+C content percentage of the nucleotide sequence, which is useful in detecting horizontal gene transfer as it is associated with low G+C regions. An example of this output can be viewed in Tables 4 and 5. These are only some excerpts from a group of the unique sequences identified in the comparisons between HI2424 and AU1054 and between AU1054 and J2315.

2.9.5 Semi-paralogs

We use the term *semi-paralog* to identify genes that are paralogous within one genome but orthologous to a single gene in the other genome. These are especially

interesting because they identify gene gain or loss in a particular gene family (Figure 13). Once we identify sequences meeting this criterion, we use the methods described in *OPUS Notes* to associate gene annotation, COG type (which summarizes function), and K_a and K_s values (which denote strength of selection). Table 6 presents an example of this output that demonstrates elaboration of an interesting gene family by two newly diverged copies. Table 7 summarizes our semi-paralog analyses among our five focal genomes and illustrates how these genomes have become enriched for genes of different functional categories.

2.9.6 In-paralogs and Out-paralogs

Semi-paralogs are only a subset of the genes gained or lost due to enhanced function or selective pressures. More specifically, semi-paralogs are a subset of in-paralogs since semi-paralogs have a one-to-many relationship and in-paralogs have a many-to-many relationship. Our tool separates the in-paralogs and the semi-paralog cases. So, in-paralogs and out-paralogs are the more general identification of gene family size variation in evolution. Figure 14 describes the overall picture of the number of genes classified as in-paralogs and out-paralogs. Notice that J2315 is comprised of nearly all in-paralogs with respect to the other genomes and very few out-paralogs.

2.9.7 Long Syntenic Blocks

Finally, we present the results from our reciprocal best BLAST method which identifies long syntenic blocks of similar sequence between genomes. This is interesting because they are long blocks that identify similarity between organisms. The identified blocks can be viewed in our scatter plot visualization of blocks of continuity (Figure 15).

Sequences producing significant alignments	Match Len	% G+C	Annotations
a. BurkHI2424-BurkAU1054_3793937_3809296_15360_NC_008543_63.57421875			
>gb|CP000459.1| Burkholderia cenocepacia HI2424 chromosome 2	15360	63.57	beta-lactamase "aminotransferase, class V" "conserved hypothetical protein" "hypothetical protein" "MoeA domain protein, domain I and II" "GCN5-related N-acetyltransferase" "transcriptional regulator, TetR family" "Glutathione S-transferase, N-terminal domain" "sodium/hydrogen exchanger" "acyl-CoA dehydrogenase domain protein" "YbaK/prolyl-tRNA synthetase associated region" "amino acid adenylation domain" "Thioesterase" "hypothetical protein" "chlorinating enzyme" "conserved hypothetical protein" "Extracellular ligand-binding receptor"
b. BurkHI2424-BurkAU1054_3845088_3848197_3110_NC_008543_67.90996784565917			
>gb|CP000459.1| Burkholderia cenocepacia HI2424 chromosome 2	3110	67.91	chaperonin GroEL "chaperonin Cpn10" "secretory lipase" "class II aldolase/adducin family protein" "binding-protein-dependent transport systems inner membrane component"
>gb|CP000152.1| Burkholderia sp. 383 chromosome 2	1781	67.88	Chaperonin Cpn60/GroEL
>gb|CP000152.1| Burkholderia sp. 383 chromosome 2	999	68.17	Sigma-54 specific transcriptional regulator, Fis family
>gb|CP000441.1| Burkholderia cepacia AMMD chromosome 2	438	59.36	chaperonin GroEL "chaperonin Cpn10"
c. BurkHI2424-BurkAU1054_3848468_3931815_83348_NC_008543_69.16662667370542			
>gb|CP000459.1| Burkholderia cenocepacia HI2424 chromosome 2	83348	69.17	binding-protein-dependent transport systems inner membrane component "binding-protein-dependent transport systems inner membrane component" "ABC nitrate/sulfonate/bicarbonate family transporter, periplasmic ligand binding protein" "ABC nitrate/sulfonate/bicarbonate family transporter, periplasmic ligand binding protein" "ABC nitrate/sulfonate/bicarbonate family transporter, periplasmic ligand binding protein" "Alkanesulfonate monooxygenase" "ABC nitrate/sulfonate/bicarbonate family transporter, periplasmic ligand binding protein" "TonB family protein" "Biopolymer transport protein ExbD/TolR" "Biopolymer transport protein ExbD/TolR" "major facilitator superfamily MFS_1" "major facilitator superfamily MFS_1" "TonB-dependent receptor" "Alcohol dehydrogenase, zinc-binding domain protein" "transcriptional regulator, LysR family" "major facilitator superfamily MFS_1" "short-chain dehydrogenase/reductase SDR" "sulfatase" "NLPA lipoprotein" "conserved hypothetical protein" "conserved hypothetical protein" "conserved hypothetical protein" "conserved hypothetical protein" "transcriptional regulator, LysR family" "protein of unknown function DUF1445" "major facilitator superfamily MFS_1" "5-oxoprolinase (ATP-hydrolyzing)" "hypothetical protein" "diguanylate cyclase/phosphodiesterase with /PACsensor(s)" "Enoyl-CoA hydratase/isomerase" "hypothetical protein" "conserved hypothetical protein"...

Table 4: Excerpt from the output of the annotation script comparison HI2424 and AU1054 genomes. This example shows how you can find potential functionality of regions by using the annotation pipeline. (a) Shows a segment identified as a beta-lactamase, which cleaves beta-lactam antibiotics. (b) An example of the GroEL gene, which encodes a molecular chaperone protein. (c) Shows a large unique segment in BurkHI2424 that must have a lot of different genes within it since it produces many annotations (these genes had to be cut off to fit on a page).

Sequences producing significant alignments	Match Length	% G+C	Annotation
BurkAU1054-BurkJ2315_971803_973008_1206_NC_008060_71.31011608623548			
>gb|CP000458.1| Burkholderia cenocepacia HI2424 chromosome 1	1206	71.31012	diguanylate cyclase with GAF sensor
>gb|CP000378.1| Burkholderia cenocepacia AU 1054 chromosome 1	1206	71.31012	diguanylate cyclase with GAF sensor
>gb|CP000151.1| Burkholderia sp. 383 chromosome 1	1062	70.15066	hypothetical protein
>gb|CP000440.1| Burkholderia cepacia AMMD chromosome 1	237	70.04219	diguanylate cyclase (GGDEF domain)

Table 5: Excerpt from a unique segment in AU1054 that is not in J2315. This segment is annotated as a diguanylate cyclase which has functions in the regulatory system and plays a role in bio-film formation.

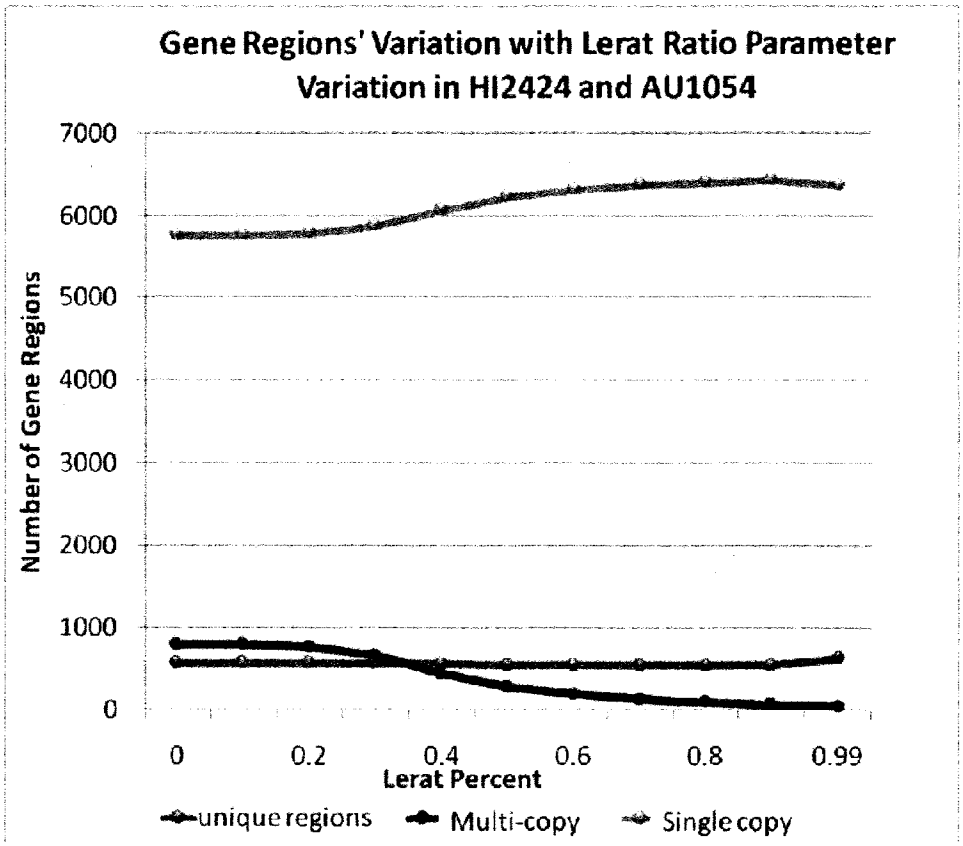
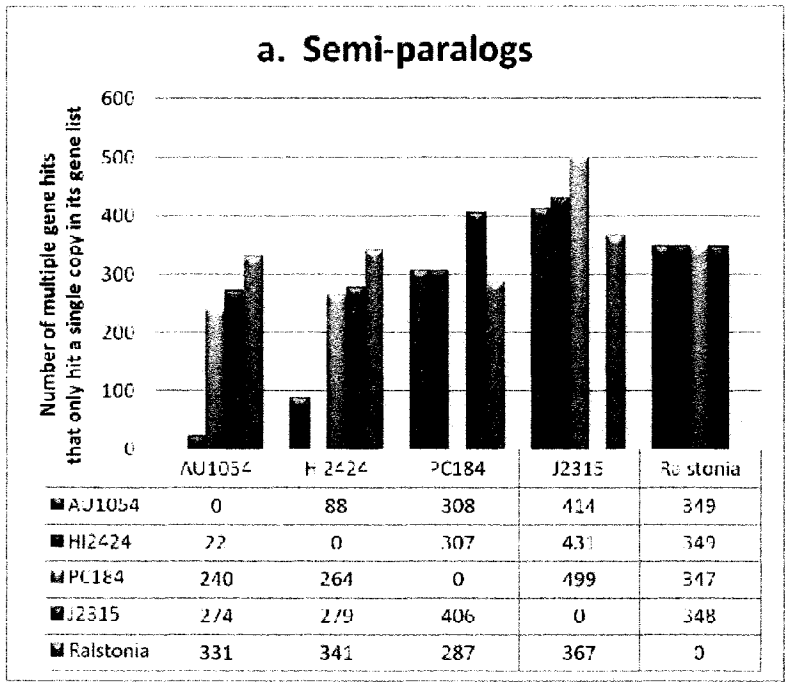
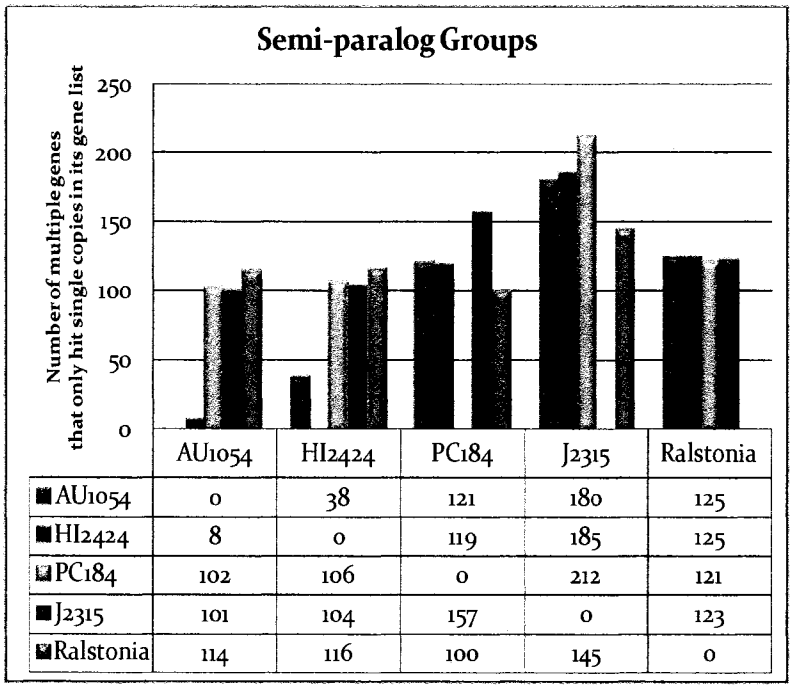


Figure 12: Plot of the variation in the number of genes identified as unique, hitting only a single sequence, or hitting multiple sequences as a function of the Lerat Ratio Parameter. Comparison of *B. cenocepacia* HI2424 to *B. cenocepacia* AU1054.



	AU1054	HI2424	PC184	J2315	Ralstonia
AU1054	0	88	308	414	349
HI2424	22	0	307	431	349
PC184	240	264	0	499	347
J2315	274	279	406	0	348
Ralstonia	331	341	287	367	0



	AU1054	HI2424	PC184	J2315	Ralstonia
AU1054	0	38	121	180	125
HI2424	8	0	119	185	125
PC184	102	106	0	212	121
J2315	101	104	157	0	123
Ralstonia	114	116	100	145	0

Figure 13: A representation of the amount of semi-paralogs in the five genome comparisons. (a) Shows the number of gene hits whose hit only hits back to it. In other words, the figure shows the one-to-many relationships by showing the number of many relationships. (b) Shows the number of multiple genes that are hit by only a single gene in the other genome. In other words, this graph shows the one-to-many relationships as well but with the number of groups described.

AU1054 gene	AU1054 COG	HI2424 gene	HI2424 COG	K_s	K_a
0866 TonB-dependent siderophore receptor	P	6742 TonB-dependent siderophore receptor	P	3.901	0.4445
0866 TonB-dependent siderophore receptor	P	3956 TonB-dependent siderophore receptor	P	0.0189	0.0047
0866 TonB-dependent siderophore receptor	P	3957 hypothetical protein	S	0.1443	0.0616

Table 6: Excerpt from the annotated semi-paralog analysis using *OPUS Notes*. Here, gene 0866 from AU1054 is homologous to genes 6742, 3956, and 3957 from HI2424. Three genes are in COG function P (inorganic ion transport and metabolism), but 3957 should also be reclassified from S (unknown) to P based on its high similarity. We may also predict that strain HI2424 could be functionally enhanced in iron uptake and metabolism based on two additional copies of siderophore receptors, since siderophores are iron chelating compounds.

	A-H	A-P	A-R	H-A	H-P	H-R	P-A	P-H	P-R	R-A	R-H	R-P
Information	113	66	47	9	80	49	36	123	53	79	100	69
Cellular processes	153	229	328	12	195	297	110	290	361	421	437	340
Metabolism	270	133	289	22	133	293	139	351	246	169	213	141
Poorly characterized	179	401	10	13	60	13	39	176	12	19	62	20
Different	176	55	22	11	87	56	38	198	20	28	75	29

Table 7: Frequency of semi-paralogs in the query genome (listed second) for each major COG functional category (A: AU1054; H: HI2424; P: PC184; J: J2315; R: *Ralstonia*) using *OPUS Notes*. A gene often is assigned multiple COGs. This analysis considers all such assignments and therefore the sum of the values in a column is greater than the number of semi-paralogs in that genome. The first four rows in the table report on gene pairs with the same major COG functional categories. The fifth row indicates the number of gene pairs where the two genes have different COG types. Note that HI2424 and *Ralstonia* are enriched for genes related to metabolism, even when compared with one another. Closer inspection of the genes within these categories suggests that, for example, PC184, a clinical isolate, may have gained additional antibiotic resistance capacity by gene duplication.

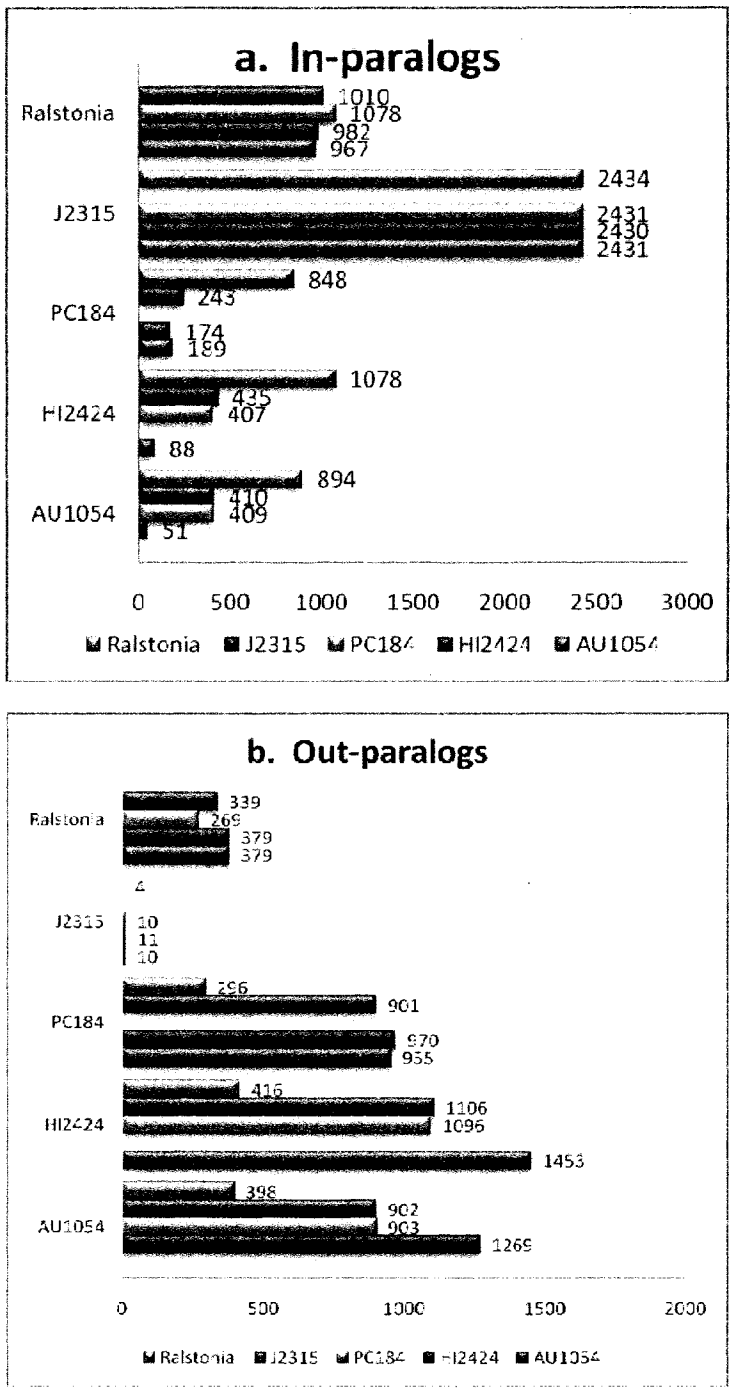


Figure 14: Separation of in-paralog genes and out-paralogs genes in each of the five pair-wise comparisons. In-paralogs are duplicate genes that arise after the lineages split. Out-paralogs are the multiple hits that duplicated prior to the lineage split. Notice that J2315 is made up of almost all in-paralogs with respect to the other genomes.

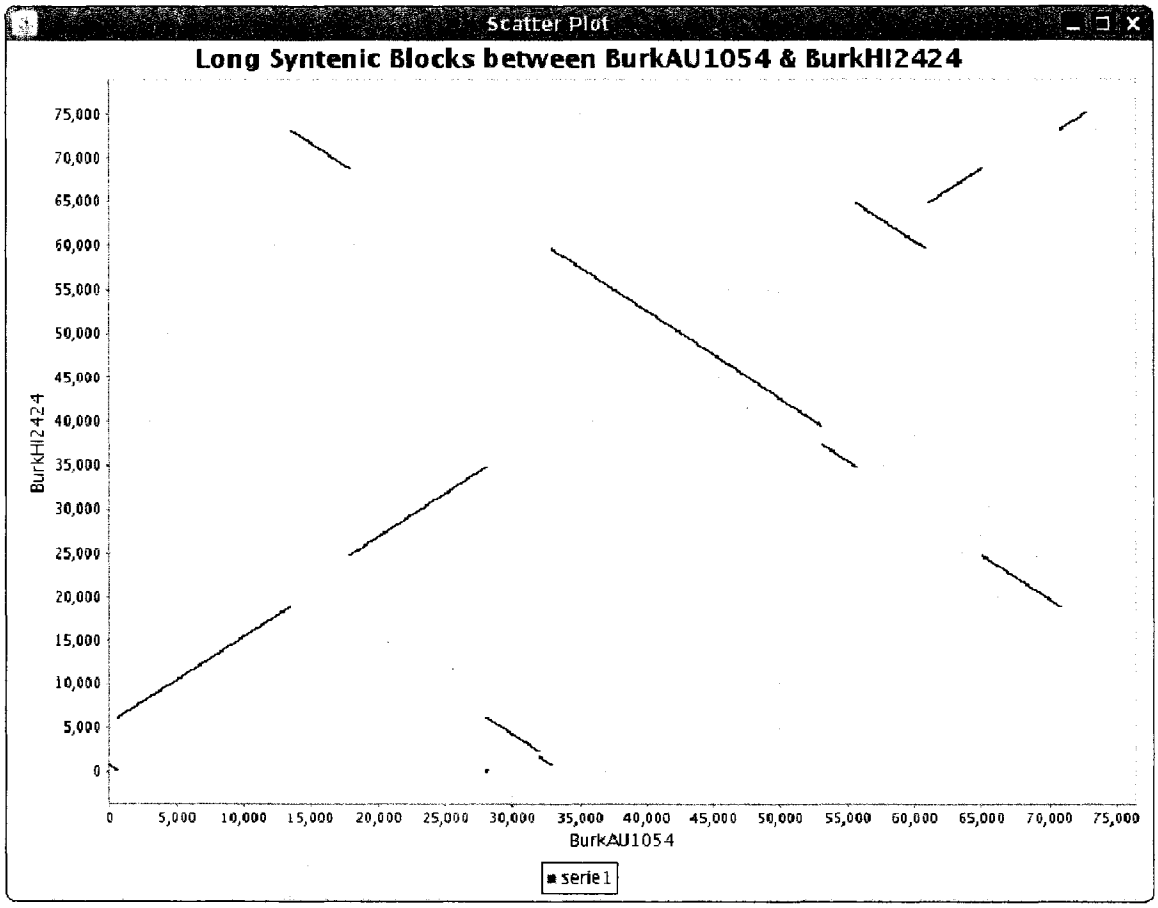


Figure 15: Figure demonstrating the long syntenic blocks found in our version of the reciprocal best BLAST method. Notice that it reduces the clutter compared to the upper left picture in Figure 8 and only finds the long matches where there is not a better hit elsewhere for that region.

2.10 Discussion

Relating genomic phenomena to logical sequence comparisons has proven to be a challenging endeavor. We propose potential organizational groups for sequence similarities and differences. The data produced in the results provides an overview of the categories and types of results that can be produced by *Magenta's OPUS*.

2.10.1 BLAST Anomalies and Optional Parameters

BLAST provides a good search tool for finding sequence similarity because it has the ability to identify our various types of sequence categories, is tolerant of sequence rearrangements, and is a well-recognized program in genomics research. However, BLAST as a search tool the way we use it, has inconsistencies that should be recognized when using this application.

2.10.1.1 Reciprocal BLASTs

First and foremost for our use, BLAST is not reciprocal by nature. There can be length differences in alignments based on which genome is the reference. We also suspect it depends on where the match happens to be when it starts the extension process. For example, if a match is strong for a long stretch of bases and then hits regions with poor matches or low sequence complexity, the algorithm will process past those occurrences to create a longer match. However, if the match is shorter before it hits these spots, it fails to extend the match and instead produces one or more small matches. This could produce incompatible, non-reciprocal matches. We addressed this issue by using an intersection matching scheme for our reciprocal BLAST method. However, the user should be aware that this inconsistency is still present when using the Lerat method and therefore should probably run Lerat in both directions as well (AxB, BxA).

2.10.1.2 E-value vs. Bit Score

There are two different alignment measures that a match can have: e-value and bit score. It is not clear which one is a more accurate measure of similarity. Bit score is the normalized alignment score of the match (NCBI 2007). Bit score is calculated using an algorithm based on the alignment lengths and measures of similarity to determine its value:

$$S' = \frac{\lambda S - \ln K}{\ln 2}$$

where S is the alignment score and K and λ are statistical values(NCBI 2007).

E-value, expect value(Bedell 2003), is the statistical value that the alignment would happen by chance and is calculated using the bit score taking into consideration the length of the reference (*m*) and query (*n*) sequences (NCBI 2007).

$$E = mn\,2^{-S'}$$

However, the use of an e-value cutoff is not always what is desired. Bit scores produce more detailed measures of variation. However, bit score values have no determined range; so it is hard to determine an optimal bit score value or bit score minimum. Also, the BLAST program does not provide documentation on a parameter to set a bit score cutoff. It does, however, provide an option to input an e-value cutoff for the alignments. We also allow the user to use this option in our program. We use each of these values for different methods. The Lerat method was based on bit score ratios, so we continued that convention. For the reciprocal 'best' BLAST method, we use e-values but this decision could be easily changed to use bit scores instead.

2.10.1.3 Complexity Filtering

One complication when using BLAST for our sequence similarity search involves hits that are artificially broken due to low complexity regions. We reduce this issue by using the soft mask filter parameter in BLAST, which helps to extend matches across short regions of low similarity and reduces the number of broken hits. NCBI BLAST has three variations on this parameter: filtering on (-FT), filtering off (-FF), and soft mask filtering (-F mD for *blastn*, -F mS for all others) (Bedell 2003). We prefer using the soft mask filtering for our data because it seemed to reduce the amount of broken segments, but this parameter can be set during run time.

2.10.1.4 BLAST program types

There are different types of blasts that can be run using different datasets. Our program can run *blastn* and *tblastx* runs. However, since *tblastx* does six frame translations, the runtime and database tables are quite large. The data we present here is all done using the *blastn* program but the user can decide to run different ones with the same blast switch (-p *blastn*, -p *tblastx*) (Bedell 2003).

2.10.1.5 Strands

The default for BLAST is to find similarity on both strands of the sequence. However, if coding regions are involved and the user is only interested in the predicted coding region strand (to better simulate protein BLASTs), then a single strand option is available (Bedell 2003). We used this flag to compare our methods to those using protein sequences. This option can be chosen by using the single strand switch (-S1) for the initial BLAST. Other strand specific commands (-S) can be input as initial BLAST options as well.

2.10.2 Methods

Our methods require the use of certain threshold cutoffs. Unique sequences for our runs of the program were set to be 300bp, which is a typical small gene size for bacterial sequences. Also, an alignment length restriction was set to be 100bp in order to reduce the number of small and potentially artificial alignments. This cutoff also applies for overlapping alignments.

2.10.2.1 Reciprocal BLAST Method

In order to verify our method, we used the sequence files produced by our method and reblasted then back against our genomes. We expected to see that unique sequences would hit the genome that it came from but would not hit the other genome. However, our reciprocal BLAST algorithm exposed the artifact of one-way hits; which are probably due to the inconsistency in BLAST in which a segment is considered unique because there is not a reciprocal hit by our method but there is a hit from one of the genomes to the other. This could happen based on which genome is the reference and which is the query. One theory is that if smaller sections are the reference sequence while larger sequences make up the query genome then the query does not align well to the reference genome. These are because the larger sequences do not have a significant enough match to the smaller segments, whereas the opposite direction (small: large) produces a significant match. We can demonstrate this result by comparing a complete genome sequence with the subset of predicted gene sequences. We found a large number (over 500) of one-way hits when running this comparison. When the gene file is the reference, it is hard for a large segment to hit the small gene-sized pieces. This causes a large number of one-way hits between these comparisons.

2.10.2.2 Lerat Method

The Lerat method generates more single copies than the reciprocal BLAST method and identifies fewer multiple copy regions than the reciprocal BLAST method. As a result, this method is more restrictive with its hit requirements for our test case. Therefore, it removes slightly more potentially ambiguous hits than the reciprocal BLAST method.

We had to set a base pair size requirement in order to decrease run-time, because even small hits would have to run through two subsequent BLASTs using this method. Therefore, we set a size restriction of 100bp for these runs of the data. With this limit, we reduced our run time from about a week to only a few hours.

We verified our Lerat comparison by comparing our output with those generated by another implementation of the method written by Philip Hatcher. The other implementation uses the original Lerat method which uses a full gene's self hit bit score for the ratio. Also, it is implemented with protein sequence using the *blastp* command. In order to compare results, our Lerat method was run with the single strand option in BLAST allowing us to better simulate the protein coding comparisons.

The two Lerat methods came up with similar results. Our Lerat method generates more unique sequences than that produced by the Hatcher method because our method identifies partial genes as being unique, which was expected. OPUS, executing the Lerat method, identified 153 unique regions in AU1054 that differed from HI2424. The Hatcher implementation found 57 unique regions. When compared, there were only a handful of unique regions that the Hatcher method found unique that we did not. One of the differences was that OPUS's tool had smaller hits inside the gene that rejected it from

being unique. We also noticed that frame shifts were another cause for incongruence. Here, we refer to frame shifts as nucleotide similarity existing but starting on a different base than the protein starting positions. However, since there were very few differences in our methods, we were satisfied that our methods were computing similar results.

The Lerat ratio percentage was varied for the AU1054 and HI2424 comparison as an example. The Lerat parameter can be adjusted to vary the level of similarity within the comparisons. For example, if the researcher is interested in looking for duplications in the genome, based on our graph (Figure 12) a low percentage value would be desired because it would produce more multiple hit regions in the genome that could be shifted though at a later point. However, if he/she is looking for orthology, single copy genes would be more desired, and a higher percentage should be set, to be more restrictive with sequence similarity.

2.10.2.3 Reciprocal Best BLAST Method

Our version of the reciprocal best BLAST method keeps overlapping sequences, which would not normally occur with the traditional RBB method. For this reason we output whether the sequences have an overlapping end. We also output the number of shorter matches that had hits to those sequences and were not considered better. This will hopefully provide the user with a general idea of the impact of this problem in their data.

2.10.3 Biological Inferences

We present a few interesting biological inferences. More analysis of the data produced by OPUS and the subsequent tools will be presented elsewhere.

Somewhat surprisingly, the vast majority of the sequence found in both *B. cenocepacia* AU1054 and HI2424 genomes hits multiple sequences in the other genome,

suggesting perhaps an ancient complete-genome duplication event. Because they were derived from the same ancestor relatively recently, relatively little sequence can be considered unique, but HI2424 has an extra 290Kb of sequence. It is also notable that more sequence in AU1054 is strictly single copy to HI2424 than vice versa, which suggests that AU1054 has lost portions of multi-copy gene groups found in HI2424. It is interesting that one of these losses involves one fewer copy of the GroEL/GroES chaperonin operon in AU1054 than in the other Bcc genomes (Table 4). Collectively, these data suggest that adaptation to a pathogenic lifestyle may have been accompanied by gene loss for the clinical isolate AU1054.

Now, we turn to the potential biological implications of semi-paralogs, in-paralogs and out-paralogs. J2315 is made up of mostly in-paralogs (Figure 14) which suggests that its duplications typically happened after its split from the other lineages. This would suggest duplications of sequences in J2315. This might imply a need for replication of gene copies in J2315 for enhanced function. This trend is also replicated through the semi-paralogs but not quite as prominently (Figure 13). However, this variation will have to be verified in order to make any biological conclusions since J2315 gene predictions were acquired from a different source that used a different method of gene identification than the other genomes, which could be one of the reasons for great distinction in the data.

2.11 Conclusions

We formulated a method and developed tools for separating and quantifying key similarities and differences among genomes. Our method uses BLAST and stores its results in a database. We can then parse the results into categories similar to the

biological categories: orthologs (mono-copy), paralogs (multi-copy), and unique (no homology) sequences and perform numerous post-analyses on these sequences. This is a valuable tool for bioinformatics and genomics research with considerable potential for inferring sources of evolutionary and functional novelty among genomes.

We have demonstrated that our OPUS toolkit can be used to explore the differences between closely related genomes. We have used the quick access of the database to find various homology relationships. Notably, we have shown how our method can be used to identify potential function for regions of a genome that are unannotated. We have also used other scripts to find potential function variation displayed in DNA sequences. We have also categorized sequences by their relationship in a pair-wise comparison with another genome including mono-hits (orthologs), multi-hits (paralogs), unique sequence, semi-paralogs, in-paralogs, out-paralogs, and 'best' hits (long syntenic blocks).

2.12 Future Work

We have focused on examples of studying orthologs, paralogs, and unique sequences in genome comparisons, but numerous alternative questions are possible. We plan to study more complicated patterns of differential paralogy, such as gene families of different copy number. Second, we remain interested in understanding the phenomenon of one-way, nonreciprocal BLAST hits, which may result from conserved modules present within genes or may be a simple side-effect of the BLAST match criteria. Third, our program is also capable of quantifying similarities and differences in non-coding sequence, which may enable the study of variation in regulatory sequences among closely related genomes. Fourth, we are exploring the potential to conduct comparisons between

multiple concatenated genomes, which can provide insight into more ancient evolutionary patterns and serve to separate in-paralogs from out-paralogs. We would also like to modify our reciprocal 'best' BLAST method to be a reciprocal 'better' method. This method would remove our overlapping block by keeping the longest sequences possible together and only breaking the blocks when another sequence has a better hit someplace else. Most importantly, Magenta's OPUS is a flexible platform that can answer a wide range of evolutionary genomic questions using various types of input sequence.

CHAPTER 3

EVALUATING TWO VISUALIZATION TECHNIQUES FOR GENOME COMPARISON*

Morel Henley, Mikkel Hagen, and R. Daniel Bergeron
Department of Computer Science, University of New Hampshire
{mhenley@unh.edu, mhagen@unh.edu, rdb@unh.edu}
(Henley, Hagen et al. 2007)

3.1 Abstract

Genomic study is fairly novel. Typical research processes are not established yet. Many new discoveries are happening in this area all the time. Are current methods of visualizations effective? What works well? What could be improved? These are some of the questions we are interested in evaluating for two graphical tools used to compare nucleotide sequences. Scatter plots and parallel coordinate-like visuals have been used in genomics for identifying similarities in genetic code. Our preliminary evaluation focuses on determining the aspects of the two visualizations that are successful and those that need enhancements.

3.2 Introduction

Genome analysis produces large amounts of data. It is hard to display this information in a way that is effective for understanding genomes. These visualization efforts are important for both fully sequenced genomes and partial sequences. Comparative genomics focuses on comparing two or more genomes. We are particularly interested in identifying and showing regions of similarity between two genomes. Two standard information visualization techniques have been used to display this information: scatter plots and parallel coordinate diagrams.

Dot plots, a form of scatter plot, have been used for comparing sequence variation since about 1981, which is early in genomic studies. It was introduced for genomic research when Maizel and Lenk used dot plots to visualize nucleotides (Maizel and Lenk 1981). Dot plots have been used in programs such as "dotup", "dotpath", and "dotmatcher" from the EMBOSS package (Rice 2000) and Dotter, another open source dot plot program (Sonnhammer and Durbin 1995). More recently, Rasko has used color in scatter plots to represent significance of BLAST hits at a protein level (Rasko 2005).

Several genome visualization tools have adopted techniques similar to parallel coordinates (Inselberg 1990). Parallel coordinate techniques have been used to compare locations of various functional genes in multiple genomes (Michaels, Carr et al. 1998) and to depict related genome sequences as in the Mauve program (Darling, Mau et al. 2004).

Our goal in this research was to evaluate the effectiveness of the two types of visualizations are for comparing sequences. We describe a heuristic study (Nielsen 1994) to make this evaluation. Section 2 provides background information on the two

visualizations, the genomics field, and the common visualizations for genome comparison. Section 3 discusses in more detail the specific images used in this study. Our evaluation approaches are discussed in section 4. Section 5 summarizes our conclusions gathered from the survey. Section 6 identifies future research.

3.3 Background

In this section, we briefly introduced the scatter plot and parallel coordinate techniques, genomics, and two current visualizations used for comparing genomes.

3.3.1 Scatter Plot and Parallel Coordinates

Scatter plots have been used primarily in statistics and database visualization for comparing two data variates in a collection of records. Two orthogonal axes are used to represent the values of the two data variates; a dot or other glyph is drawn for each data record at the position represented by its values. The pattern of dots can provide insight into the relationship of the data values in a particular data set. One weakness of a scatter plot is that multiple records will get mapped to the same position if their values are the same. Multiple variates are typically shown as an array of bi-variate scatter plots called a scatter plot matrix (NIST/SEMATECH 2007).

Parallel coordinates have also been used for comparing multi-variate data for statistical and database analysis. Each variate is represented as a parallel axis and a data record is displayed by a line connecting data values on each neighboring axis. Due to the sheer volume of genes and nucleotide sequences, the parallel coordinate technique suffers from excessive visual clutter that hides information. The loss in the visibility is a result of the number of crossing connections. This is a classic limitation of all parallel

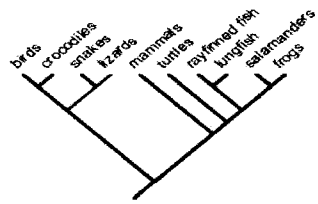


Figure 16: Phylogenetic Tree. (Carrizo 2004)

coordinate visualizations. Pillat et al. record a limitation of about 1000 links before their parallel coordinate visualization becomes indecipherable (Pillat 2005).

Dwyer et al. suggest that both scatter plots and parallel coordinates become cumbersome when dealing with large datasets, even using colors (Dwyer 2006). Since we often deal with hundreds or even thousands of genes at a time, this weakness is a significant factor in the effectiveness of any visualization technique in this context.

3.3.2 Genomics

Genomics is the study of genomes, specifically their genes and gene function. Most genes are distinct segments of DNA that are transcribed to become proteins. At the heart of genomics is the sequence of nucleotides that make up genes and other non-coding sections. Nucleotide sequences are comprised of four bases: A (Adenine), T (Thymine), C (Cytosine), and G (Guanine). Recently, a number of sequencing projects have worked to assemble the entire nucleotide code of many species of plants, animals, bacteria, and other organisms. DNA sequences produce enormous amounts of data that could use effective visualization techniques in order to view and better understand the underlying information in a genome.

Understanding DNA is valuable to many areas of biological sciences. Obviously, it is very important to medicine, as a means of discovering diseases and potential cures. Additionally, the sequencing of DNA can be used to help explain the earth's evolutionary

history. Commonalities can infer the ancestral relationship between two species as well as determining species divergence. A phylogenetic tree is a pictorial view of how species are related and predicts common ancestors (see Figure 16). Phylogenetic tree creation relies on understanding how closely related genomes are.

Recent efforts in *comparative genomics* have focused on comparing closely related species or strains of a genus for similarities and differences. Comparing genomes is not only used in deciding the evolution of organisms; it is also a means of resolving what parts of genomes affect functional behavior. Our goal is to show regions of two genomes that have high similarity measure. Related sequences can be grouped into various categories. Sequences that are shared due to common ancestry are classified as *homologs*. Homologs can be broken down into *orthologs* and *paralogs* based on the way they formed. Orthologs are caused from a speciation event while duplication events cause paralogs (Hartwell 2004). Therefore, multiple matches of a sequence to a single genome are usually considered paralogous sequence.

There are various programs for acquiring sequence similarities. BLAST (Bedell 2003), basic local alignment search tool, is a common tool used by biologists in genomic research. This tool takes genome sequences obtained from genome libraries and determines the similarity between the genomes based on nucleotide or protein matches between the two genomes. Data from the BLAST tool was used to draw the two visualizations used in this study.

3.3.3 Mauve Visualization

Mauve is a tool used to identify and show homologous sequence between genomes (Darling, Mau et al. 2004). Its visualization is similar to a parallel coordinate

diagram. Mauve uses this technique to match the whole genome of nucleotide sequence between multiple genomes as blocks of correlates. We use a version of the Mauve software package modified here at the University of New Hampshire (UNH) (Bancroft 2006). This version of Mauve uses BLAST to find homologs between two genomes. It then removes the paralogs, leaving only orthologs, or regions that are only found once between the two genomes (excluding duplications across genomes). It identifies these matching sections of the nucleotide sequence as Multiple Maximally Unique Matches (multi-MUMs). It then groups these multi-MUMs into closely related regions called Locally Collinear Blocks (LCBs) (Darling, Mau et al. 2004). These blocks are displayed on the screen with connecting lines to identify the shared sequence. The visualization can be seen in Figure 19. The boxes are randomly color coded to help distinguish between different blocks. Mauve can progressively simplify the visualizations by creating higher levels of abstraction for regions of the genome, but each level of abstraction loses information.

3.3.4 Dot Plot Visualizations

The dot plot methodology is a classic visualization technique for comparing nucleotide sequences which was initially referred to as *graphics matrix analysis* (Maizel and Lenk 1981). Data groups are plotted on the x and y axes like a scatter plot. Researchers have identified different patterns a dot plot can produce including: no features, diagonals, broken diagonals, squares, diagonal texture, and square texture (Thomson, Howard et al. 2006). [1]

1 Dot plots have also been used to compare news stories, lines of source code, and DNA sequence (Church and Helfman, 1993).

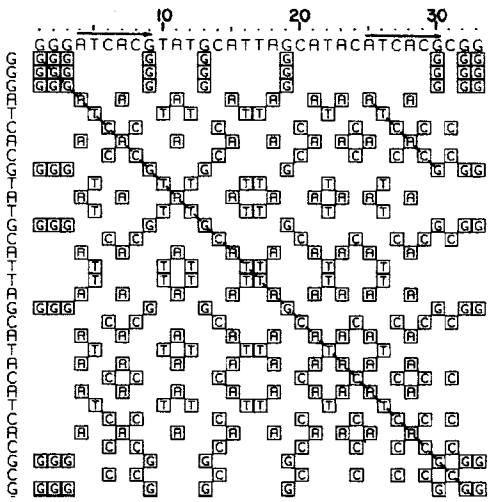


Figure 17: Illustration of Dot Plot Method. Notice the diagonal representing the sequence similarity. (Maizel and Lenk 1981)

For genome comparison, nucleotide sequences are essentially plotted on the horizontal and vertical axes. The graphic matrix marks each location in the matrix where the nucleotide bases are the same (Figure 17). Some common features that can be highlighted in genomics are homology and large areas of duplicated DNA (paralogy).

3.4 Visualizations

In this section we discuss the visualizations used for our study including a brief presentation of the images' attributes.

3.4.1 Dataset

We have identified three closely related strains of a *Burkholderia* bacterium to use in our study. Bacteria tend to have smaller denser genomes than more complex organisms. Even with the smaller genomes, the amount of data generated is still quite large, but manageable. Various strains of *Burkholderia* have very recently diverged but have very different functionality; some are extremely pathogenic while others are benign or even beneficial (Mahenthiralingam, Urban et al. 2005). Helping identify the homology of the strains in an effective manner will be very beneficial to the research of

these organisms. Our goal was to evaluate the effectiveness of two visualization techniques to identify the homology in this data. We used three strains in the *Burkholderia cenocepacia complex* (*Bcc*) genomes, *B. cenocepacia AU1054*, *B. cenocepacia HI2424*, and *B. cenocepacia J2315*. *Bcc AU1054* and *Bcc HI2424* are within the same clad and therefore are more closely related.

The genome data is represented as sets of strings of letters (A, C, G, and T). In a finished complete assembly, there is one sequence for each chromosome. Most assemblies are not finished, however; in this case, the sequences are called *contigs* and each represents a contiguous region of the genome. In both visualizations, the sequences are concatenated into a single string representing the entire genome. However, there is no expectation that neighboring contigs in the layout are in fact neighbors in the actual genome.

3.4.2 Visualization Methods

Our goal was to compare visualization techniques based on the scatter plot method and the parallel coordinate method as implemented in Mauve. In order to obtain a consistent comparison, we created a version of scatter plot with the same homologs identified by a nucleotide BLAST in the Mauve program. We then were able to run a small heuristic evaluation with members of the genomics community at UNH to compare the visualizations.

3.4.2.1 Scatter Plot Implementation

We implemented a scatter plot visualization similar to that of Rasko (Rasko 2005). Our scatter plot tool plots two genome sequences on each axis like the dot plot method. However, with an entire genome comparison, the plots can become cluttered

when matching base by base. Therefore, instead of matching at the nucleotide or protein level, we reduced the clutter by only plotting the similar sequences selected by BLAST, based on a moderate similarity measure.

This visualization was developed using an open source software package called JFreeChart (Viklund 2005-2007). This package can create a variety of elaborate graphs and charts using the Java programming language. Producing a scatter plot graph of the data consisted of two steps. The first step converted the data obtained from the BLAST results into a simple x, y data file representing the locations of pairs of sequences that were identified as similar. The second step read this data file for the scatter plot graph provided by the JFreeChart package. Horizontal and vertical lines were added to the graph to show the division of chromosomes. Finally, the scale of the axes of the scatter plots was varied between 10, 100 and 1000 base pairs per dot. After observing the resulting scatter plots, we determined that the 10 base pair graph provided the right mixture of detail and clarity.

3.4.2.2 Scatter plot Features

The scatter plot images used in this study are shown in Figure 18. Notice that when a sequence is homologous and sequential it forms a straight diagonal line (Figure 18A). The more differences in the genomes, the more jagged the lines appear and the more breaks in continuity (Figure 18B-C). A slope in the negative direction is a match of reverse sequence. The chromosome breaks are the horizontal and vertical darker lines on the graph. Paralogous sequence shows up as lines that overlap the same x or y region.

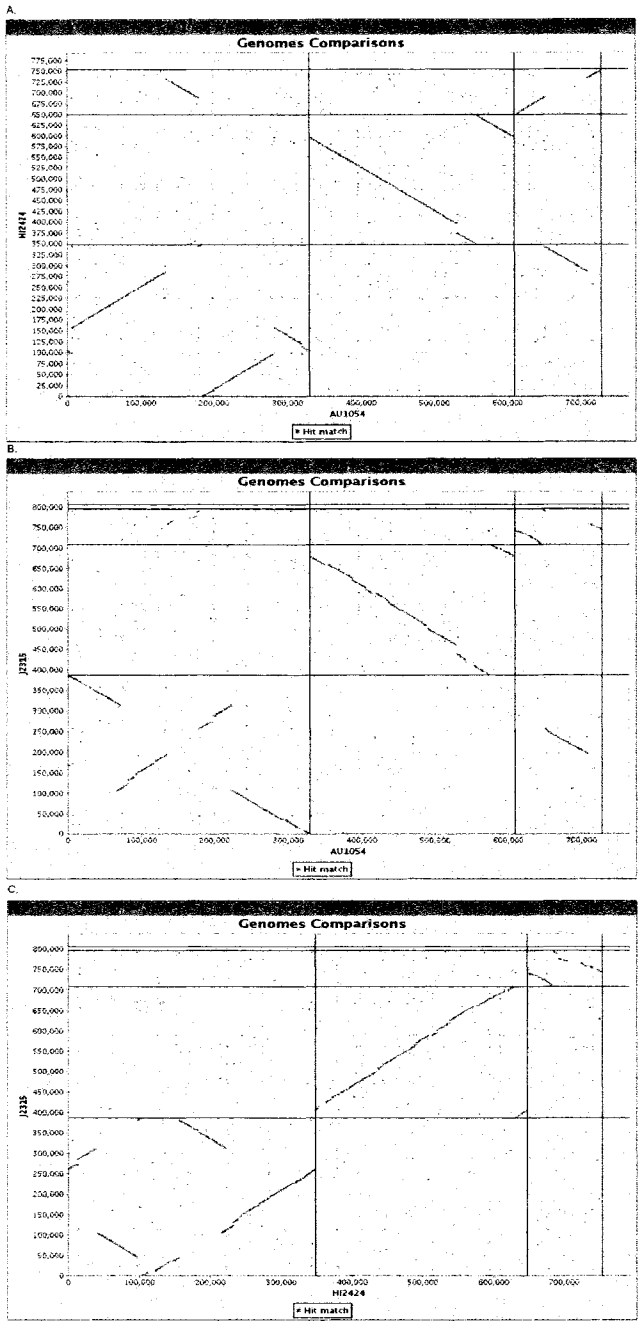


Figure 18: Scatter Plot Visualizations. (A) AU1054 vs. HI2424 (B) AU1054 vs. J2315 (C) HI2424 vs. J2315

3.4.2.3 Parallel Coordinate Features

The images used in this survey can be seen in Figure 19. The inverted sequences can be seen in the blocks below the line in the second genome. The large blocks show segmental changes in the genome well (Figure 19A), while small segments of similarity are more difficult to see (Figure 19B-C). In Figure 19C, the blocks on the right create many crossing lines that muddle the arrangement. The more blocks and rearrangements you have, the more cluttered the lines become. Color aids in separating the blocks that are next to each other.

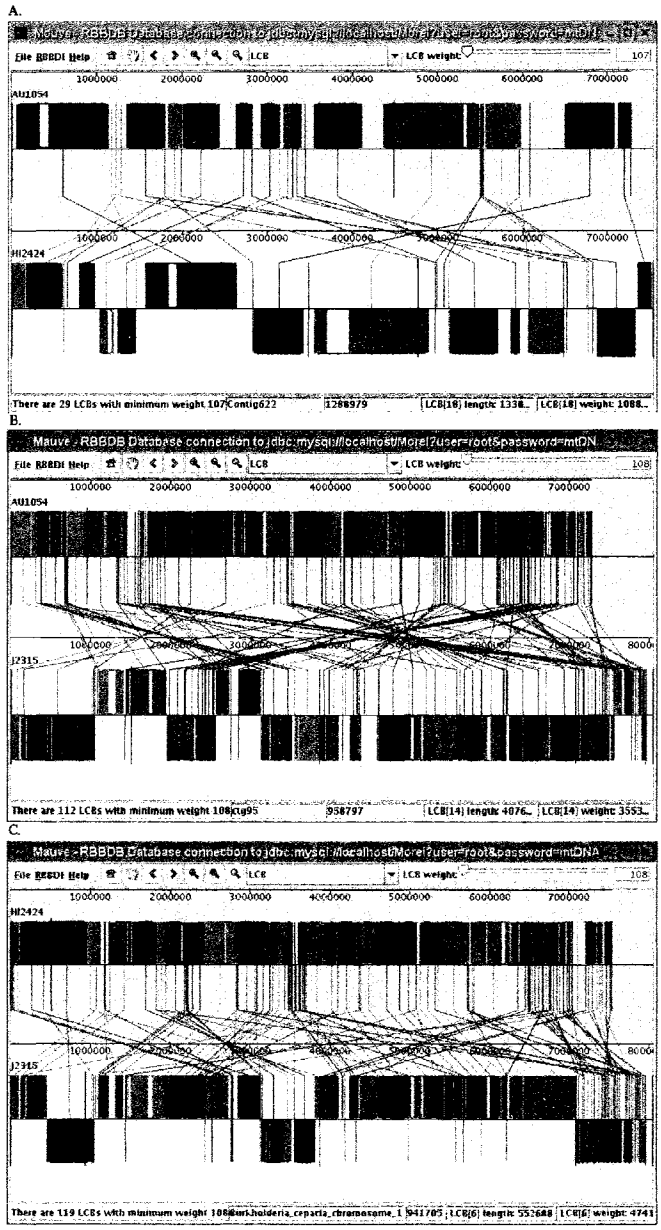


Figure 19: Mauve visualization of the three pair-wise comparisons of the Burkholderia Genomes. (A) Bcc AU1054 vs. Bcc HI2424. (B) Bcc AU1054 vs. Bcc J2315. (C) Bcc HI2424 vs. Bcc J2315.

3.5 User Evaluation Study

In this section we describe the heuristic evaluation that we performed.

3.5.1 Participants

The methodology employed in this study amounted to a small survey of professionals in the field of bioinformatics. The seven subjects included computer scientists, biologists, geneticists, and biochemists. Each subject was supplied a two part survey and asked to fill it out to the best of their ability.

3.5.2 Survey

The first part of the survey involved a task requiring the subject to establish a phylogenetic tree (Figure 16) among three unknown genomes labeled A, B, and C to conceal their identity. The subject was given a brief introduction to the project and a basic description of how the different visualization techniques represented the genomes. The subject was then given three visualizations of the *Burkholderia cenocepacia* complex genomes: *Bcc AU1054*, *Bcc HI2424*, and *Bcc J2315*. These genomes belong to a family of closely related bacteria. The subject was asked to draw a phylogenetic tree based on the three scatter plots (Figure 18), identifying the more closely related species. Then he/she is provided with the parallel coordinate visualization from Mauve (Figure 19) of the same three genomes. After providing the subject with time to form the phylogenetic tree based on the parallel coordinate visualizations, the subject moves on to the second part of the survey.

The second part of the survey provided the subject a scatter plot and a parallel coordinate diagram of the two most closely related species, *Bcc AU1054* and *Bcc HI2424* genomes, for a side-by-side comparison. The subjects were asked a series of generic

questions including which graph is more effective at various tasks, the advantages/disadvantages of each, and the subject's opinion of which is better overall.

3.6 Results

In this section we present the results from the study and provide preliminary analysis based on them.

3.6.1 Phylogeny Prediction Test

For the task of constructing a phylogenetic tree for the three unidentified strains, the subjects were more successful at recognizing the more closely related genomes when using the parallel coordinate method. Only four of the seven subjects identified the correct phylogeny using the scatter plots as a reference, while six subjects were able to identify the correct phylogeny using the parallel coordinate technique, generally, in significantly less time. However, this may be a result of the order in which the graphs were presented. It could also be a result the very simple genomes and therefore favorable for the parallel coordinate diagrams.

3.6.2 Survey Results

The subjects were then asked to evaluate both visualizations. Each of the visualizations has its own advantages and disadvantages. Therefore, each subject was asked a series of questions to try and quantify them. Because of the small scale of the study, it is not possible to infer significantly valid conclusions. However, we have found the subjects' observations interesting and report them as anecdotal results.

Six of the subjects said that the scatter plot was the visualization that was better overall. All the subjects stated that our version of a scatter plot was better at showing chromosome breaks and hits in the reverse direction than the parallel coordinate diagram.

Four subjects identified a disadvantage of the scatter plot as the difficulty in noticing the small changes and small scale differences in the genomes. The remaining critiques were suggestions for improvements to the visualization, such as adding color and clearly showing the boundaries between contigs. The scatter plot advantages were that it is easy to obtain a big picture and it is clear and easy to understand once you know how to read them. Also, some other comments included, "seems like it has more information" available and "probably more powerful."

All subjects indicated that the drawback of the parallel coordinate diagram was the confusion resulting from the crossing lines. One subject stated that there was "too much clutter in the center of the diagram." Another subject noted that the diagram caused "too much stimulation with the colors, lines, breaks, etc." There were some expressed advantages for the parallel coordinate diagrams. For example, it "shows long common sequences well." One subject also indicated that it shows the "horizontal comparison" or side-by-side comparison well.

3.6.3 Observations Based on Anecdotal Comments

The scatter plot visualization method is the more traditional method used for genome comparisons, while the parallel coordinate method has appeared recently. To the amazement of many of the subjects, the scatter plot was the method they preferred, even though it is older and takes longer to learn how to read. Once you learn how to use it, it seems to show much more information, and has the capability of being more powerful and descriptive. The parallel coordinate method appears more intuitive at first. One subject stated that it "seems easier at first." It shows position in the genomes more naturally. However, it appears to be more limited when showing detail.

Small breaks appear less obvious using the scatter plot diagram and can be easily masked as a long hit in the visual. This was not a problem for parallel coordinates because matching sequences could be identified by their matching color. Color also helps to show long contiguous sequences and complex rearrangements. For these reasons, one subject indicated that the parallel coordinate diagram could be "superior to the scatter plot when the re-arrangements are complex." These perceived advantages of the parallel coordinate visualization are all based primarily on its effective use of color, whereas our scatter plot implementation does not use color at all. These advantages would be significantly reduced or disappear with a relatively minor enhancement to the scatter plot visualization: we could assign a color to each dot based on the BLAST hit from which that dot was generated. With this change, it would be much easier to see breaks between hits, long contiguous sequences, and large rearrangements.

On the other hand, the parallel coordinate approach suffers fairly quickly from excessive clutter as the data size increases (Pillat 2005; Dwyer 2006). The number of crossing connecting lines grows rapidly as the evolutionary distance between genomes becomes larger. One subject stated that the problem with the parallel coordinate representation was that with a more complex comparison the "diagram becomes too busy to make sense of."

Although, we initially thought that the directionality of the sequence hits would show more information in the scatter plot, this function can become an obstruction since it breaks up the similarities of the genomes. One person's comment suggested that the "directionality" of the lines "hinders the visualization of similarities." Parallel coordinate diagrams have the advantage of keeping the hits next to each other for this purpose.

The scatter plot has the advantage of showing paralogs better than parallel coordinate diagrams. Parallel coordinates becomes even more confusing when you add paralogs to the visualization. Many subjects asked about paralogs in the visualizations and liked being able to see them in the scatter plot.

Although overall the scatter plot was preferred, many of the subjects suggested that the visualization they would choose depends on what they were looking for. Most of them wanted the option of looking at both.

3.7 Future Work

The case study needs to be expanded to include more subjects. An increased subject size would help analyze the qualities of the two visualization techniques in a more general and significant way. Also, randomizing the order in which the phylogenies question is distributed would help the significance of the results.

The case study also needs to be run on more complex genome comparisons. The current study used fairly closely related, simple, single celled organisms. There are much more complex genomes in the animal kingdom that could provide different results when comparing the two visualization techniques.

We did not test the current interactive capabilities of these visualizations. Interactively, Mauve can show as far down as the nucleotide level. Also, clicking on a block aligns the connecting segments. Our scatter plot currently has a zoom feature, as well, but not down to the nucleotide level. This feature could be added to the scatter plot.

A suggestion was made to identify contig boundaries along with the chromosome boundaries. There were identifiable breaks in the scatter plot method that looked to be contig boundaries; it would be beneficial to be able to verify when these breaks are

caused by contig boundaries. It could be especially helpful to have either automatic or user-assisted reordering of the contigs to be able to show more continuous similarity in the genomes. This reordering could also be beneficial for the parallel coordinate method by reducing the number of crossing lines.

Multiple genome comparisons could be very helpful in the visualizations. Mauve already can show multiple genomes laid out in parallel with connections between neighboring pairs. The scatter plot can only have two-by-two comparisons on a single graph, but could be extended using a scatter plot matrix approach (NCBI 2007; NIST/SEMATECH 2007).

Scatter plot has the potential of showing more information such as bolding to identify sections that are continuous hits or showing paralogs more easily. Color could also be used to distinguish breaks in hits. Colors could be added to the visualization to add more detail. For example, Colors could be mapped to the different bases in the nucleotide sequence: A (Adenine), T (Thymine), C (Cytosine), and G (Guanine).

3.8 Conclusion

This study compared the advantages and drawbacks of parallel coordinate and scatter plot methods for comparing similar sequences in different genomes. We performed a heuristic evaluation with three closely related genomes of the Burkholderia species. Subjects were asked a series of questions related to two different visualizations expressing the same data.

The scatter plot diagram was strongest at showing chromosome boundaries, reverse directions, and paralogs. The parallel coordinate diagram appears to be better at determining phylogenetic relationships and breaks in continuity but this could be an

artifact of the nature of the study. Although, it appears that the scatter plot approach is probably more effective than parallel coordinates for visualizing two genomes, this may not hold when comparing three or more genomes simultaneously.

CHAPTER 4

DSNP PROJECT: THE DAPHNIA PULEX SINGLE NUCLEOTIDE POLYMORPHISM (SNP) PROJECT

4.1 Abstract

In order to identify single nucleotide polymorphisms (SNPs) within the *Daphnia pulex* genome, we developed a pipeline of analyses that uses the comparative assembly of whole genome shotgun reads against reference scaffolds to conservatively estimate sites of true polymorphism within The Chosen One (TCO), the isolate of the *Daphnia* Genome Project. Here, we offer a first pass overview of the genome-wide level of polymorphism and identify a large number of variable sites in the ecological model organism, *Daphnia pulex*. We also provide some analysis of the SNPs detected by our pipeline. This paper focuses on the programs produced for our SNP detection pipeline. Further discussion and analysis of results can be found elsewhere.

4.2 Introduction

The DNA structure that encodes genomic data is composed of a sequence of complementary base pairing of four different nucleotides: Adenine, Thymine, Guanine, and Cytosine. Diploid organisms have two copies of each chromosome: a maternal and paternal copy. Alleles are gene variants on the chromosome that may code for a

particular trait. In diploid organisms, the allele pairs from each chromosome make up the organisms genotype. This allows for homozygous (same alleles) and heterozygous (different alleles) sites in the genomic makeup. Variable sites in the DNA sequence at the nucleotide level of individuals within a population are called single nucleotide polymorphisms (SNPs). These base changes contribute to variation in individual characteristics within a population and species.

Since single nucleotide polymorphisms (SNPs) are a fundamental aspect of genetic variation within a population, they are indispensable tools for genetic research (Hartwell 2004). Variation associated with diseases, disorders, and traits of interest can be mapped using SNPs as markers, and can be used in gene discovery, positional cloning, and medical diagnosis (Hartwell 2004). Additionally, SNPs are important for population studies and evolutionary research, as SNP patterns (haplotypes) are the basis of tracking gene flow among populations (Hartwell 2004). In fact patterns of polymorphism across a genome can leave clues about evolutionary forces acting on the genome.

Daphnia pulex was chosen for whole genome sequencing based on its proven utility as an ecological model organism. The potential to decipher ecologically relevant genetic variation has been a selling point of the *Daphnia* model (Colbourne, Singan et al. 2005).

Here, we outline a series of programs and scripts used to detect SNPs in *Daphnia pulex* Gnome by generating conservative estimates of variable sites on a scaffold-by-scaffold basis. For genome projects, *contigs* or continuous sequences are formed and then assembled into *scaffolds*. Because the genomic DNA for the *Daphnia* Genome Project was prepared from a clonal population started from a single, low-heterozygosis

individual (DGC 2007), this study is equivalent to an assay of heterozygosity within an individual *Daphnia*. However, heterozygous sites within a diploid individual represent segregating alleles of the larger population, and thus, with a first pass of SNP detection, we are able to describe some trends and patterns of genetic variation across the whole genome of the species *Daphnia pulex*.

4.3 Pipeline

We developed a pipeline for SNP detection in the *Daphnia pulex* genome (Figure 20). The basic concept of the design was to align the shotgun sequencing reads to each of the scaffolds. Because each read could come from two alleles, the variation within the reads that aligned to any given locus should give us an accurate measure of variation across the genome. Finally, a set of strict criteria were applied in order to improve accurate SNP detection.

4.3.1 Poor Quality Trimming

Since error can occur when creating the shotgun sequencing reads the reads were trimmed for poor quality. Therefore, mostly high quality sequence is used in our analysis. Lucy (Chou and Holmes 2001) is a program that removes vector sequence and identifies poor quality sequences of the 5' and 3' ends. Another program was then used to trim these ends from the read sequences. We had to create a program to trim the quality scores of the reads as well, in order to use the quality scores for the subsequent alignment of raw reads to the reference scaffolds.

4.3.2 Alignment and Assembly

The AMOS Comparative Assembler (Pop, Phillippy et al. 2004) is a program that aligns the sequence reads to each of the reference genome scaffolds. The AMOS

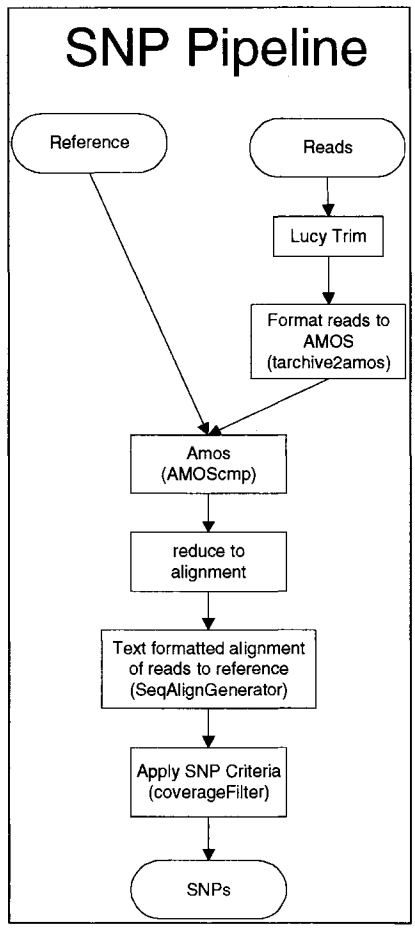


Figure 20: SNP Pipeline

program can take the quality score files as input to improve the alignment quality. The AMOS process involves building the reads into contigs or contiguous sequences of overlapping reads. These contigs are then aligned to the reference sequence. We set a 98% pairwise match requirement in order to assure high quality alignments and reduce paralogous misassembly. AMOScmp produces many files that are used in the following steps.

4.3.3 Reduce Alignment

Since we are only using a single scaffold as a reference for each assembly, some of the contigs formed by the reads do not align to the reference sequence. In order to attain an accurate placement of the reads, we needed a file with only the reads used in the

final alignment. We used a custom program that reduces the assembled reads file (delta format) to those reads only in the contigs that aligned to the reference (a reduced delta file).

4.3.4 Site by site Formatting

The SeqAlignGenerator (Kulkarni 2007) program takes the AMOS output file and produces a single file that formats the alignment information by scaffold reference positions. This file includes for each location in the reference: a reference position number, the reference base, and the list of nucleotides of the reads that align there.

4.3.5 Insertion Fix

Since the SeqAlignGenerator was creating for a project that was not interested in indels (insertions and deletions with respect to the reference), we had to incorporate an addition to the SeqAlignGenerator that includes information from reads that do not have a reference for that base but that spanned that location. In other words, we wanted to include the number of reads that were missing a base, with respect to the reference, at each insertion location. We therefore created a script to insert a '-' for every read that spanned a position but that did not have a nucleotide for that position.

4.3.6 SNP detection

We created a program (SNPfilter) that reduces the positions to ones that pass our SNP criteria. Our SNP criterion includes:

- At least two nucleotides of each base at a site
- Maximum and minimum coverage values
- Only two variable bases at a given position

- Insertions with respect to the reference must have at least two bases for a given insertion site.

We require that there are at least two nucleotides of only two different bases in order for us to call the position a SNP. This helps us reduce the alignment and sequencing errors.

Our coverage cutoff set the maximum and minimum number of reads that aligned to a specific location. We give the user the option of keeping a maximum and minimum number of nucleotides that align to a given locus. These extreme values are based on an overall coverage analysis. The minimum coverage was used to reduce the number of positions that are not covered enough to give us a good SNP call. Because we required at least two bases of two different types for a SNP call, our minimum coverage was set at 4 bases. Our maximum coverage was used to reduce paralogous misassembly. We chose our maximum cutoff by evaluating the range of values that will cover 99% of the positions (Poisson with average = 8.79X).

Our last criterion was that there could only be two alleles at a given position. Since we are working with a diploid species and the genomic DNA was prepared from a single individual, only two alleles at a specific location is possible; the locations that contain more are likely sequencing errors or local paralogs. Therefore, we excluded any sites that had three different bases at the same site to further reduce the probability of base-call error, or paralogy, when identifying SNPs.

4.4 SNP Analysis and Results

The *Daphnia* Genome Project produced 2.7 million reads with an average length of 1011 base pairs (bp) (774 after LUCY trimming). After reducing our dataset with our coverage cutoffs and variable types, we were left with over 89 million sites spanning 103 scaffolds (Figure 21). After we applied our SNP criteria, we are left with a list of putative SNPs in the genome. We determined there were 207,197 putative SNPs giving us an average variation across the genome of 0.23% SNPs per analyzed site. The number of sites that were reduced along the pipeline can be seen in Figure 21. We developed various other scripts to analyze this data including: quantifying SNP types, variation

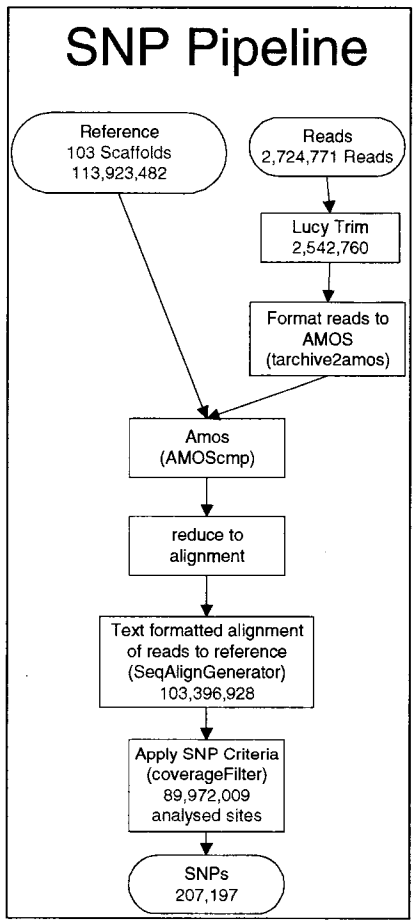


Figure 21: SNP Pipeline with numbers after each step

across scaffolds, SNP variation in coding and non-coding regions, segmental SNPs, and segregation of simple sequence repeats.

4.4.1 SNP types

SNPs are comprised of various types based on the varying bases (Figure 22). Insertions and deletions make up one type of variation called indels. We identified 90,592 indels or 43.7% of our SNP list. Another SNP type we detected was base changes: a/t, a/c, a/g, t/c, t/g, g/c. These can be grouped into transition (a/g, t/c) and transversions (a/t, a/c, t/g, g/c). Transitions make up 55,280 or 26.7% of SNPs. Transversions make up 61,325 or 29.6% of SNPs.

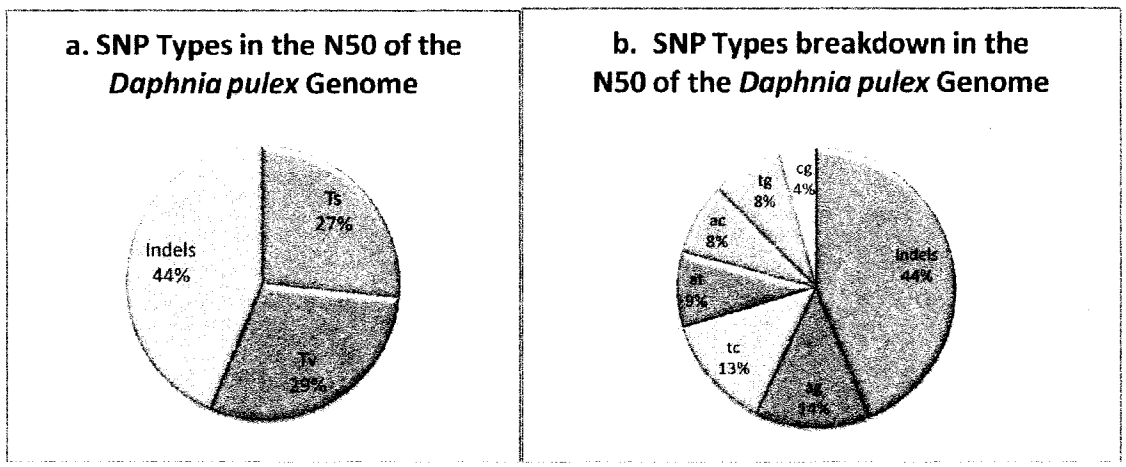


Figure 22: Distribution of SNP types in the Daphnia pulex genome. (a) Distribution of Indels, Transitions (Ts), and Transversion (Tv). (b) The SNP types broken down into each of the various base changes.

4.4.2 Variation between locations

We were interested in looking at SNPs between two specified positions. We created a script to determine variation between lists of genomic regions such as coding regions, genetic markers, and variation over base pair window sizes.

We investigated the diversity of SNPs within and outside coding regions in the genome. Exons are regions that most likely code for proteins. Introns are the regions in

the genome that are between exons in gene regions. Intergenic regions are regions that are between gene regions. We found that the exon regions in the genome have fewer SNPs than the average SNPs per genome. Exon regions have a SNP percent of 0.087% whereas the average over the entire genome is 0.2%. The amount of variation within and between coding regions can be viewed in Table 8.

	% SNPs
Exon	0.087
Intron	0.125
Intergenic	0.122

Table 8: Percent SNPs in various coding regions: exons, introns, and intergenic regions.

We were also interested in whether SNPs were clustered throughout the genome. We scanned the genome at various window sizes to see if the genome had high or low SNP frequencies over several ranges. We sampled high, moderate and low SNP density regions using BLAST to assess if these regions were due to paralogous misassembly.

4.4.3 Removal of Homopolymers and Microsatellites

Microsatellites are simple sequence repeats regions that are comprised of one, two, three, etc. base sequential repeats. Homopolymers are the sequential repeats of a single base. It has been stated that homopolymers and microsatellites have a higher level of mutations than other parts of the genome (Mahtani and Willard 1993). Consequently their heterozygosity level is expected to be high. We wanted to remove these regions from our analysis so we could obtain a better understanding of the variation levels of SNP in the genome. So we wrote scripts to remove sites between homopolymers and other microsatellites. The average size of a homopolymer or microsatellite was determined with an evaluation of the heterozygosity levels at differing lengths. This analysis can be read about in additional papers. After the removal of the homoploymer

and microsatellites, the SNP analysis would give us the rates of variation outside these highly variable regions.

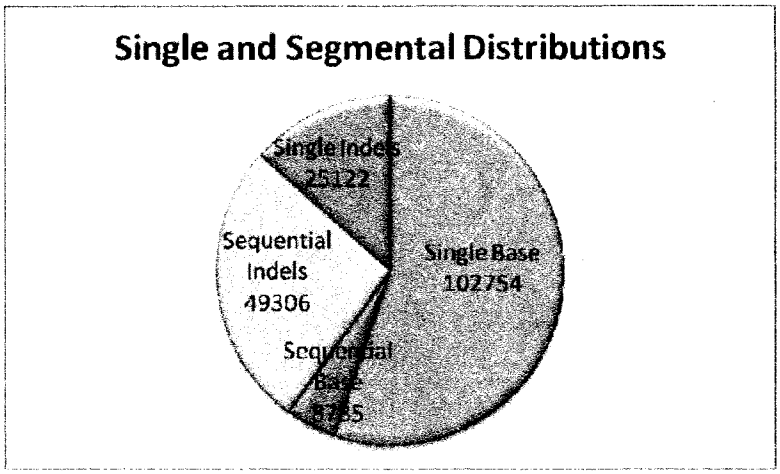


Figure 23: Single and segmental SNPs Distributions. This graph breaks down the amount of SNPs that are next to each other (segmental) and those that are only one location (single).

4.4.4 Sorting results into segmental and single variates

We also think segmental variations have a different rate than other regions in the genome. We, therefore, wrote a series of scripts to cluster sequential (2 or more) SNPs into single variation. We could then analyze the variation rates for each of the categories of SNPs (segmental and single). As well as giving us a better estimate of SNPs events (clustering segmental SNPs into a single event). We produced an output file that determined how many SNPs were in clusters of two, three, four, etc. as well. An overview of the distribution of single and segmental indels and bases can be seen in Figure 23.

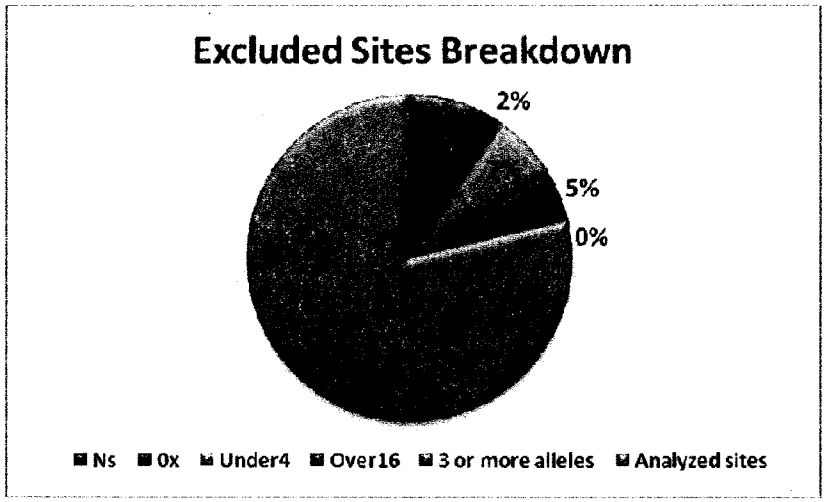


Figure 24: A breakdown of the excluded sites from our analysis

4.5 Discussion

We assembled the trimmed sequence reads against the largest scaffolds containing 50% of the sequence in the genome (N50). This included the first 103 scaffolds in our analysis. After applying all our requirements for excluding sites of paralogy and sequencing errors, we were reduced to less than a fourth of the sites (Figure 24). While exclusions were mostly due to undetermined sequence in the reference scaffolds (Ns), the effects of sequencing error and poor sequence quality were minimized with our initial trimming and rejecting of ambiguous sites. Therefore, the SNP calls reported here have a high probability of being true sites of variation.

We verified our analysis by checking some of our SNPs by blasting regions around SNPs to the scaffolds. We were able to identify these regions were the reference base by this method. We also checked some of the SNPs by using the AMOS visualization tool, bankViewer. We also checked a couple of the sites with PCR.

4.5.1 Criteria of Two bases

Since we required two bases of each SNP base then we are looking at an error rate for detected SNPs as one in 100 Mbp. For instance, a base with a quality score of 20 has

a 1% probability of being an incorrect call. The probability of two independently sequenced, orthologous bases, each with a quality score of 20 having the same sequencing error is 0.0025%. If the average quality score in our data set was 20, we would expect over 600 incorrect SNP calls over the 24 million analyzed sites just based on sequencing error. However, after quality trimming, the average quality of base calls is much higher (~40), making the expectation of false SNP calls, based on sequencing error alone, less than 1 for our entire data set.

4.5.2 Paralogy filter

In order to reduce our paralogy positions in the genome, we set our maximum coverage for a specific position to be 16X. We were able to determine the average coverage of reads at each location. The average coverage using the LUCY-trimmed reads was 8.79 X. We were then able to determine the maximum coverage by using the Poisson distribution of the average coverage over the genome (Figure 25). Using this distribution, we determined the coverage value at which 99% of the genome would be covered at least once. This allowed us to arrive at a maximum coverage of 16X. Any coverage above this value would more likely be paralogous regions such as transposable elements and other misassemblies that could generate false SNP calls.

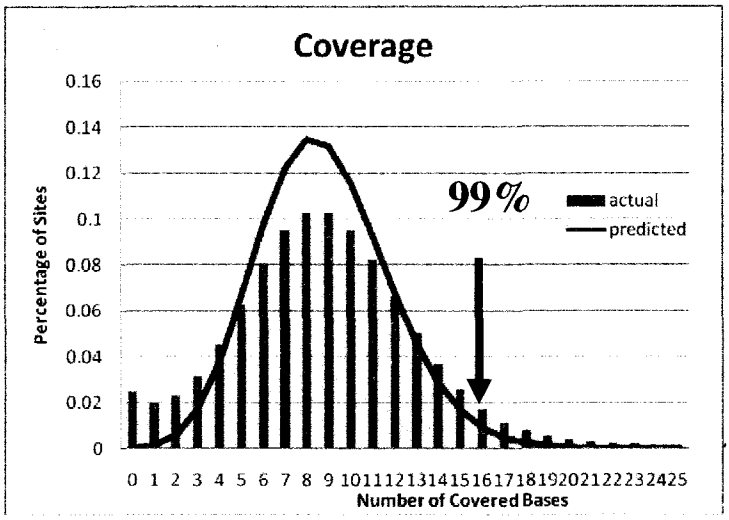


Figure 25: Average sequence coverage for the largest 103 scaffolds. The arrow shows the coverage at which 99% of the distribution would be included and corresponds to xmax = 16. Regions with coverage above 16 were therefore excluded from SNP analysis.

4.6 Conclusion

Data from this analysis provides an initial pass at the putative SNPs in the *Daphnia pulex* genome. The pipeline was designed to be conservative in its designation of SNP calls. Future work will be done to analyze further the individual variations in the genome and the significance of them.

4.7 Future Work

We intend to do future analysis on this data and provide the reader with more data analysis. This can be viewed in a paper presented by Abraham Tucker.

SNPs between individuals would provide more interesting discoveries in SNP detection. Comparing The Chosen One with the 1X covered Prince reads data set would provide us with an estimate of variation between strains.

LIST OF REFERENCES

Altschul, S. F., W. Gish, et al. (1990). "Basic local alignment search tool." J Mol Biol **215**(3): 403-10.

Bancroft, C. L. (2006). Magenta: A Reciprocal Best Blast Environment for Performing Multiple Genome Alignments, University of New Hamshire, Unpublished Work: 8.

Bansal, A. K. and T. E. Meyer (2002). "Evolutionary analysis by whole-genome comparisons." J Bacteriol **184**(8): 2260-72.

Bedell, J., Ian Korf, Mark Yandell (2003). BLAST.

Carrizo, S. F. (2004). "A Colour-Filling Approach for Visualizing Trait Evolution with Phylogenies." ACM International Conference Proceeding Series; Vol. 99, Proceedings of the 2004 Australasian symposium on Information Visualisation **35**: 117-126.

Chou, H. H. and M. H. Holmes (2001). "DNA sequence quality trimming and vector removal." Bioinformatics **17**(12): 1093-104.

Colbourne, J. K., V. R. Singan, et al. (2005). "wFleaBase: the Daphnia genome database." BMC Bioinformatics **6**: 45.

Conant, G. C. and A. Wagner (2002). "GenomeHistory: a software tool and its application to fully sequenced genomes." Nucleic Acids Res **30**(15): 3378-86.

Cooper, V. (pers. comm.). University of New Hampshire, Unpublished Data.

Darling, A. C., B. Mau, et al. (2004). "Mauve: multiple alignment of conserved genomic sequence with rearrangements." Genome Res **14**(7): 1394-403.

Delcher, A. L., S. Kasif, et al. (1999). "Alignment of whole genomes." Nucleic Acids Res **27**(11): 2369-76.

Dewey, C. N. and L. Pachter (2006). "Evolution at the nucleotide level: the problem of multiple whole-genome alignment." Hum Mol Genet **15 Spec No 1**: R51-6.

DGC (2007). Daphnia Genomics Consortium, The Center for Genomics and Bioinformatics, and The Trustees of Indiana University.

Dubchak, I. and D. V. Ryaboy (2006). "VISTA family of computational tools for comparative analysis of DNA sequences and whole genomes." Methods Mol Biol **338**: 69-89.

Dwyer, T., Seok-Hee Hong, Dirk Koschützki, Falk Schreiber, and Kai Xu (2006). "Visual analysis of network centralities." Proc Asia-Pacific Symposium on Information Visualisation **60 APVis '06**: 189 – 197.

Haas, B. J., A. L. Delcher, et al. (2004). "DAGchainer: a tool for mining segmental genome duplications and synteny." Bioinformatics **20**(18): 3643-6.

Hartwell, L. H., Leroy Hood, Michael L. Goldberg, Ann E. Reynolds, Lee M. Silver, Ruth C. Veres (2004). Genetics: from Genes to Genomes, The McGraw-Hill Companies.

Henley, M., M. Hagen, et al. (2007). Evaluating Two Visualization Techniques for Genome Comparison. Information Visualization (IV), Zurich, Switzerland, IEEE.

Hirsh, A. E. and H. B. Fraser (2001). "Protein dispensability and rate of evolution." Nature **411**(6841): 1046-9.

Inselberg, A. D., B (1990). "Parallel coordinates: A tool for visualizing multidimensional geometry." Proc IEEE Visualization (Vis '90): 361-378.

Koski, L. B. and G. B. Golding (2001). "The closest BLAST hit is often not the nearest neighbor." J Mol Evol **52**(6): 540-2.

Kulkarni, S. (2007). Comparative Genomics Exploration Tools. Computer Science. Durham, NH, University of New Hampshire. **Master of Science**.

Kurtz, S., A. Phillippy, et al. (2004). "Versatile and open software for comparing large genomes." Genome Biol **5**(2): R12.

Lerat, E., V. Daubin, et al. (2003). "From gene trees to organismal phylogeny in prokaryotes: the case of the gamma-Proteobacteria." PLoS Biol **1**(1): E19.

Lerat, E., V. Daubin, et al. (2005). "Evolutionary origins of genomic repertoires in bacteria." PLoS Biol **3**(5): e130.

Lynch, M., M. O'Hely, et al. (2001). "The probability of preservation of a newly arisen gene duplicate." Genetics **159**(4): 1789-804.

Mahenthiralingam, E., T. A. Urban, et al. (2005). "The multifarious, multireplicon Burkholderia cepacia complex." Nat Rev Microbiol **3**(2): 144-56.

Mahtani, M. M. and H. F. Willard (1993). "A polymorphic X-linked tetranucleotide repeat locus displaying a high rate of new mutation: implications for mechanisms of mutation at short tandem repeat loci." Hum Mol Genet **2**(4): 431-7.

Maizel, J. V., Jr. and R. P. Lenk (1981). "Enhanced graphic matrix analysis of nucleic acid and protein sequences." Proc Natl Acad Sci U S A **78**(12): 7665-9.

Markowitz, V. M., F. Korzeniewski, et al. (2006). "The integrated microbial genomes (IMG) system." Nucleic Acids Res **34**(Database issue): D344-8.

Michaels, G. S., D. B. Carr, et al. (1998). "Cluster analysis and data visualization of large-scale gene expression data." Pac Symp Biocomput: 42-53.

NCBI (2007). The Statistics of Sequence Similarity Scores. BLAST, National Center for Biotechnology Information.

Nielsen, J. (1994). Guerrilla HCI: Using discount usability engineering to penetrate the intimidation barrier. Cost-Justifying Usability. R. G. In Bias, and Mayhew, D. J. (Eds.), Academic Press, Boston, MA.

NIST/SEMATECH (2007). NIST/SEMATECH e-Handbook of Statistical Methods. http://www.itl.nist.gov/div898/handbook.

Overbeek, R., N. Larsen, et al. (2003). "The ERGO genome analysis and discovery system." Nucleic Acids Res **31**(1): 164-71.

Parke, J. L. and D. Gurian-Sherman (2001). "Diversity of the Burkholderia cepacia complex and implications for risk assessment of biological control strains." Annu Rev Phytopathol **39**: 225-58.

Pillat, R. M., E.R.A. Valiati, and C.M.D.S. Freitas (2005). "Experimental study on evaluation of multidimensional information visualization techniques." Proc Latin American Conf on Human-computer interaction: Cuernavaca, Mexico: 20-30.

Pontius J.U, L. W., G.D. Schuler (2003). "UniGene: a unified view of the transcriptome." The NCBI Handbook. Bethesda (MD): National Center for Biotechnology Information: 12.

Pop, M., A. Phillippy, et al. (2004). "Comparative genome assembly." Brief Bioinform **5**(3): 237-48.

Rasko, D. A. (2005). "Visualization of comparative genomic analyses by BLAST score ratio." BMC Bioinformatics **6**: 1471–2105.

Remm, M., C. E. Storm, et al. (2001). "Automatic clustering of orthologs and in-paralogs from pairwise species comparisons." J Mol Biol **314**(5): 1041-52.

Rice, P. L., I. and Bleasby, A. (2000). "EMBOSS: The European Molecular Biology Open Software Suite." Trends in Genetics **16**(6): 276—277.

Schwartz, S., Z. Zhang, et al. (2000). "PipMaker--a web server for aligning two genomic DNA sequences." Genome Res **10**(4): 577-86.

Sonnhammer, E. L. and R. Durbin (1995). "A dot-matrix program with dynamic threshold control suited for genomic DNA and protein sequence analysis." Gene **167**(1-2): GC1-10.

Sonnhammer, E. L. and E. V. Koonin (2002). "Orthology, paralogy and proposed classification for paralog subtypes." Trends Genet **18**(12): 619-20.

Tatusov, R. L., N. D. Fedorova, et al. (2003). "The COG database: an updated version includes eukaryotes." BMC Bioinformatics **4**: 41.

Thompson, J. D., D. G. Higgins, et al. (1994). "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice." Nucleic Acids Res **22**(22): 4673-80.

Thomson, N. R., S. Howard, et al. (2006). "The Complete Genome Sequence and Comparative Genome Analysis of the High Pathogenicity Yersinia enterocolitica Strain 8081." PLoS Genet **2**(12): e206.

TIGR (2005). Sybil: Web-based software for comparative genomics.

Viklund, A. (2005-2007). JFreeChart. http://www.jfree.org/jfreechart/.

APPENDIX

OPUS DOCUMENTATION

Installation:

- Install Java's JDK on the computer
- Install mySQL on the computer.
    - Follow the mysql instructions for installation including changing the root password (mysqladmin –u root password 'newrootpassword')
    - Set up a mysql username and password. Log into mysql as root (mysql –u root –p). Set up security measures and set up a new user replacing only username and password (keeping single quotes) with new usernames:
        - CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';
        - GRANT ALL ON *.* to 'username'@'localhost';
- Install NCBI's blastall
    - Set the user path to include the directory with blastall executable.
- Put Magenta's OPUS jar file and lib directory in folder in which you would like to run the file.
- In order to run subsequent programs, Perl, csh, and tcsh must be installed on the computer, as well as these packages.

Usage: Magenta OPUS

java -jar MagentaOPUS.jar
-U username (required arguments)
-P password (required arguments)
-D database (required arguments)
-f file1 file2 (names of sources or filenames) (required arguments)
BLAST parameters (optional)
-lerat leratPercentage (optional)
-best epsilonDifference (optional)
-h (optional help)

Possible BLAST parameters:
-p blastType
-e Evalue
-F filter parameter
-S strand

Detailed OPUS Usage Description:

Required Parameters:

The user must specify the username, password and database name that he/she wishes to connect to. This is the username and password for the mysql account. When a new database name is specified, the program will create the new database.

The file1 and file2 name of sources or filenames to use in the comparison. The first time a filename is used as an argument the program adds information for this genome to the database using the filename (minus the extension) as the source name. Once the data is in the database, subsequent invocations will treat the –f argument as a source (with or without the extension). These files should be in fasta format. If you use source names these names must already be in the database. In order to reuse files, the program should be run from the same directory and the same path and filename should be used. It is recommended that you rename your files if you want to use different BLAST parameters.

Possible BLAST parameters:

There are various different BLAST parameters that can be input (Bedell 2003). We have listed the ones that we have incorporated into our program. See blastall documentation for more detail.

-p blastType

The blast type parameter can be used to specify either *blastn* or *tblastx*. The default for this parameter is *blastn* for the OPUS program.

-e Evalue

The e-value will restrict the level of similarity in the BLAST hits. This value can be set as a minimum level of similarity. E-value is a number representing the probability that

the match was chosen at random. Therefore, the lower the e-value input, the higher the chance that the sequence was not acquired by random chance. The blast default for this parameter is 1E1.

-F filterParameter
The filter parameter can be set to help filter complex sequences. This parameter will help extend matches. There are three possible filter parameters: true (T), false (F), and masking parameters. True (default) turns filtering on. False turns filtering off. We often use the soft mask filter (mD – *blastn*, mS - *tblastx*) which extends matches and is recommended instead of turning filtering off.

-S strand
The strand parameter determines how many strands the program will look at. There are three possible strand options: 1, 2, or 3. The 1 strand option will only look at the single top strand. The 2 strand option will look at the bottom strand (reverse complement strand) for similarity. The default is 3 which will search both strands.

Optional Parameters:

-lerat leratPercentage
The Lerat optional parameter, allows the user to use the Lerat method for identifying homology, instead of the reciprocal BLAST method. The Lerat method requires a minimum Lerat percentage cutoff for matches. If you use the Lerat optional parameter, the first file (file1) in the -f command will be the query genome.

-best epsilonDifference
The best parameter allows the user to choose to separate the long syntenic blocks using our version of reciprocal best BLAST method. This will populate the best fields and reduce the multiple copy fields.

-h
This is the help command. Use this command to display the basic usage information.

Examples:
java -jar MagentaOPUS.jar -U mhenley -P password -D Burk –f
/home/mhenley/BurkGenomes/BurkAU1054.fsa
/home/mhenley/BurkGenomes/BurkHI2424.fsa -e 1E-10 -F mD

java -jar MagentaOPUS.jar -U mhenley -P password -D Burk -f
/home/mhenley/BurkGenomes/BurkHI2424.fsa
/home/mhenley/BurkGenomes/BurkAU1054.fsa -lerat 0.30 -F mD

java -jar MagentaOPUS.jar -U mhenley -P password -D Burk -f
/home/mhenley/BurkGenomes/BurkAU1054.fsa
/home/mhenley/BurkGenomes/BurkHI2424.fsa -best 1E-10 -e 1E-10 -F mD

Scripts

Annotation Package

The Annotation Package can be run with the script runRegionAnnotation.

Usage:

runRegionAnnotation <file> <path_nt_database>

Notes:

- file must have a .seq extension but the name must be entered without the extension.
- path_nt_database is the path to the directory holding the NT database on the computer. Download the NT database from NCBI to a directory on the computer.
- -h will give you usage help

Required Programs:

blastFilter.pl, getAnnotations.pl programs in the AnnotationPackage folder.

Semi-paralog Package

The Semi-paralog Package can be run with the script runSemiParalogs.

Usage:

runSemiParalogs <pathToRequiredPrograms> <genomeName1>
<genomeName2> <pathToFastasWithGenomeName1and2>

Notes:

- Genome Names must be equal to those in the orthoPair and paraPair output file names.
- -h will give you usage help

Required Programs:

semi-paralogFinder2.pl removeSingletons.pl, getFullGeneName.pl programs

In-out Paralog Package

The in-out paralog package will allow the user to differentiate different copy numbers (in-paralogs) of paralogs from similar copy number (out-paralogs).

Usage:

runInOutParalogs <pathToRequiredPrograms>
<MagentaParaPairsGenomesAxA> <MagentaParaPairsGenomesAxB>
<fastaFileA>

Notes:

- MagentaParaPairsGenomesAxA is a file that contains all the paraPair gene files from a single genome (A) against itself.
- MagentaParaPairsGenomesAxB is a file that contains all the paraPair gene files from a genome (A) to another genome (B).
- fastaFileA is the original genome (A) fasta file.

Required Programs:

in-out-paralogScript.pl, getFullGeneName.pl programs

OPUS Header Scripts

Output header files for OPUS sequence files which contain individual sequence information.

Usage:

getHdrsOPUS <file>

Output:

file_ortho.hdr, file_para.hdr, file_uniqueSeq.hdr

Required Programs:

getSeqHdr (written by RDB)