INVESTIGATING THE NATURE OF METAGENOMIC ORFANS: UNKNOWN PROTEINS OR ANALYTICAL ARTIFACTS?

by

Prasad Gajare

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Bioinformatics and Computational Biology

Fall 2014

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

INVESTIGATING THE NATURE OF METAGENOMIC ORFANS:

UNKNOWN PROTEINS OR ANALYTICAL ARTIFACTS?

by

Prasad Gajare

Approved: _____

Shawn Polson, Ph.D.
Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____

Errol L.Lloyd, Ph.D.
Chair of the Department of Computer & Information Sciences

Approved: _____

Babatunde Ogunnaike Ph.D.
Dean of the College of Engineering

Approved: _____

James G. Richards, Ph.D.
Vice Provost for Graduate and Professional Education

ACKNOWLEDGMENTS

I would like to sincerely thank my thesis advisor, Professor Shawn Polson, who guided me through every step of the project. I gained a lot of knowledge from his valuable tips, and would like to express my utmost gratitude for the support and advice he provided me during my course of completion of my Master's degree at the University of Delaware. Special thanks to Ryan Moore, Dan Nasko and Steven Smith for their time and help in providing a part of the dataset and processing some of the data. My sincere thanks to all members of the Delaware Biotechnological Institute (DBI), at the University of Delaware.

I would like to thank Dr. Cathy Wu and Dr. Eric Wommack for serving on my thesis committee and providing me with their valuable insights and suggestions. I also express my gratitude towards all the Computer Science and DBI Staff and faculty members. Thanks to the University of Delaware CBCB Bioinformatics Core and Delaware Biotechnology Institute for providing computational infrastructure that was supported in part by Delaware INBRE (NIH NIGMS *2P20GM103446-14)*. A special thanks to Karol for his help. Thanks for the support for the VIROME project form the Gordon and Betty Moore Foundation (GBMF MMI 2732) and the National Science Foundation (NSF 1356374).

PRASAD GAJARE

University of Delaware

December, 2014

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Study of viruses is limited by inability to culture and lack of universal genetic markers (e.g. 16S rRNA). Shotgun metagenomics, simultaneous sequencing of all viral DNA from a sample, has emerged as an approach to overcome many of these limitations. However, analysis poses unique challenges due to fragmented sequences, gene structure, and viral underrepresentation in sequence databases. VIROME is a bioinformatics platform that simplifies viral metagenomic analysis and exploration. A key step is prediction of open reading frames (ORFs) from metagenomes. Despite comparison of these ORFs against several reference databases, a substantial number show no homology to previously observed proteins, thus classified as ORFans. This study characterized ORFans to determine if they represent unknown proteins, or may be artifactual. A BLAST was carried out comparing predicted ORFs from metagenomic samples on VIROME, against UniRef100 and MgOl environmental database releases since 2005. An increasing number of hits and decrease in ORFans was observed over the timecourse due to the growing number of proteins accounted for in the databases, indicating that some ORFans were real proteins. However, a significant number remain classified as ORFans. We studied these ORFans to find if any characteristics, distinguish them from non-ORFans. ORFans in general were observed to have lower ORF caller score and shorter read lengths than non-ORFans. The ORFan fraction was more likely to have over-representation of several kmers. Homopolymeric kmers were particularly overrepresented in 454 pyrosequencing

ORFans, potentially indicative of sequencing platform artifacts. We next assessed various ORF callers to determine if ORFs are being wrongly predicted. Three – MetaGeneAnnotator, MetaGeneMark and Orphelia – were applied to eleven viruses, both whole genome and shredded to simulate metagenomes. MGA had the best overall performance: precision (0.82), sensitivity (0.74). Precision results indicate a significant number of false-positives would be expected, and likely contribute to ORFans. Varying cutoffs filters for ORF length and ORF score was assessed and indicate increasing cutoffs does increase precision, but lowers sensitivity. The findings indicate that a significant fraction of ORFans are likely artifacts of sequencing platform and ORF caller. These false-positives can be managed by applying cutoffs, but lowered sensitivity must be balanced.

Chapter 1

INTRODUCTION

1.1 Background

1.1.1 Viruses

Viruses are small infectious agents that replicate only inside the living cells of other organisms (Koonin, Senkevich, & Dolja, 2006). They are the most abundant biological entities on the planet, with the majority affecting microorganisms (Edwards & Rowher, 2005). Measuring and identifying the community dynamics of viruses in the environment is complicated because less than one percent of microbes have been cultivated in laboratory. The evolutionary origin of viruses is not clear and is likely polyphyletic: some of them may have evolved from plasmids (fragments of DNA that can move between cells) and others are speculated to have evolved from bacteria. Viruses play key roles in evolution of cellular life, including acting as an important method for horizontal gene transfer, which in turn boosts genetic diversity and thus drives evolution (Edwards & Rowher, 2005). They also play a major role in aquatic ecosystems. A teaspoon of seawater contains about one million viruses (Shors, 2011a). Most of these bacteriophages are harmless to plants and animals and they play an important role in the regulation of saltwater and freshwater ecosystems (Shors, 2011a). Viruses also infect and destroy bacteria in marine microbial communities, and help in recycling of carbon in the aquatic environment. The organic molecules which

are released from the dead remains of bacterial cells contribute to fresh bacterial as well as algal growth (Shors, 2011b).

Viruses infect all forms of cellular life including Plants, Animals, Bacteria, and Archaea. Thus, we look at some of the types of viruses and the roles they play in the ecosystem such as bacteriophages about which we will talk in detail in the next section. Viral enzymes help in the breakdown of the cell membrane. In the case of the T4 phage, once the enzymes have been injected, in a span of twenty minutes more than three hundred phages are released (Shors, 2011b). Viruses also infect livestock in large numbers. Diseases such as foot-and-mouth disease and bluetongue are caused by viruses (Goris, Vandenbussche, & De Clercq, 2008). They infect plants by causing a loss of yield, but since preventing viral infections in plants is expensive, it is not considered profitable to try and control them. These plant viruses are often spread from plant to plant by organisms, known as vectors. Viral particles of plants are modified genetically for use in biotechnology, by enclosing the foreign material and later blending into supramolecular structures (Caranta, Aranda, Tepfer, & Lopez-Moya, 2011). Some viruses replicate within Archaea: these are mostly double-stranded DNA viruses with uncommon and peculiar shapes (Lawrence et al., 2009; Prangishvili, Forterre, & Garrett, 2006). Protection against these viruses includes RNA interference from repetitive DNA sequences that are related to the viral genes (Makarova, Grishin, Shabalina, Wolf, & Koonin, 2006; Mojica, Díez-Villaseñor, García-Martínez, & Soria, 2005).

1.1.2 Viruses of Microorganisms

Viruses of microorganisms, also known as VoMs, are the world's most abundant biological entity and includes viruses infecting domain Bacteria (bacteriophages), domain Archaea, algae, protists, fungi such as yeasts (mycoviruses), and the viruses of other viruses (satellite viruses) (Hyman & Abedon, 2012).They play important part in ecology, public health, infectious disease, and environmental science. They help in research on evolution of viruses and can be used to kill some antibiotic-resistant bacteria. The viruses of Bacteria, called bacteriophages or phages, the most common VoMs and in fact the most common type of viruses. Total number of bacteriophages exceeds about 10^{30}. Thus we can say, for every cellular organism (Whitman, Coleman, & Wiebe, 1998), more than one virus is present. There are two categories into which phages can be classified – their genome size and virion morphology (*Virus Taxonomy: Ninth Report of the International Committee on Taxonomy of Viruses*, 2012).

1.1.3 Viruses of Microorganisms in aquatic environment

The most common and varied viral group bacteriophages, are found abundantly in marine environments. There are about ten times more of these viruses in the seas than there are bacteria (K. E. Wommack & Colwell, 2000), reaching levels of 250,000,000 bacteriophages per milliliter of seawater (Bergh, Børsheim, Bratbak, & Heldal, 1989). As we have seen, though viruses of microorganisms (VoM) are found everywhere, the ones found in marine environments have been well studied while

environments such as the human microbiomes are relatively unexplored but are receiving considerable attention recently (Turnbaugh et al., 2007).

1.1.4 Human Microbiome

The human microbiome includes prokaryotes, viruses and microbial eukaryotes that occupy the human body. The National institutes of Health (NIH) started a project that primarily concentrates on discussing the diversity of microbial species related to health and disease. In the first phase, this project sequences hundreds of microbial reference genomes and couples them to metagenomic sequencing from numerous body sites (Nelson et al., 2010). The human microbiome project (HMP) tells us that we are organisms made up of human as well as microbial components. HMP helps us understand the two aspects of our microbial components - genetic and metabolic characteristics, and how they play a part in our standard physiology and a tendency to contract diseases. This project is an interdisciplinary one, combining the disciplines of medical and environmental microbiology as well as a global initiative receiving contributions from all over the world (Turnbaugh et al., 2007).

1.1.5 Identification of Viruses

Over the years, we have found a great deal of information on viruses and viral assemblages and the role they play in the biogeochemical cycles (Brussaard et al., 2008). In spite of these developments, an in-depth understanding of the viral infections

caused and the biological process responsible for it, still remain to be explored as most of this research has been restricted to a few known host viruses (Polson, Wilhelm, & Wommack, 2011). This does not completely account for the large and varied types of viruses found in the environment. When sequencing viruses, a number of unusual difficulties are faced which we do not encounter while sequencing cellular microorganisms. The free availability of DNA in the environment, cloned host cells getting killed by viral genes and modified viral DNA that cannot be cloned are some of the issues faced, and biases and challenges caused by low DNA concentration (Edwards & Rowher, 2005; Marine et al., 2011, 2014). In case of viruses, there are few genetic markers developed, none which are universal to all viruses and can be used a basis to get a universal understanding of viral infections and evolution. Current understanding is thus based on a limited subset of distinctive genes from different groups, which cannot be considered as sufficient representation for the wide viral community. One example is the widely found and preserved SSU rRNA gene (Polson et al., 2011).

1.2 Metagenomics

The problems faced while classifying and identifying viruses discussed above, have to some extent been overcome by the advent of viral metagenomics techniques. In Metagenomics, DNA of microorganisms are directly extracted and cloned from an assemblage of microoorganisms (Handelsman, 2004). It is also known as community

genomics and aids in the study of the ecology and physiology of microorganisms in the environment. This method consists of sequence-based and functional analysis of environmental samples obtained directly from water and soil and is mostly associated with eukaryotic hosts (Handelsman, 2004). Metagenomics has led to the discovery of some of the unique genes and gene products such as the first bacteriorhodopsin; some new small molecules which demonstrates antimicrobial activity and previously unknown new members of the known protein families (Handelsman, 2004). There have been recent advances in the understanding of marine phages. Ribonucleotide Reductase (RNR) enzymes are found largely in marine phages (>90 percent) thus making them an effective marker for these aquatic organisms (Sakowski et al., 2014). Also, it has been discovered that DNA polymerase A which is essential for replication of DNA can be used to predict the behavior of a phage – lysogenic or lytic (Schmidt, Sakowski, Williamson, Polson, & Wommack, 2014).

1.2.1 Challenges in Metagenomics

Analysis of metagenomes presents a number of challenges. The gene identification algorithms being used presently are most helpful to identify ORFs in bacterial or eukaryotic genomes. Identifying ORFs in viral genomes has several issues and considerable work needs to be done in this area. In case of viral metagenomes, many of the ORFs are not detected or get missed out as read lengths of viral metagenomes are significantly shorter than bacterial genomes and they are found in close proximity, at times overlapping each other.

When we consider bioinformatics analysis of metagenomic sequences, we are presented with several challenges because of the nature of viruses such as less information on viral proteins is found in databases. In case of larger read lengths too, much is not known about the gene products (Polson et al., 2011). This being more of a problem in case of environmental samples where newer and unknown samples are found frequently. Even ORFs which are found in large numbers, many a times do not have homologs in known databases like UniRef (Polson et al., 2011). Databases like SEED subsystems and similar ones used for annotation of microbial genomes, fail to include terms common to viral genomes. As the metagenomic data available is increasing in recent years, it is hard for computational resources to keep up with the pace (Polson et al., 2011). Upcoming bioinformatics tools have begun to allow processing and analysis of metagenomic libraries with the help of automated web and command line interfaces. Even though this is true, the large amount of data available and the individual algorithms or methods used by each of the analysis software, makes it necessary for a person using these tools to have a sound understanding of scripting languages and Unix commands, to find his way through even the simplest of analysis (Polson et al., 2011).

1.2.2 Viral Informatics Resource for Metagenome Exploration (VIROME)

VIROME (http://virome.dbi.udel.edu/) addresses some of the challenges faced in metagenomic analysis. It is a bioinformatics pipeline that classifies predicted open-reading frames ORFs obtained from viral metagenomes. This is done by obtaining results from a homology search carried out against both the known and environmental

sequences. UniRef100 database with five annotated sequence databases linked to it is used for functional and taxonomic information. For classification with reference to environmental database, MetaGenomes On-Line database (http://metagenomesonline.org/) is used, which contains a total of 49 million environmental peptide samples. VIROME plays three major roles – it serves as a bioinformatics analysis pipeline, is an efficient web-based visualization environment as well as a metagenomic annotation repository. This tool obtains data from three sources - three subject protein sequence databases, five annotated databases and CD-Hit 454. The VIROME bioinformatics pipeline is mainly divided into steps. In the first step, the sequences are filtered for quality based on a set of criteria such as to remove poor and duplicate sequences. The next step consists of analysis of these sequences which includes identification of sequences containing tRNA followed by BLASTP against the UniRef 100 and environmental database. The VIROME tool takes a single input file in either fasta, qual, fastq or 454 sequencing .sff format (K. Eric Wommack et al., 2012).

A web user interface is provided to access the VIROME pipeline and view the relevant results. This web UI receives information from the databases mentioned above (Bhasvar, Polson, Dhankar, & Wommack, 2009; K. Eric Wommack et al., 2012). The homology search results obtained by carrying out BLAST, are displayed in a comprehensive summary using charts like pie charts and bar graphs. VIROME categorizes sequences in a variety groups which in turn helps researchers in their analysis such as clustering and assembly. It gives an additional advantage by providing a web UI that helps to get predicted peptides, retrieve read sequences, obtain

predicted ORFs and an ability to sort BLAST results based on a variety of different specifications (K. Eric Wommack et al., 2012).

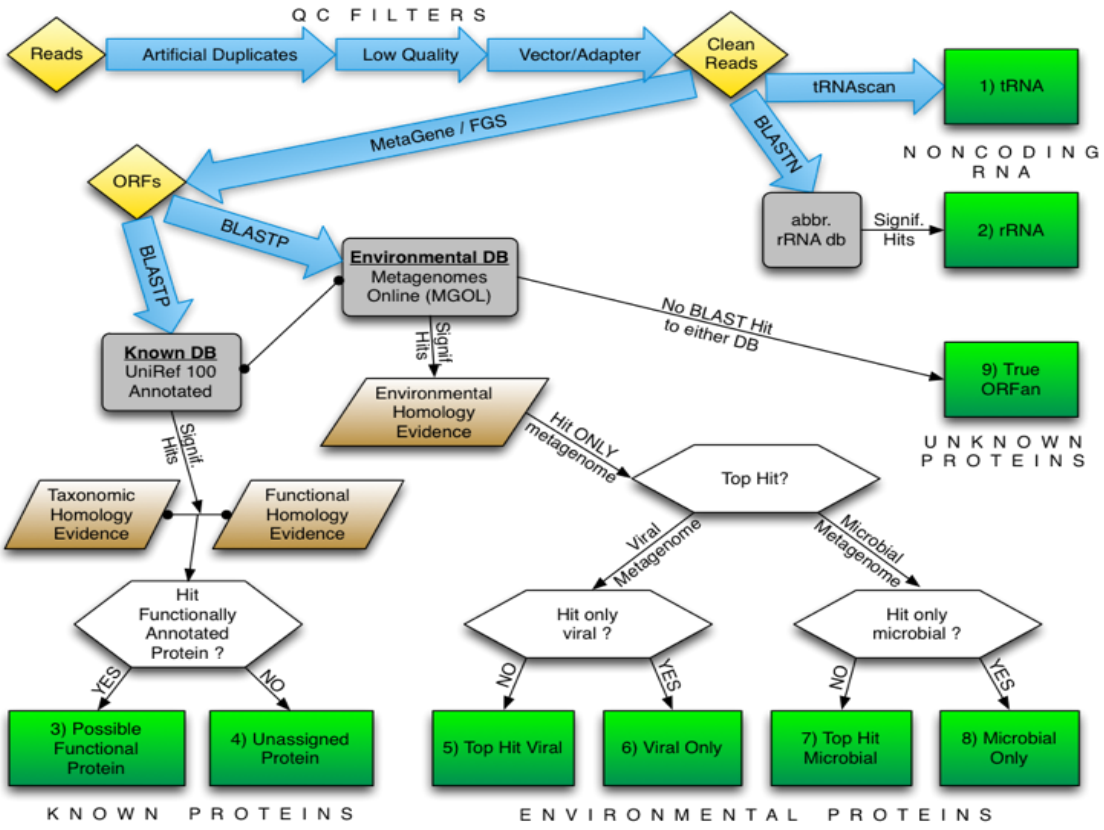


Figure 1.1 VIROME Pipeline

1.2.3 ORFs/ORFans

Open reading frames or ORFs are used to identify candidate protein coding regions in a DNA sequence. They are part of reading frame which has no stop codons. One common use of ORFs is in gene prediction. ORFs are used along with other factors, to initially identify candidate protein coding regions in a DNA sequence.

ORFans are open reading frames with no known homology to known databases or wrongly predicted ORFs by tools or ORF callers. They neither have references in metagenomic databases i.e. thy have never been seem before even in metagenomic related experiments. Here we concentrate on the ORFs predicted by the VIROME pipeline.

1.3 Metagenomic ORF Callers

ORF callers are tools that aid in prediction of genes from metagenomic sequences. Conventionally we can classify gene prediction programs into two different groups. One of these are the programs that are being used from earlier times - the ones that use known annotation for training models and predict unknown annotations from these trained models (Stanke & Waack, 2003). The second category of gene predicting programs use a reference database to find homologous sequences and try to find as close a match as possible for the input sequences (Yok & Rosen, 2011). Additionally there are some hybrid approaches that combine the traditional approach and some newer approaches have been suggested (Allen, Majoros, Pertea, & Salzberg, 2006; Pavlović, Garg, & Kasif, 2002; Shah, McVicker, Mackworth, Rogic, & Ouellette, 2003; Yada, Takagi, Totoki, Sakaki, & Takaeda, 2003). Unfortunately, it is not possible to use traditional gene prediction methods in metagenomics. Application of these methods to metagenomics for identification of open reading frames (ORFs) is a problem due to their small size, overlapping sequences and lack of known genes in reference databases for homology search. Therefore, recent tools have been developed to address some of these problems for metagenomic reads. Some of the ORF callers

being used for this purpose are: Orphelia (Hoff, Lingner, Meinicke, & Tech, 2009b), MetaGene (MG) (Noguchi, Park, & Takagi, 2006), MetaGeneAnnotator (MGA) (Noguchi, Taniguchi, & Itoh, 2008), GeneMark (Besemer & Borodovsky, 1999), MetaGeneMark (MGM) (Zhu, Lomsadze, & Borodovsky, 2010), FragGeneScan (Rho, Tang, & Ye, 2010b), Metagenomic Gene Caller (MGC) (El Allali & Rose, 2013) and Glimmer-MG (Kelley, Liu, Delcher, Pop, & Salzberg, 2012b). Here we review these ORF callers, the methods and algorithms applied by them and their respective pros and cons. At the end we compare a few of them with respect to their precision and sensitivity.

1.3.1 MetaGene (MG)

MetaGene (Noguchi et al., 2006) is an ORF caller that helps in gene predictions for a range of prokaryotic genes with an estimated sensitivity of 95% and a specificity of 90%, which has been tested on a dataset of artificial shotgun sequences. MetaGene consists of two sets of codon frequency interpolations for Bacteria and Archaea and the right one is selected for a particular sample automatically, with the help of domain classification methods. It uses a two stage approach. In the first step, the input sequence is taken and all possible ORFs are found. They are then scored as per their lengths and base compositions. Here, sequences of codons having a start and stop codon is defined as an ORF. Apart from these whole ORFs, partial ones are those that are located on the ends of given sequences or are the entire sequence. These partials ones are also extracted in the first step. As part of the second step, a scoring scheme is used that takes into account the scores of the neighboring ORFs and the

ORFs themselves as well as the orientation score. The best score is selected from these set of scores. The advantages of this two-stepped approach is prediction of overlapping genes. Different statistics are used for prediction in MetaGene such as ORF lengths, di-codon frequencies, orientation score, distances from neighboring ORFs, etc.

1.3.2 MetaGeneAnnotator (MGA)

MetaGene Annotator (MGA) (Noguchi et al., 2008) is an upgraded version of t MG which is used in gene prediction of metagenomic sequence data. MG predicts genes in two stages as discussed above. First, all possible ORFs are extracted from the input sequences and the ORFs are scored by their base compositions and lengths and the best score is found. As part of the second step, a scoring scheme is used that takes into account the scores of the neighboring ORFs and the ORFs themselves as well as the orientation score. The best score is selected from these set of scores. MGA was developed to overcome limitations posed by MG and improve its prediction accuracy. Di-codon frequencies represents conditional probabilities of occurrences of codons. MGA uses these di-codon frequencies for prediction. Along with di-codon frequencies, it uses GC content to group the genomes. The di-codon frequencies are then calculated for each of these groups. These improved features of the MGA give it an edge over other ORF callers and help in better predictions by not only prediction longer reads but also an ability to detect shorter ones.

1.3.3 FragGeneScan

FragGeneScan (Rho, Tang, & Ye, 2010a) uses a statistical model called Hidden Markov Model (HMM) and also sequencing error models. By using this approach it claims to carry out effective prediction even for shorter read lengths. FragGeneScan is able to identify ORFs in both complete genomes and shredded metagenomic sequences. Onbe of the criteria used by FragGeneScan to identify ORFs are that the length of the sequences should be longer than 60bp. Additionally, the genes should start with a start codon or in an internal region and genes should end in a stop codon or in a match state. When contrasted with other ORF callers, FragGeneScan stands out due to two distinct features. Firstly, it can find ORFs from fragmented metagenomes apart from whole genomes. Secondly, the reads generated from next generation sequencing methods have some frameshifts due to insertion deletion errors. FragGeneScan is able to correct these frameshifts errors.

1.3.4 Orphelia

Orphelia (Hoff, Lingner, Meinicke, & Tech, 2009a) is another ORF prediction tool for metagenomic sequences, especially for short fragments and sequences which have unknown phylogenetic origins. It uses a two stage approach. As part of the first stage, various features are obtained from the input sequences such as translation initiation sites and monocodon & dicodon usage. In the later stage, the sequence features obtained from previous step along with ORF lengths and GC content of the

fragment are used to construct a neural network. From this neural network, probability of an ORF for encoding a protein is found out.

Another important feature of Orphelia is its post-processing algorithm which makes use of the scoring scheme probabilities in order to find an overlap. The entire software implementation is in Java, while some of the earlier algorithmic processing is done using MATLAB and faster C code for functions that need to be executed in lesser time. Orphelia is show to have high gene prediction accuracy on shorter DNA fragments as compared to other ORF callers as well as high specificity for gene prediction.

1.3.5 Metagenomic Gene Caller (MGC)

MGC (El Allali & Rose, 2013) uses a machine learning approach which is very similar to Orphelia. It too uses a two-step approach. Unlike Orphelia, which constructs a single model, Metagenomic Gene caller constructs separate model for different GC-content ranges. Later, it applies the corresponding model to every fragment based on its GC-content. The training dataset is separated by the GC content with mutual exclusion which in turn helps train multiple models. This gives it an added benefit over Orphelia. Idea of GC-content usage was taken from the relation found between amino acid composition and nucleotide bias. The amount of GC content affects codon usage which on the other hand affects amino acid usage. GC content is also linked to genes length, genes with rich GC content are seen to be the longest in length while the ones poor in GC content are seen to be the shortest. Also, MGC uses two amino-acid

related features and this amino-acid usage helps in improving the overall efficiency of gene finder.

1.3.6 Glimmer-MG

Glimmer (A. L. Delcher, Bratke, Powers, & Salzberg, 2007; A. Delcher, 1999; Kelley et al., 2012b; Salzberg, Delcher, Kasif, & White, 1998) uses a statistical method called Interpolated Markov Models (IMM) for gene prediction. It uses two concepts – overlapping of prokaryotic genes and translation initiation sites (TIS). An ORF of particular length (above a certain determined threshold) is extracted and a score is obtained based on its log-likelihood ratio. Later on a dynamic programming approach is used to find the ORFs having the maximum scores, keeping in mind the condition that overlapping of genes cannot occur above a certain threshold. Glimmer has taken three extra features from MetaGeneAnnotator (MGA) – distance from neighbors, ORF length and gene orientation.

Clustering of sequences is done to categorize organisms that are most likely to fall in the same groups also called phylogenetic classification. In case of advanced Glimmer, it uses Phymm classification, instead of classification based on GC-content. In this approach, the classification is done by taxonomy and the Interpolated Markov Model is trained with respect to reference genome in the GeneBank. An unsupervised clustering method called SCIMM (Sequence Clustering with Interpolated Markov Models) is also used which is based on clustering. SCIMMS uses sequences belonging to assigned clusters to train IMMs. Next, it uses cluster IMM to score each sequence

and each sequence is then re-assigned to the respective cluster with highest scoring IMM.

1.3.7 MetaGeneMark(MGM)

MetaGeneMark (Zhu et al., 2010), which is a successor of GeneMark (Borodovsky & McIninch, 1992) applies a shotgun sequencing approach for gene prediction. A conventional algorithm for gene finding, applies a probabilistic model to genomic sequences which consists of protein and non-coding regions. In such cases, the gene prediction accuracy is dependent on the precision with which model parameters can be estimated, which in turn is specific to genomes. In cases of metagenomic sequences, it consists of short fragments where the main aim is to identify complete or partial protein-coding regions in short fragments.

In this ORF caller, firstly, genomes with known annotations were taken and analyzed. One genome was taken at a time and its frequency of occurrences was found in a set of annotated protein coding regions. Nucleotide frequencies seen in short DNA fragments can help to give an estimate of global nucleotide frequencies in the whole genome. This whole genome is in turn is a source of the short fragment. Taking the global nucleotide frequencies estimated in the earlier step as a basis, later on genome-specific codon frequencies were extracted.

1.3.8 Comparison of the ORF callers

Looking further into the ORF callers, some literature on the comparison was reviewed. In the paper "Combining gene prediction methods to improve metagenomic gene annotation" (Yok & Rosen, 2011), the authors talk about GeneMark (GM), MetaGeneAnnotator (MGA) and Orphelia. They compare the specificity and sensitivity of the three ORF callers individually as well as their logical combinations. They not only analyze the programs' performance efficiency at different read-lengths like done in similar studies, but also categorize reads into intra- and inter-genic regions, for analysis. For this study, their dataset consists of simulated samples of 28,000 artificial metagenome fragments from 96 genomes. These genomes include 19 different phyla, 14 Archaea species and 70 bacterial species. 4000 reads for each read length (100bp – 700bp) were considered.

They found that MGA had the highest sensitivity (for reads in the range of 200bp-500bp) while GM has the best specificity. Further on, the logical combinations of these tools were considered. The logical combination of GM & Orphelia has best specificity and lowest sensitivity. GM | Orphelia | MGA boosts sensitivity for gene prediction but has lowest specificity. Overall, for reads of length 100bp – 200bp, GM| Orphelia | MGA performs best. The consensus combination gives best result for reads with sizes in the range of 300bp – 400bp and GM & Orphelia performs best for 500bp – 700bp read lengths. Various algorithms demonstrate a trade-off between sensitivity and specificity at different read lengths. For all the ORF callers, we can see a clear reduction in sensitivity and specificity for shorted reads. Intersection of ORF callers seems to have given improved accuracies in terms of annotation but has given poor

prediction accuracies. Union of the methods improved prediction accuracies but resulted in poor annotation.

As seen above, this paper compares the different ORF callers as well as their logical combinations. However, the dataset used in the paper is completely bacterial data. Viral ORF have quite different characteristics than bacteria. They are significantly shorter, more densely packed in a genome and often overlap with each other. All these factors need to be taken into consideration while predicting ORFs. So further step was to carry out tests on a dataset of viral samples, directly obtained from the environment. As part of our study, we run the metagenomic ORF callers on a more realistic, wide and accommodating set of viral samples. Our aim is to see which performs the best in terms of prediction accuracy.

1.4 Thesis Overview

Earlier, we saw in detail the importance of viruses, the viral infections caused by them and the major role human microbiome project plays. It is essential to identify and classify viruses to know more about them. But studying them has various difficulties like their shorter lengths, lack of reference viral genes, varied characteristics of each virus, etc. One of the new and innovative approaches – Metagenomics overcomes many of these limitations and helps in efficient identification of viruses. Even metagenomic analysis faces bioinformatics related challenges like poor knowledge of viral proteins, lack of sufficient homologs in reference database, lack of computational resources required to analyze the huge

metagenomic data. VIROME is a bioinformatics pipeline that addresses many of these challenges and helps in prediction of open reading frames as well as servers as a metagenomic annotation repository. The ORFs predicted by VIROME when BLASTed against the known and environmental database, surprisingly gives pretty less hits than expected and a large number of ORFans. This made us curious and further we analyzed the ORFs predicted by the VIROME pipeline. We carried out a BLAST against the known UniRef and MgOl database followed by kmer analysis and analysis of other characteristics which might distinguish ORFans from non-ORFans. In the later part we moved on to the next component of the VIROME pipeline – ORF caller so get a clue on the reason behind ORFans. Hence, we reviewed the different ORF callers and ran them on our viral dataset to compare their efficiency. Here we discuss the process.

1.4.1 Problem Statement

My main goal is to determine if ORFans are truly unknown proteins or are they artifacts of the bioinformatics methodologies.

To address this goal we divide the problem into sub-aims:

- Aim 1 - Determine nature of ORFs, finding their homologs and factors affecting their classification as ORFans from Non-ORFans
- Aim 2 - Assess accuracy of ORF callers on viral metagenomes

1.4.2 Aims

My first aim is to find the nature of these ORFs. We analyze how many ORFs have found homologs to the known databases over time. Has the number of hits improved over the years? For this we carry out BLAST against the known UniRef database and MetagenomeOnline environmental database. Next we see whether ORF length or any other factors like ORF callers plays any role in classifying them as ORFans or non-ORFans. Finally we carry out k-mer analysis to determine difference sequence difference between ORFans and non-ORFans at the sequence level.

In Aim 2, I assess the accuracy of prediction by the ORF callers. In the VIROME pipeline, MGA is being used as the ORF caller and here we try to determine if there is any ORF caller that would perform more accurately. In the process, we looked into different ORF callers and compared their effectivity in terms of precision and sensitivity. We carried out analysis on both whole and shredded genome sequences for eleven viral samples using three ORF caller tools.

Chapter 2

METHODS

2.1 Aim 1 Dataset – VIROME

The dataset used for the first aim to carry out BLAST, kmer analysis, and analysis of other characteristics were downloaded from the VIROME database (http://virome.dbi.udel.edu/). Libraries downloaded for these analyses are presented in Table 2.1. The downloaded libraries have been sequenced using either of the two sequencing technologies – 454 Titanium or Sanger sequencing. CBAY2 library has been sequenced by both sequencing technologies. The libraries downloaded from VIROME were as follows: CBAY2 (Sanger), CBAY2 (454), GMF, STCS, and M601K.

Table 2.1 Libraries Downloaded From VIROME

Library	VIROME Prefix	Sample Location		# of ORFs
CBAY2 (Sanger)[+]	CFB	Chesapeake Bay858, MD	Sanger	33,639
CBAY2 (454)[+]	CFF	Chesapeake Bay858, MD	454 Titanium	288,457
GOM[+]	GMF	Gulf of Maine, ME	454 Titanium	111,322
STCS[*]	POF	Scripps Pier, CA	454 Titanium	560,833
M6O$_1$K[*]	POB	Line67 – 150km off Mass Pt, CA, 1000m depth	454 Titanium	144,198

[+] MV Williamson & Wommack [*] POV Hurwitz & Sullivan, 2013

2.2 Basic Local Alignment Searching Tool (BLAST) on VIROME libraries

BLAST was carried out on the VIROME libraries against two databases – UniRef100 and Metagenomes Online MgOl database. The UniRef 100 database files were downloaded from: ftp://ftp.uniprot.org/pub/databases/uniprot/previous_releases/ The UniRef100 files downloaded are for the following years and corresponding versions: 2005 – Release 1.0, 2006 – Release 7.0, 2008 – Release 13.0, 2010 – Release 2010_01, 2012 – Release 2012_03 and 2013 - Release 2013_09. The files for the corresponding databases for particular years were downloaded in a fasta format (*e.g.* input.pep.fa). The command used to create the database for carrying out BLAST was:

```
makeblastdb -in /path/to/input.pep.fa -dbtype prot
```

A BLASTP of each library was performed against each database using NCBI BLAST+ software (version 2.2.29). The command used was:

```
blastp -query /Path/to/library_pep.fasta -db /Path/to/input.pep.fa \
       -out /Path/to/output.btab -outfmt 6 -num_threads 8 -evalue 1e-3
```

The output obtained from the blastp command is a tab format file which was later processed to get the relevant values by using unix commands. These values were inserted in an excel sheet to make the corresponding graphs. The MgOl database was downloaded from: http://metagenomesonline.org/blast/downloads/MgOl-Feb2013/MgOl-All.vFeb2013.pep.fasta.gz

2.3 K-mer Analysis

The term ***k-mer*** typically refers to all the possible substrings, of length k, that are contained in a string. In Computational genomics, k-mers refer to all the possible subsequences (of length k) from a read obtained through DNA Sequencing. A Python script written by Ryan Moore was used to carry out k-mer (2 to 9 k-mers) analysis of sequences. Though k-mer analysis for k=2 to 9 was performed, results presented are only for 7-mers for analysis on our dataset of the VIROME libraries. The script calculates k-mer usage deviation (KUD) for each of the sequences that are analyzed. KUD is the actual number of a given k-mer divided by expected number of k-mers. For each k-mer for each sequence, a k-mer "fingerprint" is calculated.

2.4 Aim 2 Dataset For ORF Callers Comparison

A Chesapeake Bay shotgun metagenome was collected from surface water off the Smithsoanian Environmental Research Station pier on the Rhode River, Chesapeake Bay, MD and submitted to VIROME (prefix SRI). Protein taxonomic hits from VIROME were used to construct rank abundance curve. Examining the frequency of BLAST hits to known viral genomes in a viral shotgun metagenome collected from Chesapeake Bay hits to 10 viral genomes at various frequencies were chosen (Fig. 2.1) and the genomes were downloaded from NCBI. The following requirements were considered while selecting the viral genomes:

- Presence of a full length genome sequence in NCBI.
- Selecting viruses appropriate for the environment to be modeled.
- Preserving the GC content distribution of the original community.

Table 2.2 shows the rank of the different viral species. The rank abundance curve and table were generated by Ryan Moore and Steven Smith for their study on analysis of assembly metrics. Table 2.3 shows the size of each of the ten viral samples and the number of genes contained within them.

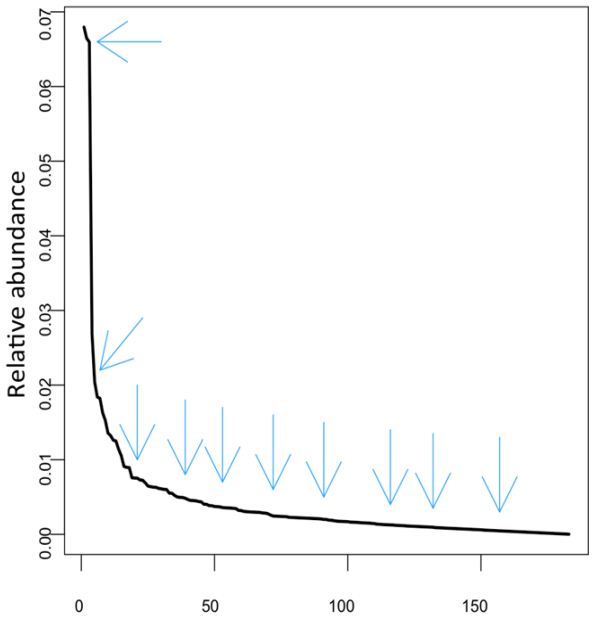


Figure 2.1 Rank abundance curve for representative viruses (arrows) chosen from the SERC (SRI) viral metagenome

Table 2.2 Rank Table For The Viral Samples

Virus Species	Rank	Hits	Abundance	GC
Punceispirillum Phage HMO-2011	1	3937	6.80%	43%
Prochlorococcus Phage P-SSM2	6	1066	1.81%	36%
Pelagibacter Phage HTVC011p	21	436	0.75%	32%
Cyanophage S-TIM5	39	280	0.48%	41%
Cellulophaga Phage phi14:2	53	208	0.36%	30%
Cyanophage KBS-P-1A	72	142	0.25%	47%
Sulfitobacter Phage pCB2047-C	91	116	0.20%	59%
Cyanophage NATL2A-133	116	73	0.13%	40%
Pseudomonas Phage tf	132	55	0.10%	53%
Klebsiella Phage JD001	157	27	0.05%	49%

Table 2.3 Viruses Size And Genes Contained

Virus	Genome Size	Genes
Enterobacteria Phage T4	168,903	278
Punceispirillum Phage HMO-2011	55,283	74
Prochlorococcus Phage P-SSM2	252,401	334
Pelagibacter Phage HTVC011p	39,922	45
Cyanophage S-TIM5	161,441	180
Cellulophaga Phage phi14:2	100,419	133
Cyanophage KBS-P-1A	45,731	57
Sulfitobacter Phage pCB2047-C	41,475	73
Cyanophage NATL2A-133	47,537	62
Pseudomonas Phage tf	46,272	72
Klebsiella Phage JD001	48,815	68

2.5 ORF Callers Used And Commands Used To Run The ORF Callers

The ORF callers – MetaGeneAnnotator (MGA) (Noguchi et al., 2008), MetaGeneMark (MGM) (Zhu et al., 2010) , Orphelia (Hoff et al., 2009a) and Glimmer-MG (Kelley, Liu, Delcher, Pop, & Salzberg, 2012a) were selected for the study. Initially, Glimmer-MG was used on some part of the dataset but later on due to changes in NCBI dataset, Glimmer-MG relies upon it could not be run on the remaining data, ultimately only the other three ORF callers were considered.

Following commands were used to run each ORF caller on the viral samples dataset to give the predicted ORFs:

- MetaGeneMark (MGM)

```
./gmhmmp -m MetaGeneMark_v1.mod -a -d -f G -r -o test.gff
input.fasta ./metagenemark2fasta.pl -i test.gff -- convert
.gff into fasta
```

  (Each of the arguments indicate the following:
  -m [filename] File with gene finding parameters
  -o [Ouptut] output file name
  -a Show protein sequence of predicted genes
  -d Show nucleotide sequence of predicted genes
  -f [L|G] Output format: [L] LST or [G] GFF2
  Default = L but we have used G
  -r use RSS for gene start prediction)

- MetaGeneAnnotator:

```
./mga_linux_ia64 input.fasta -m > mga_output perl mga2seq_pep.pl -i
input.fasta -m mga_output -p enterobacteria_phage_t4 -o
mga2pep_seq_output/
```

  (-m is given as parameter to indicate multiple species (sequences are individually treated))

- Orphelia

```
./orphelia -s genome_test.fasta -m Net700 -slots 1 -o .
```

  (Here Net700 is set as parameter to accommodate the metagenomes which use 454 and Illumina sequencing technology
  -s indicates seq-file
  -m model=Net700
  -slots indicated number of CPU to be used in parallel. Default is 1)

- Glimmer-MG

```
python glimmer-mg.py input.fasta
```

2.6 Whole Genome Analysis

For the whole genome analysis, annotations for the viral samples were downloaded from NCBI (.txt format). The output was converted to a common format by annotation parser (annotationParser.py) called annotation standard output format (ASOF). Throughout the analysis and for parsing Python (version 2.7) scripts have been used. Whole viral genome file was input to the ORF callers –MGM, MGA and Orphelia for ORF predictions. Each generated output in a different format, which was converted to a common format called standard tool output format (STOF) by respective tool parsers coded in Python (Fig. 2.2). The annotation output and tool output were then compared using a script called final parser script to generate the statistics. The entire process is depicted in Fig. 2.3.

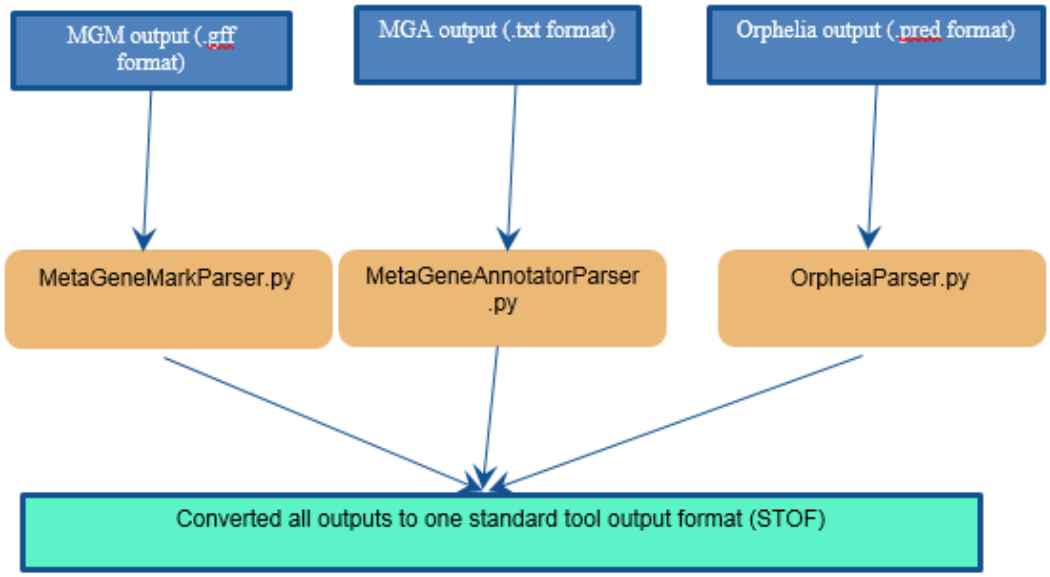


Figure 2.2 Flowchart Depicting the conversion of ORF callers outputs to a standard format (STOF). The figure shows the outputs from different ORF callers. MGM gives output in .gff format, MGA in .txt format and Orphelia in .pred format. Each of these outputs were processed through three separate python parser scripts to produce the output in a common format for further analysis.

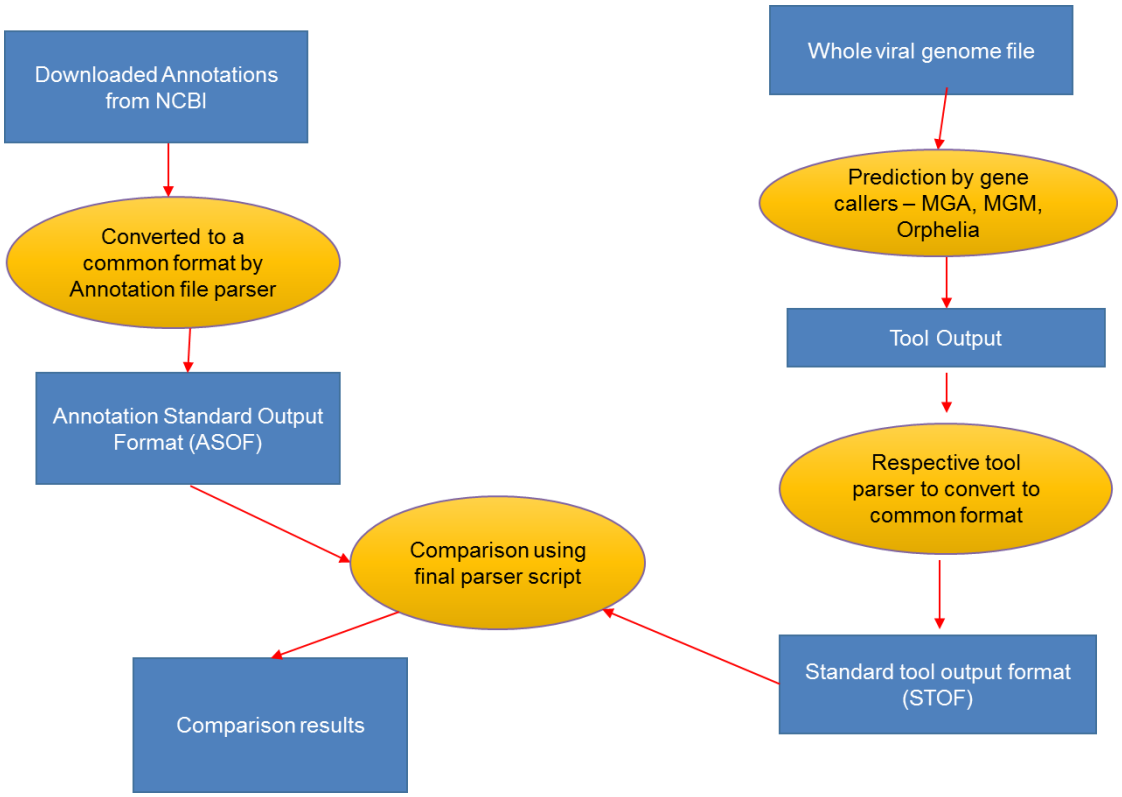


Figure 2.3 Flowchart Showing Analysis Process for whole genome ORF analysis. The figure shows the process for whole genome analysis, where in initially the annotations are downloaded from NCBI site. These are converted to a common format called Annotation Standard Output format (ASOF) by an Annotation file parser. The whole viral genomes are also input to the ORF callers, which make their respective predictions and give the output. As discussed in Fig. 3, these different tool outputs are converted to a common format called Standard tool output format (STOF) using tool parser script. Finally the annotation file and tool file outputs are compared using a final parser script to generate the results.

2.7 ORF Matching Cases For Annotation File and ORF Caller (Tools) Prediction Files

The annotation file and tool file (*i.e.* output from annotations and the ORF caller predictions respectively), give a file consisting of “Read name” and the ”ORFs”

predicted for each read. These ORFs are represented as a pair of numbers where the first number indicates the start co-ordinate while the second number indicates the end co-ordinate, indicating where an ORF is located on the read. Each read might have one or more ORFs predicted on it. We compare every ORF to every other ORF for a particular read. While comparing we consider different cases, for example "Exact Matched" where both the start and end coordinates from the expected annotation (ASOF) and tool prediction (STOF) match exactly. This is the simplest case, but there can be instances where one of the coordinates matches while the other doesn't. To consider all such combinations we divide the possibilities into a set of cases. Here we explain each of these cases pictorially (Fig. 2.4 (a) – (l)) along with their corresponding examples. Consider the two files –main reference file called Annotation file and the file to be compared with – ORF caller / tool file. Their co-ordinates are represented using A and T respectively. The numbers in brackets show the start and end co-ordinates (ORFs) for the Reads (i.e. 5' and 3' ends). Following cases are considered:

- Exact matched (Fig. 2.4 (a))

  This case consists of conditions where there is precise match
- Exact matched – Near matches (Fig. 2.4 (b) – (d))

  This includes conditions where one of either the 3' or 5' end differs by 1 or 2.

  Here the ORF predictions are out of frame.
- Partial matched (Fig. 2.4 (e) – (l))

  This includes conditions where either or both the 3' or 5' end is shorter or longer.

  Here the ORF predictions are in frame.
- False negatives - ORFs found in annotation but not predicted by tool

This case consists of ORFs of tool which do not satisfy any of the cases, so they are predicted exclusively by ORF callers.

- False positives - ORFs in tool but not in annotation

  This case consists of ORFs of annotation which do not satisfy any of the cases so they are predicted exclusively in annotation file.

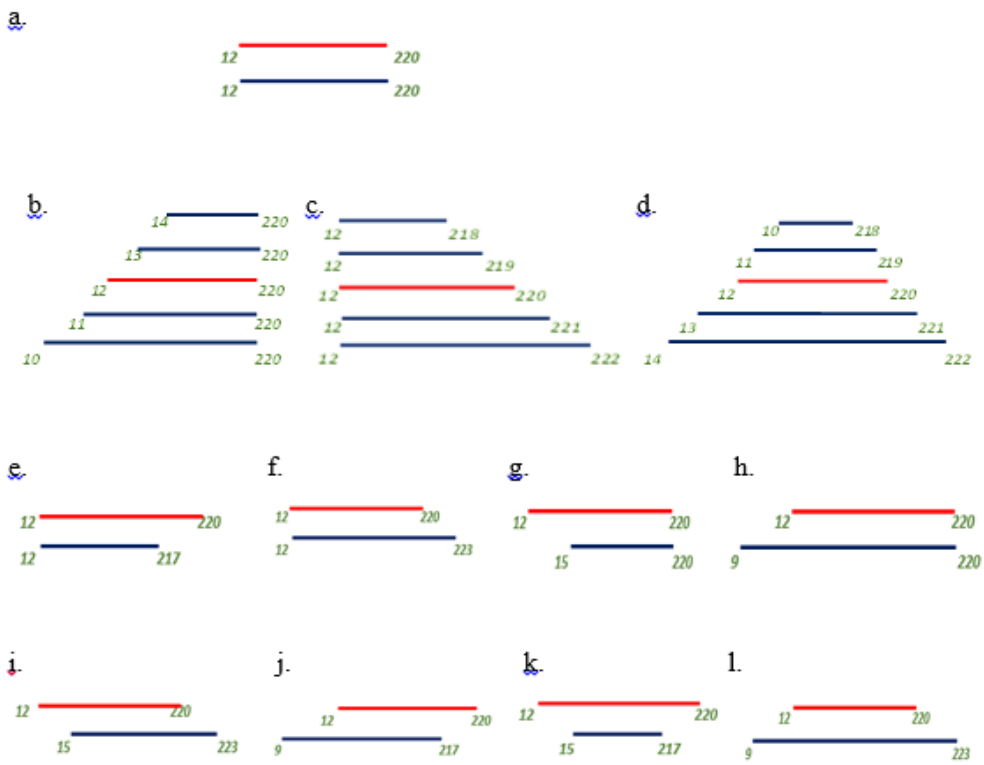


Figure 2.4 (a) Exact Matched. Expected (red) and predicted (blue) are identical. A – (12,220) T – (12,220). In this case, both the co-ordinates (5' end and 3' end exactly match

Figure 2.4 (b) Near Matches - Exact Matched - 3’ matched 5’ end off by 1 / 2. Consider A - – (12,220) and T – (10,220) (11,220) (13,220) (14,220). Here the 3’ end matches but the 5’ end differs either by 1 or 2 i.e. it can be more or less than 1 or 2. Thus, 3’ end of both annotation ORF and tool ORF matches while the 5’ end differs by either 1 or 2.

Figure 2.4 (c) Near Matches - Exact Matched - 5’ matched 3’ end off by 1 / 2. Consider – (12,220) and T – (12,219) (12,218) (12,221) (12,222). Here the 5’ end matches but the 3’ end differs either by 1 or 2 i.e.it can be more or less than 1 or 2.

Figure 2.4. (d) Near Matches – Exact matched - both 3’ and 5’ sides off by 1 / 2. Consider A - (12,220). If the ORF prediction is off by 1 then T-(11,219)(13,221). If the ORF prediction is off by 2 then T – (10,218)(14,222). Here both the 5’ end and 3’ end differs by either 1 or 2 i.e. it can be more or less by 1 or 2.

Figure 2.4 (e) Partial Matched - 5’ match 3’ short. Consider A – (12, 220) and T – (12, 217). Here 5’ end matches and 3’ end is short.

Figure 2.4 (f) Partial Matched - 5’ match 3’ long. Consider A – (12, 220) and T – (12, 223). Here 5’ end matches and 3’ end is long.

Figure 2.4.(g) Partial Matched - 3’ match 5’ short. Consider A – (12, 220) and T – (15,220). Here 3’end matches and 5’end is short.

Figure 2.4 (h) Partial Matched - 3’ match 5’ long. Consider A – (12,220) and T – (9,220). Here 3’end matches and 5’ end is long

Figure 2.4 (i) Partial Matched - 5’ short 3’ long. Consider A – (12,220) and T – (15 , 223). Here 5’ end is short and 3’ end is long.

Figure 2.4 (j) Partial Matched – 5’ long 3’ short. Consider A – (12,220) and T – (9,217). Here 5’ end is long and 3’ end is short.

Figure 2.4 (k) Partial Matched – 5’ short 3’ short. Consider A – (12,220) and T – (15,217). Here 5’end is short and 3’end is also short.

Figure 2.4 (l) Partial Matched – 5’ long 3’ long. Consider A – (12,220 and T- (9,223). Here 5’ is long and 3’ end is also long.

2.8 Shredded Genome Analysis For Viral Sample Datasets.

For shredded analysis, each whole genome (virus) was “shredded” using DWGsim (version 0.1.11) to create simulate a metagenomes. The cutoff for DWGsim was decided based on the sequencing technology being used. A fragment size of 300 was chosen as the Illumina platform was rapidly becoming the favored technology for metagenome sequencing at the time of the study. At that time 250bp was the maximum single end Illumina MiSeq read, 300bp was chosen as it represented a length that would likely be available in the near future. Command used for DWGsim:

```
dwgsim -1 300 -2 0 input.fasta fasta_file prefix virus_name
```

(-1 is supposed to be an INT value and indicates the length of the first read,

-2 is supposed to be an INT value and indicates the value of the second read)

The file obtained obtained from DWGsim was fasta file which was converted to fastq format by the following command:

```
cat in.fastq | perl -e

     '$i=0;while(<>){if(/^\@/&&$i==0){s/^\@/\>/;print;}elsif($i==1){p

     rint;$i=-3}$i++;}' > out.fasta
```

Simulated metagenomes were mapped to reference genome by CLC Bio and exported in sam format. A script called samParser.py was written to transfer expected ORFs from the genome annotation onto the fragmented reads. Specifically, this script maps the shredded sequences (metagenomes) to the whole genomes and gives information on the exact positions where they match. This file is called “Mapped metagenomes to reference annotation” file. The eleven simulated metagenomes were run separately through the four gene callers to find the corresponding ORF predictions by each ORF caller. Each ORF caller produced an output in different format which was converted to a common format by respective tool parser scripts (Fig.2.2). Subsequently, using a final parser script called finalParser.py, these predicted ORFs identified by ORF callers were compared to the expected ORFs in the “Mapped metagenomes to reference annotation” file. Fig. 2.5 illustrates the process for analysis of shredded genomes.

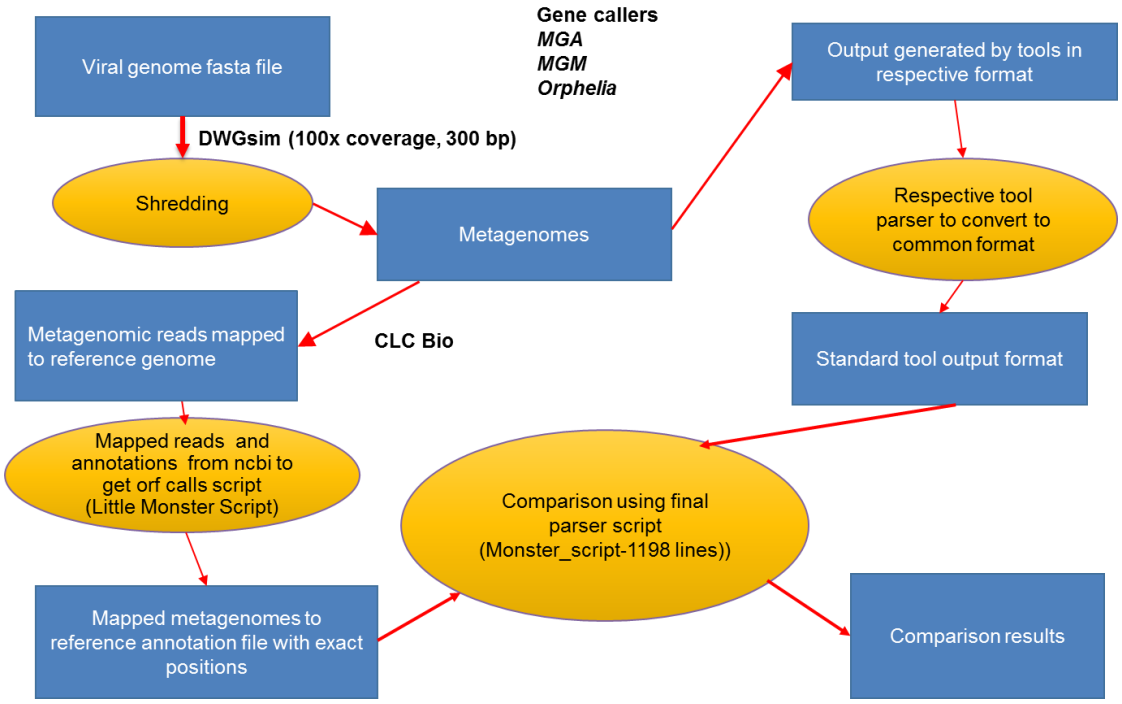


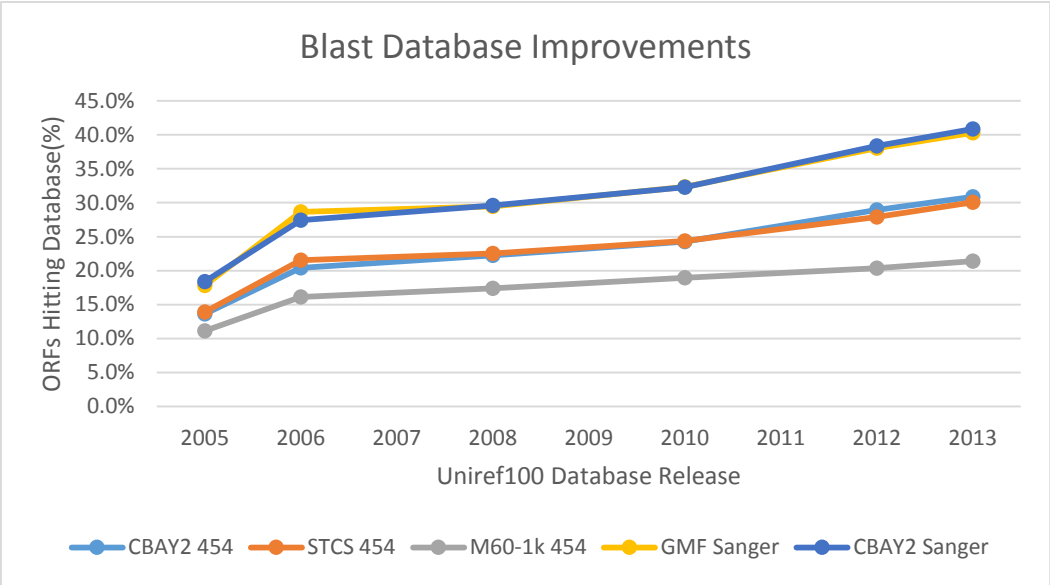
Figure 2.5 Shredded Genome Analysis flowchart. The figure shows the process for shredded genome analysis. The Fasta file for each of the genome libraries was given as input to DWGsim at100x coverage and 300 bp, which shredded these sequences to simulated metagenomes. The metagenomes were mapped to the whole reference genome using CLCBIO to get output file with mappings. This file gives information on to where in each whole genome each shredded read originated. A script transfers expected annotation from the genome to each shredded read. Subsequently, the simulated metagenomes from earlier step were given as inputs to ORF callers which made respective predictions. These outputs were converted to common format. The final parser script compares the outputs obtained from ORF callers file and annotation file to see how accurate the ORF callers predictions are.

Chapter 3

RESULTS

3.1 Monitoring ORFans Over Time

BLAST was carried out on the libraries (CBAY2 454, STCS 454, GMF Sanger, M601k 454, CBAY2 Sanger) downloaded from VIROME database to ascertain the effect of growing reference database on the number of ORFans. For BLAST against the reference database, UniRef 100 – one release was choose for each year 2005, 2006, 2008, 2010, 2012 and 2013. The percentage of ORFs with a significant BLAST hit against the databases steadily increases over the time period (Fig. 3.1 a). BLAST hit quality (median E-values) similarly increase over time (3.1 b). A BLAST was also carried out against the combined databases - UniRef100+-2011/2012 and MgOl-2013 environmental database, to see the increase in percentage of hits. The BLAST results (Fig. 3. 2 a, b) against the UniRef 100+ – 2012 and MgOl – 2013 database and against the UniRef 100+ – 2012 and MgOl – 2013 database are shown.



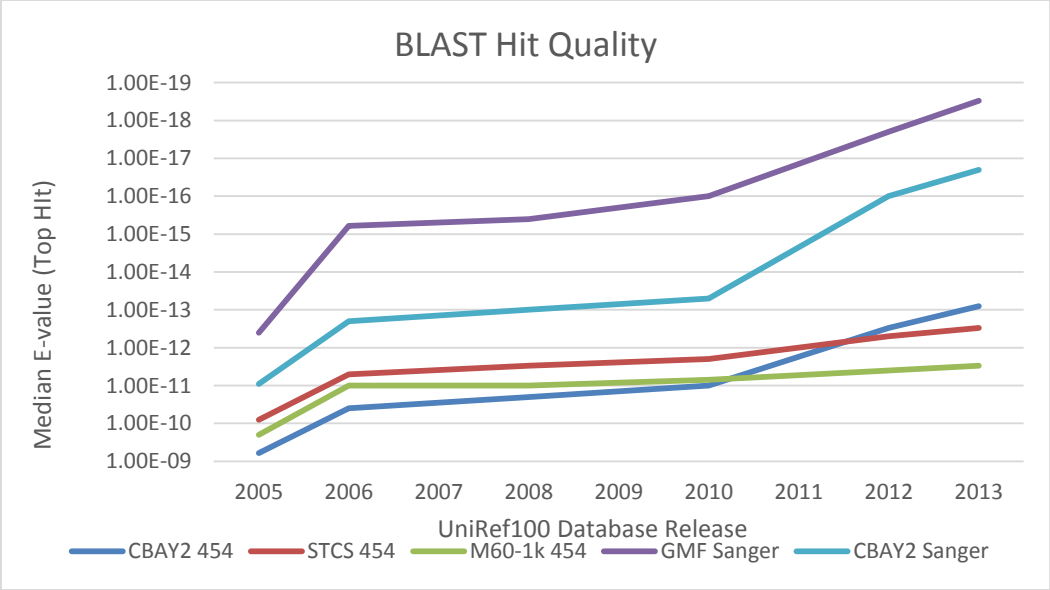
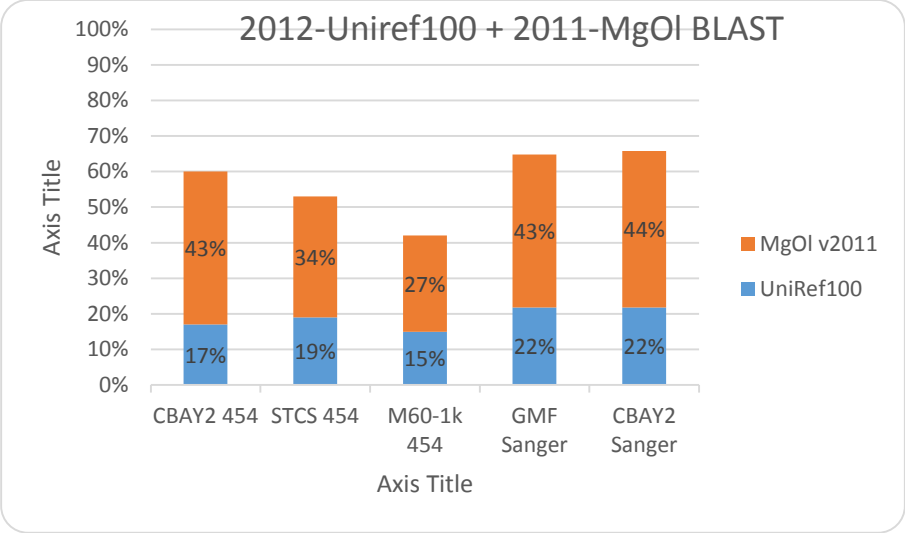


Figure 3.1 (a) BLAST database improvements from 2005-2013. This is a plot of BLAST database results for the five libraries against UniRef 100 from 2005-2013.

Figure 3.1 (b) BLAST hit quality for the five libraries against UniRef 100 from 2005-2013



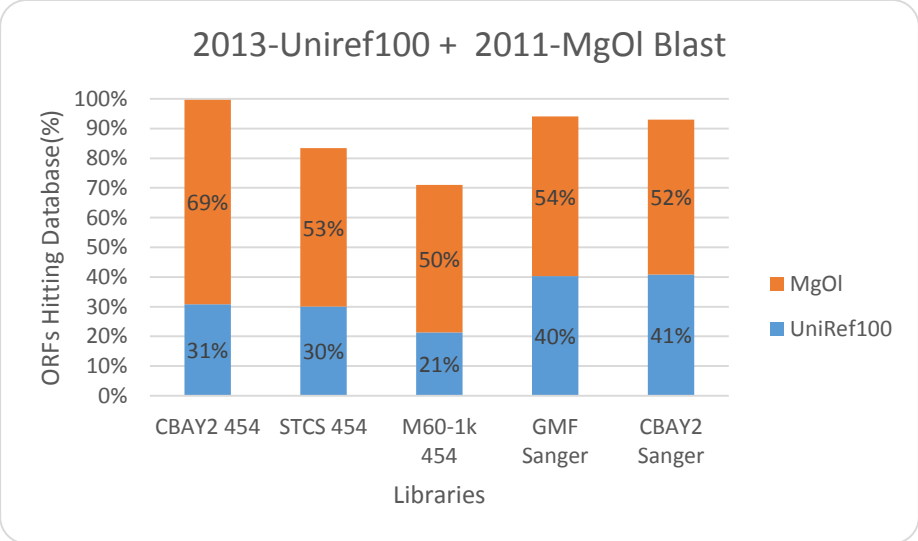
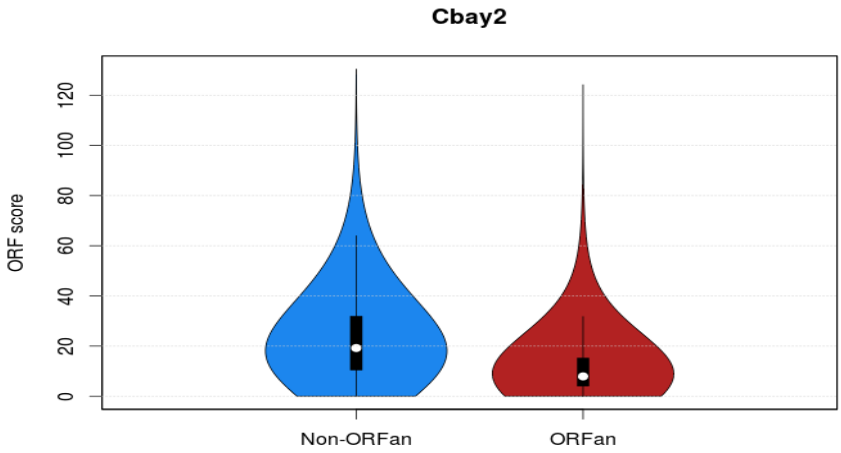


Figure 3.2 (a) BLAST results against UniRef 100+ (2012) and MgOl database (2011)

Figure 3.2 (b) BLAST results against UniRef 100+ (20113) and MgOl database (2011)

3.1.1 Characteristics Of ORFans Vs Non-ORFans

In the earlier step we carried out BLAST against the UniRef and MgOl database to determine if there have been any changes in the BLAST hit results and BLAST quality as the reference databases have grown. Our next step was to determine if there are any characteristics which might distinguish ORFans from non-ORFans. We took into consideration the ORF caller score and ORF lengths and plotted these values for for ORFans and non-ORFans. The ORF caller scores (Fig. 3.3 (a), (b), (c)) are the ones generated by the ORF caller MetaGeneAnnotator (MGA) and has been done internally by MGA's algorithm. It can be seen that ORFans have lesser ORF caller score than non-ORFans. As seen (from Fig. 3.3 (d), (e), (f)) also ORFans have considerably lesser length than non-ORFans. This could be as a result of lesser length ORFans going potentially undetected by the ORF callers. Looking ahead, we seek to find if there are sequence level differences between ORFans and non-ORFans. Hence we carry out k-mer analysis on them.



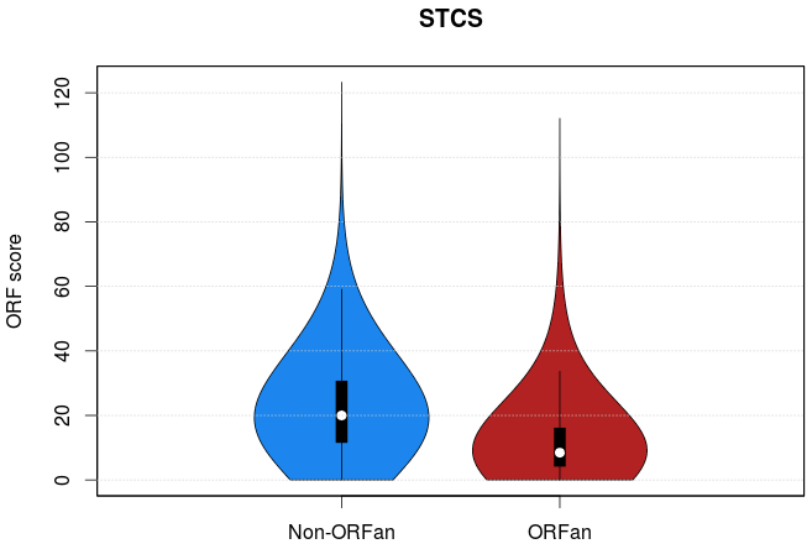
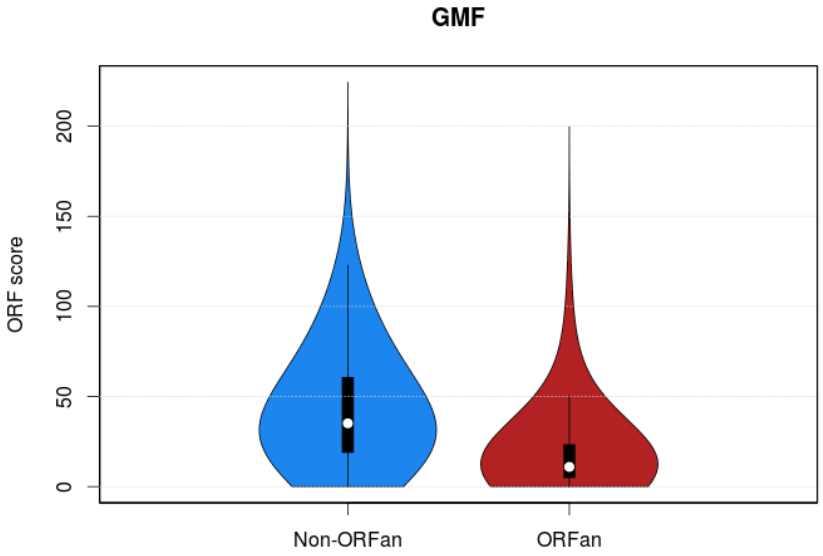


Figure 3.3 (a) ORF caller score plot for CBAY2 library

Figure 3.3 (b) ORF caller score plot for STCS library



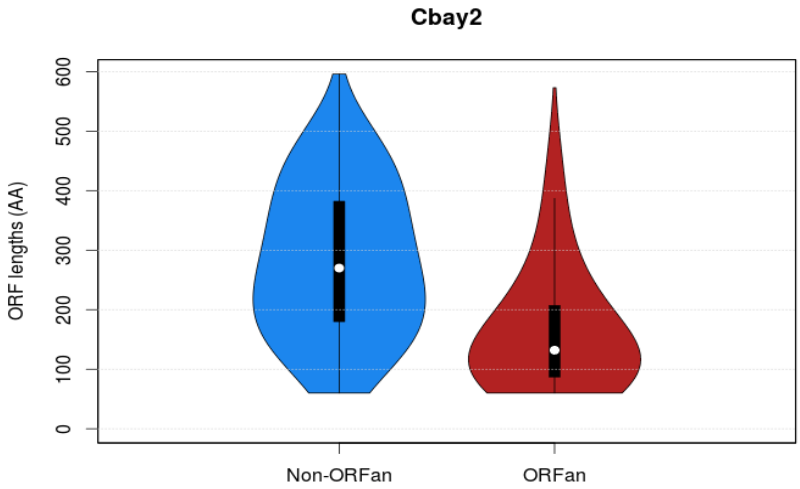
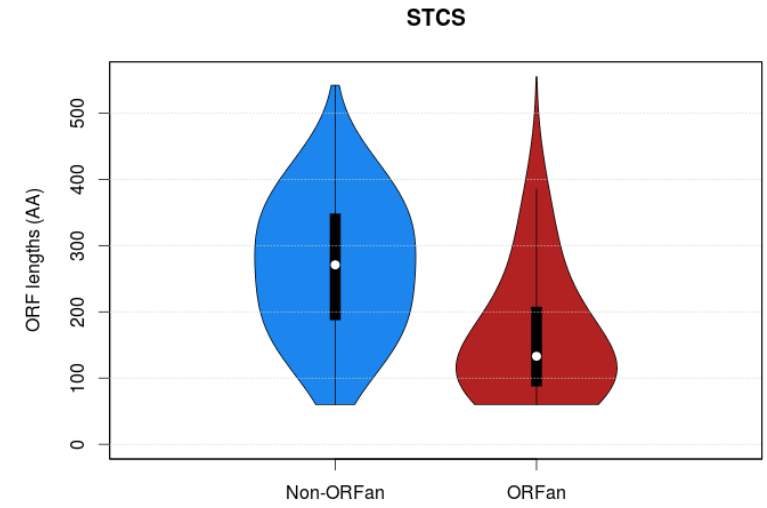


Figure 3.3 (c) ORF caller score plot for GMF library

Figure 3.3 (d) ORF length Plot for CBAY2 Library



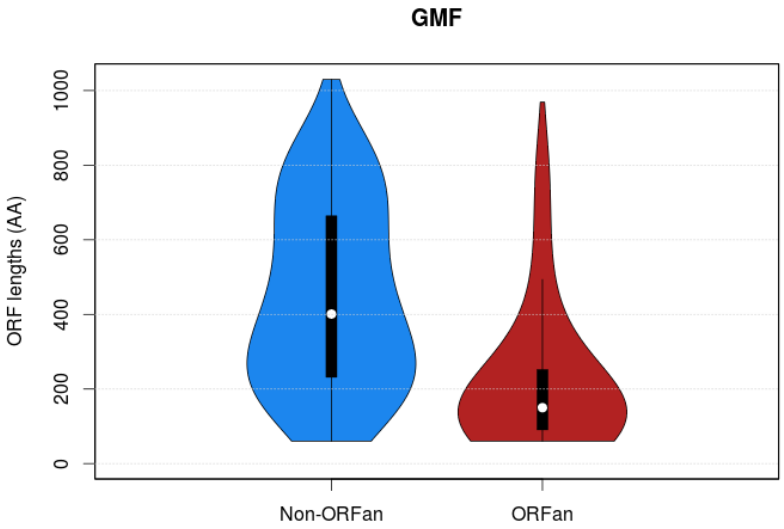


Figure 3.3 (e) ORF Length Plot for STCS library

Figure 3.3 (f) ORF Length Plot For STCS library

3.1.2 K-mer Analysis On ORFans & Non-ORFans

Earlier, we analyzed the BLAST database improvements and BLAST hit quality changes over the years, followed by an analysis of ORF caller score and ORF characteristics for ORFans and non-ORFans. The next kmer analysis was used to determine if sequence-level differences exist between ORFans and non-ORFans. For the k-mer analysis, the ratio of ORFans/non-ORFans Vs non-ORFans/ORFans was determined for each kmer and the results were sorted in decreasing order. The results (Table 3.1) shows that maximum range of the ratio of ORFans/non-ORFans is considerably greater than that of non-ORFans/ORFans. That is to say the most abundant kmers in ORFans are the most prevalent than the most abundant kmers in non-ORFans. This implies that there is an imbalance of kmers in non-ORFans. Many of the most over-represented kmers in ORFans represent homopolymeric sequences.

Table 3.1 Over-Represented K-mers Showing Ratio of Non-ORFans / ORFans & ORFans / non-Orfans

Non-ORFans			ORFans		
7-mer	**Non-ORFan/ ORFan**	**Rank**	**7-mer**	**ORFan/ Non-ORFan**	**Rank**
CGTAAAT	1.39	1	CCCCCCC	3.23	1
TAACTTT	1.39	2	GGGGGGG	2.89	2
CCTTTAC	1.39	3	GCCCCCC	2.35	3
GTTTAAC	1.37	4	CCCCCCG	2.28	4
TTTAGAT	1.37	5	TCCCCCC	2.22	5
TTTGATG	1.37	6	CCCCCTC	2.09	6
GGGTGAT	1.37	7	CCCCCCT	2.07	7
AACTTTA	1.37	8	CCCCCCA	1.99	8
TCAGGGC	1.36	9	CCCCCTT	1.95	9
GTTTGAT	1.36	10	GCCCCCT	1.94	10
AACTTTG	1.36	11	AGGGGGG	1.94	11
TGAGGGC	1.35	12	CTCCCCC	1.94	12
ACCTTTG	1.33	13	TTCCCCC	1.92	13
GTTTGGT	1.32	14	TTTTTTT	1.89	14
AGATTTA	1.32	15	GGGGCCC	1.88	15
AGCTTTA	1.32	16	CCCCCTA	1.87	16
ATTTAGT	1.32	17	CGCCCCC	1.85	17
GTTTAAG	1.32	18	ACCCCCC	1.82	18
ACTTTGA	1.32	19	CCCCCGC	1.79	19
ATTTAGA	1.31	20	GCCCCCG	1.79	20

The table shows the 7-mers for the libraries considered and their non-ORFans / ORFans VS ORFans/ non-ORFans ratios along with their ranks.

To determine if the prevalence of homopolymers may have been due to sequencing platform biases (earlier releases of 454 chemistry is known to produce homopolymeric artifacts) (Luo, Tsementzi, Kyrpides, Read, & Konstantinidis, 2012), frequency of kmers that were exclusively homopolymeric were examined. (Table 3.2). It is observed that homopolymers poly-C, G, and T homopolymers were all among the

100 most abundant kmers in 454 pyrosequencing libraries, but with Sanger, we can also see significant number of homopolymers, particularly poly-C.

Table 3.2 Over-Represented Homoploymers

	Library-Cbayvir2 Overrepresentation Rank		
Homopolymer	**454**	**Sanger**	**Hybrid Assembly**
CCCCCCC	1	1	71
GGGGGGG	2	2722	1200
CCCCCC	39	35	361
TTTTTTT	56	87	62
GGGGGG	94	1133	1218

The table shows the number of homopolymers for the 454 and Sanger sequencing technologies and also for a hybrid of these two technologies.

3.2 Aim 2 Results Overview

o determine whether ORFans are really unknown proteins or analytical artifacts, we decided to analyze the ORF callers. We know that in the VIROME pipeline, the ORF prediction is done my MGA. Are these predictions accurate or the ORFs have been wrongly predicted? We look into the other ORF callers and analyze from the results, if others might be able to do better than MGA. Here we discuss the different results obtained from analysis of the three different ORF callers –MGA, MGM and Orphelia on our dataset.

3.2.1 Whole Genome Results

Here we pictorially (Fig. 3.4, Fig. 3.5, Fig. 3.6; Additional charts in Appendix D) show the results of running ORF callers on the viruses and later on show the statistics (Table 3.3; Additional details in Appendix B) for each ORF caller for each of the virus. Here we can see that, amongst the ORF callers, MGA has the least number of false negatives (Found in tool but not in annotation) and false positives (Found in annotation but not in tool). MGA makes the maximum number of predictions and maximum exact and partial matches. Thus we can say that MetaGeneAnnotator (MGA) has the best precision and sensitivity amongst all other ORF callers, in case of whole genomes.

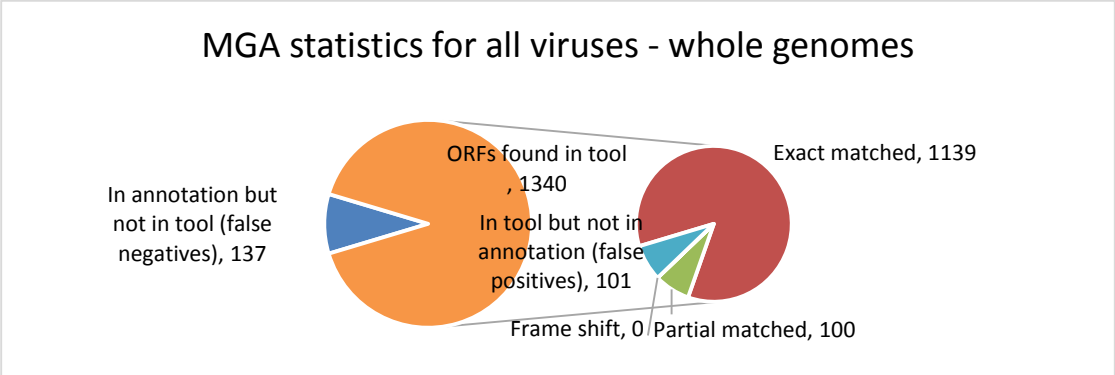


Figure 3.4 Pictorial representation of distribution of statistics for all viruses for whole genomes – MGA

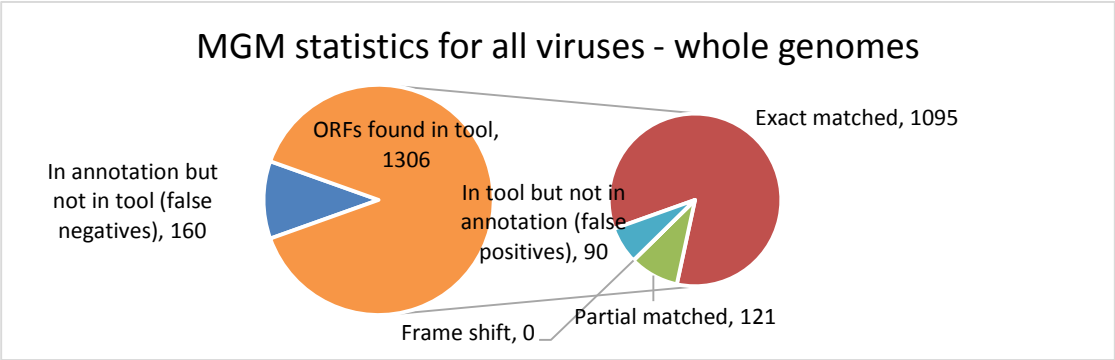


Figure 3.5 Pictorial representation of distribution of statistics for all viruses for whole genomes – MGM

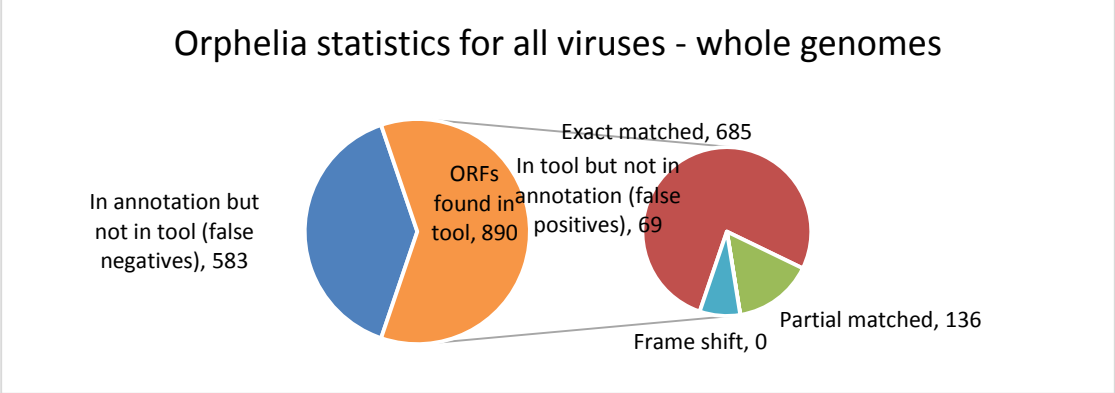


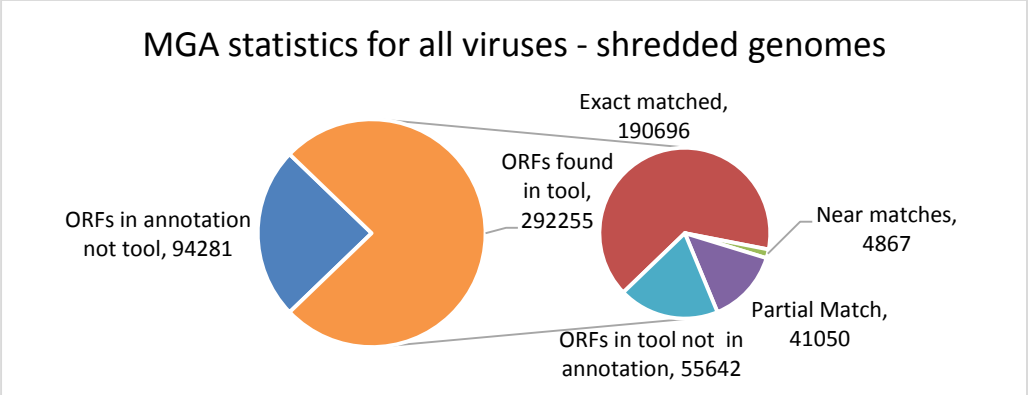
Figure 3.6 Pictorial representation of distribution of statistics for all viruses for whole genomes – Orphelia

Table 3.3 Statistics for the virus Enterobacteria Phage T4 for ORF callers –MGA, MGM, Orphelia and Glimmer-MG

Enterobacteria Phage T4 (168,903 bp)				
Type	**MGA**	**MGM**	**Orphelia**	**Glimmer-MG**
Total annotation count	278	278	278	278
Total tool count	264	263	244	247
Exact matched	240	225	206	219
Partial matched	14	28	31	22
Frame shift	0	0	0	0
Found in tool but not in annotation (false positives)	10	10	7	6
Found in annotation but not in tool (false negatives)	24	25	41	37

3.2.2 Shredded Genome Results

Here we pictorially show the results of running the three ORF callers (Fig. 3.7 a, b, Fig. 3.8 a, b, Fig. 3.9 a, b; Additional charts in Appendix E) and the shredded viruses and later on show the statistics (Table 3.4; Additional details in Appendix C) for each ORF caller for each of the virus. Here we can see that, amongst the ORF callers, MGA has the least number of false negatives (Found in tool but not in annotation) and false positives (Found in annotation but not in tool). MGA makes the maximum number of predictions and maximum exact and partial matches. Seeing these results, it is clear that MetaGeneAnnotator (MGA) has the best precision and sensitivity amongst all other ORF callers, in case of shredded genomes.



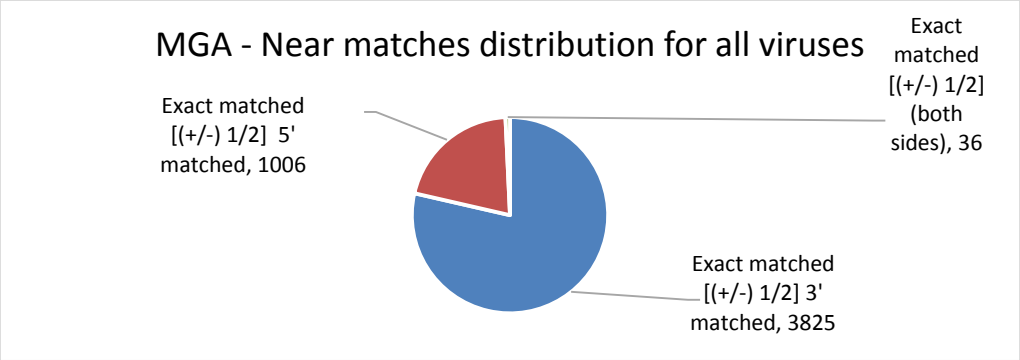


Figure 3.7 (a) Pictorial representation of distribution of statistic for all viruses – MGA
(b) Pictorial representation of distribution of Near matches statistic for all viruses – MGA

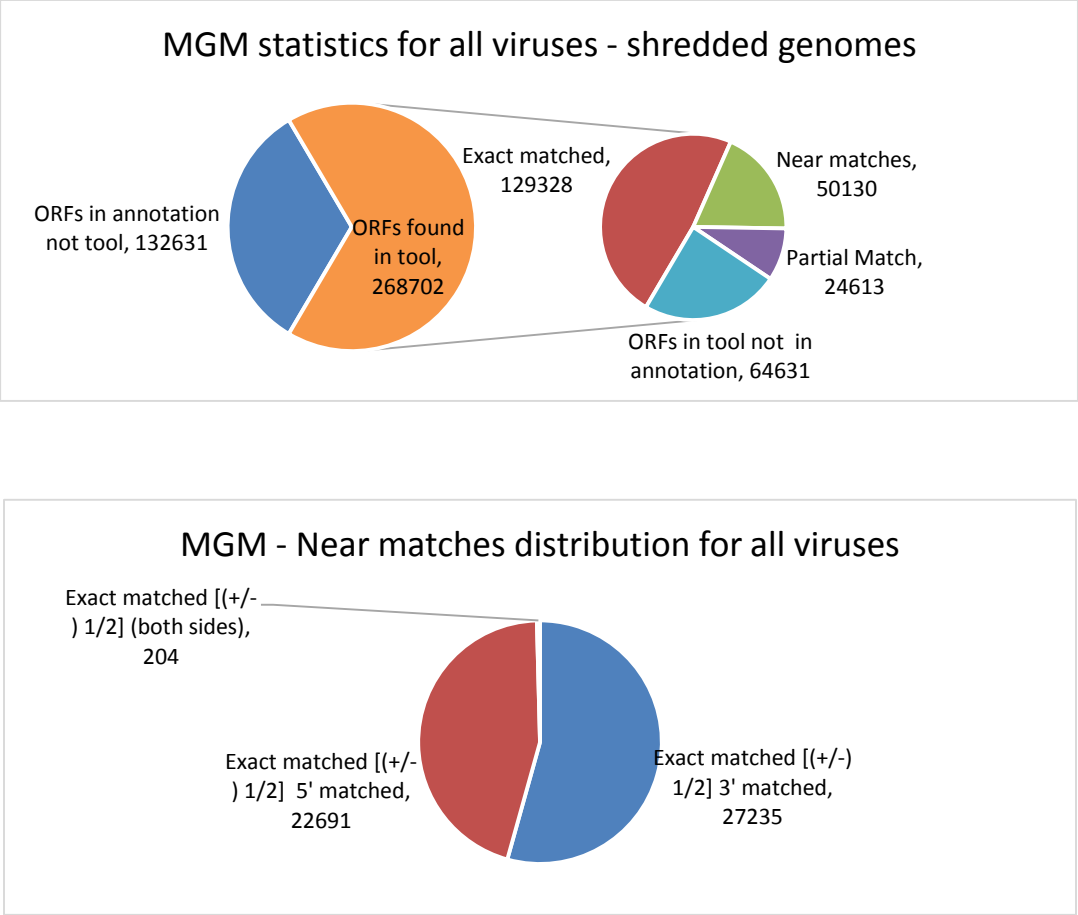
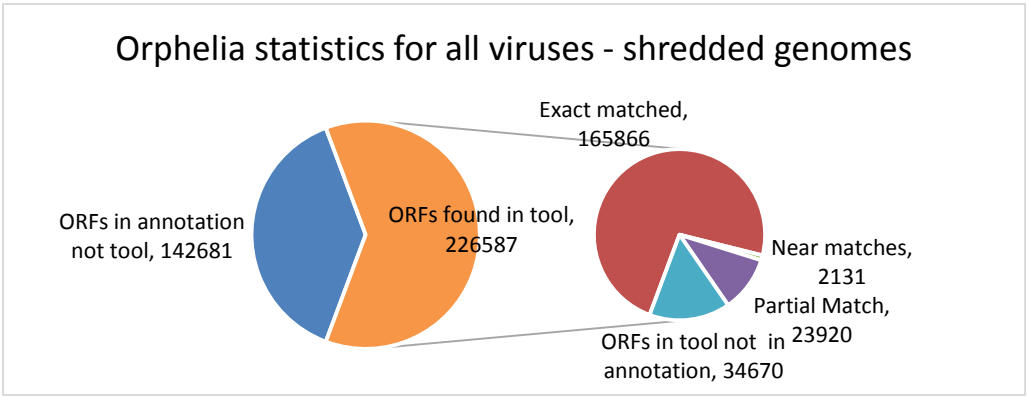


Figure 3.8 (a) Pictorial representation of distribution of statistic for all viruses – MGM
(b) Pictorial representation of distribution of Near matches statistic for all viruses –MGM



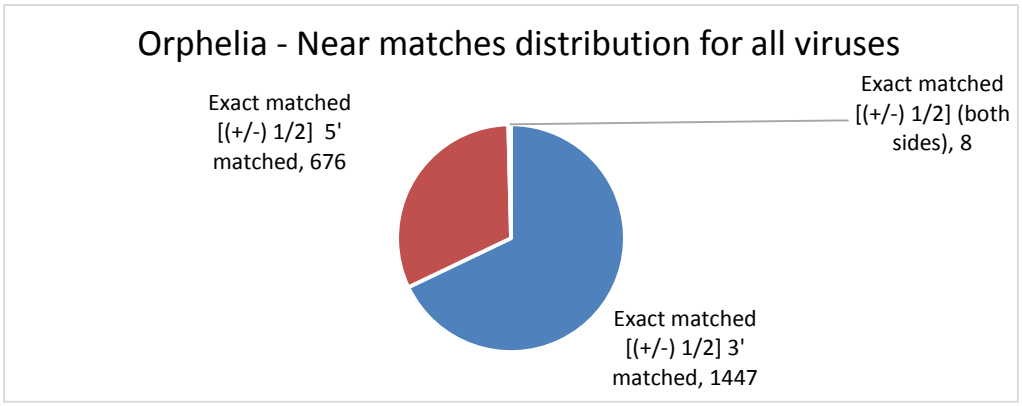


Figure 3.9 (a) Pictorial representation of distribution of statistic for all viruses – Orphelia **(b)** Pictorial representation of distribution of Near matches statistic for all viruses – Orphelia

Table 3.4 Statistics For The Enterobacteria Phage T4 For ORF Callers – MGA, MGM And Orphelia

Enterobacteria Phage T4 (168,903 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	79,178	79,178	79,178
Total tool count	72,239	64,698	51,276
Exact matched	46,278	28,546	19,594
Exact Matched - 3'matched 5' end off by 1 / 2	838	8,175	124
Exact Matched - 5' matched 3' end off by 1 / 2	364	7,148	147
Exact matched – both 3' and 5' end off by 1/2	12	76	1
Partial Match	9508	6039	2626
ORFs in tool not in annotation (false positives)	11038	14315	28625
ORFs in annotation not tool (false negatives)	21,070	29,544	56,570

3.3 Analyzing MetaGeneAnnotator(MGA's) Results

From the above statistics it is evident that MetaGeneAnnotator (MGA) has the better prediction ability (maximum sensitivity) amongst all other ORF callers. MetaGeneMark (MGM) has maximum precision. So we concentrate on MGA's results

and seek to find out how we can better the results. Initially, to check if there were particular length or score ranges in which ORFans and non-ORFans fall we made a few plots. Scatter plots of ORF caller score and length (Fig. 3.10 (a), (b)) were made. A distribution of ORF caller scores (Fig. 3.11 (a), (b)) and ORF lengths (Fig. 3.12 (a), (b) for ORFans and non-ORFans is shown.

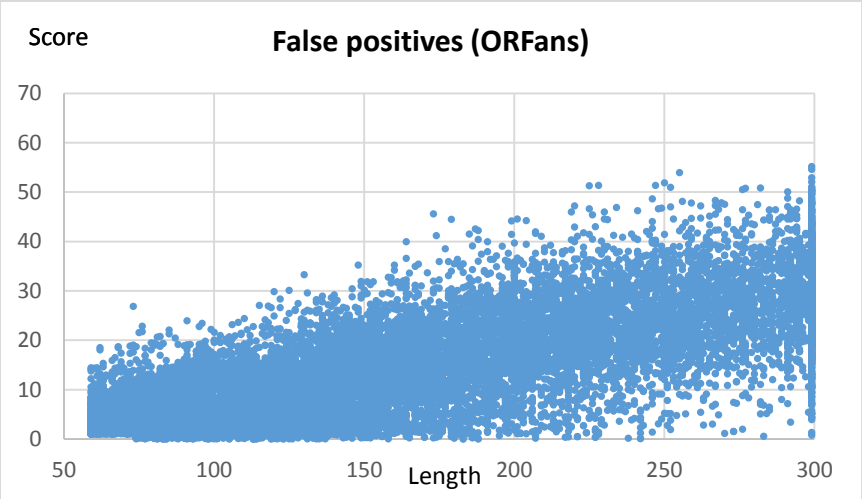


Figure 3.10 (a) ORF caller scores for ORFans of MGA for Enterobacteria Phage T4

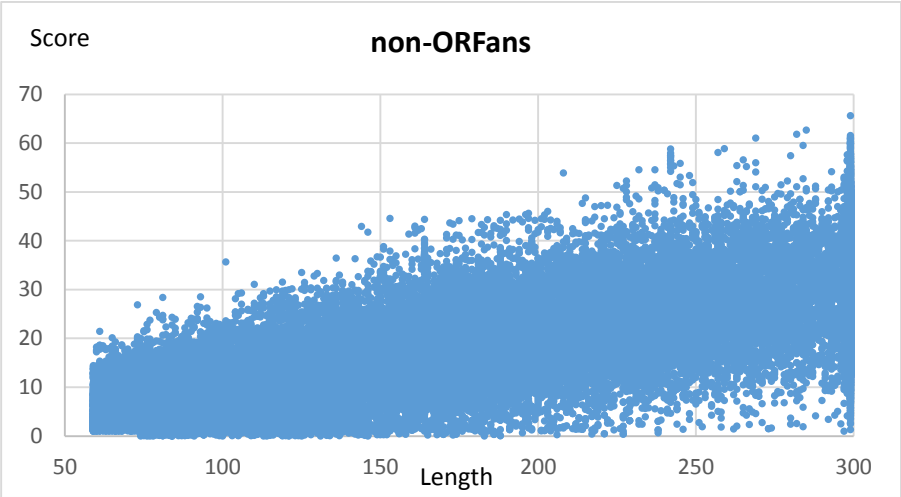


Figure 3.10(b) ORF caller scores for non-ORFans of MGA for Enterobacteria Phage T4

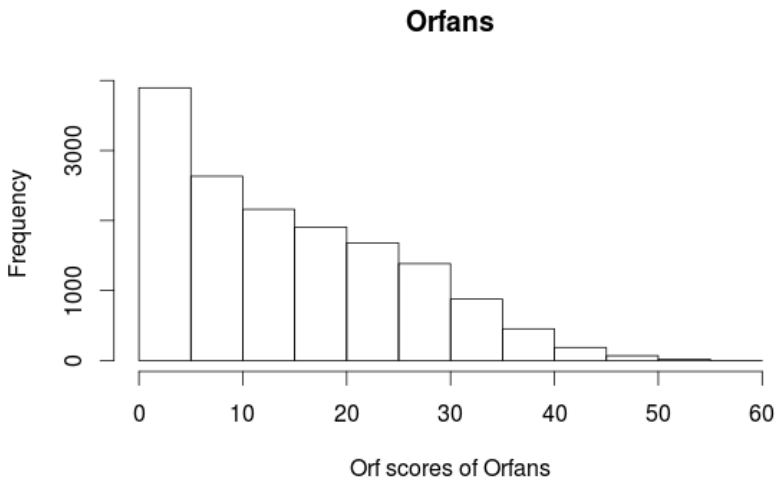


Figure 3.11(a) ORF caller score distribution for ORFans of MGA for Enterobacteria Phage T4

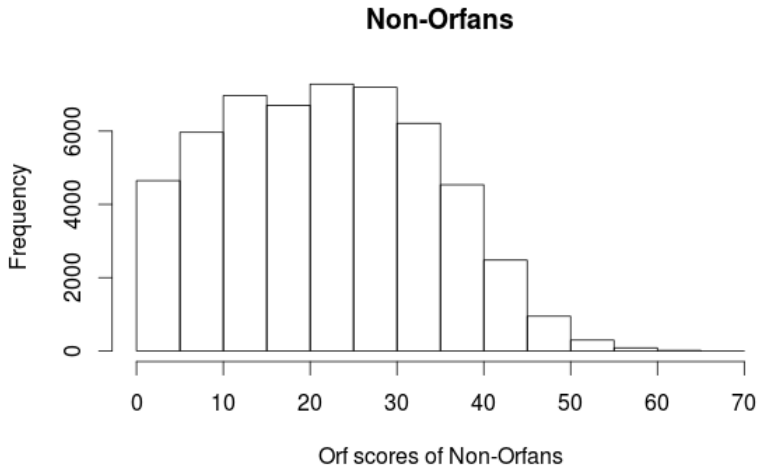


Figure 3.11(b) ORF caller score distribution for non-ORFans of MGA for Enterobacteria Phage T4

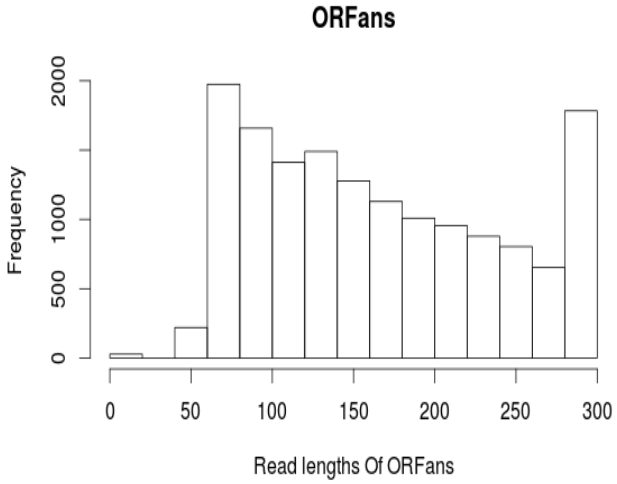


Figure 3.12(a) ORF lengths distribution for ORFans of MGA for Enterobacteria Phage T4

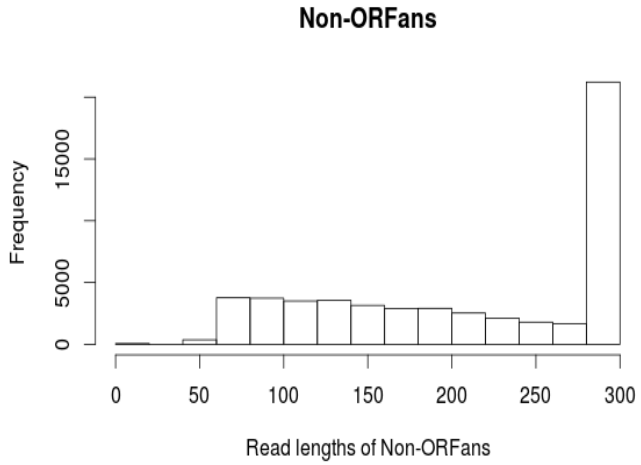


Figure 3.12(b) ORF lengths disctribution for non-ORFans of MGA for Enterobacteria Phage T4

This analysis of data wasn't very conclusive so as to be able to give a definitive range of length and scores in which ORFans or non-ORFans are likely to fall. So our next step was to generate results for different cutoffs for length and ORF caller scores. Results for ORF length cutoffs for values 60, 66, 72, 78, 84 and 102 were generated. In the Fig. 3.12 (a), we graphically show these results consisting of Exact matched, False positives, False negatives and False negatives lesser than ORF length cutoff. Similarly the results for different ORF score cutoffs (Fig. 3.13 (b)) have been shown for the values 0.5, 1.0, 1.5, 2.0, 5.0, 10.0. These cutoffs implies that only reads above the decided cutoffs are considered while generating statistics. From these graphs, we can see that the false negatives steadily increase while the false positives steadily decrease, as the cutoff is increased. The rate at which they change is different, false positives decrease at a slower rate as compared to both false negatives and false

negatives lesser than the cutoff length of 60. This can be seen both in case where the ORF caller score and ORF length cutoff is being increased.

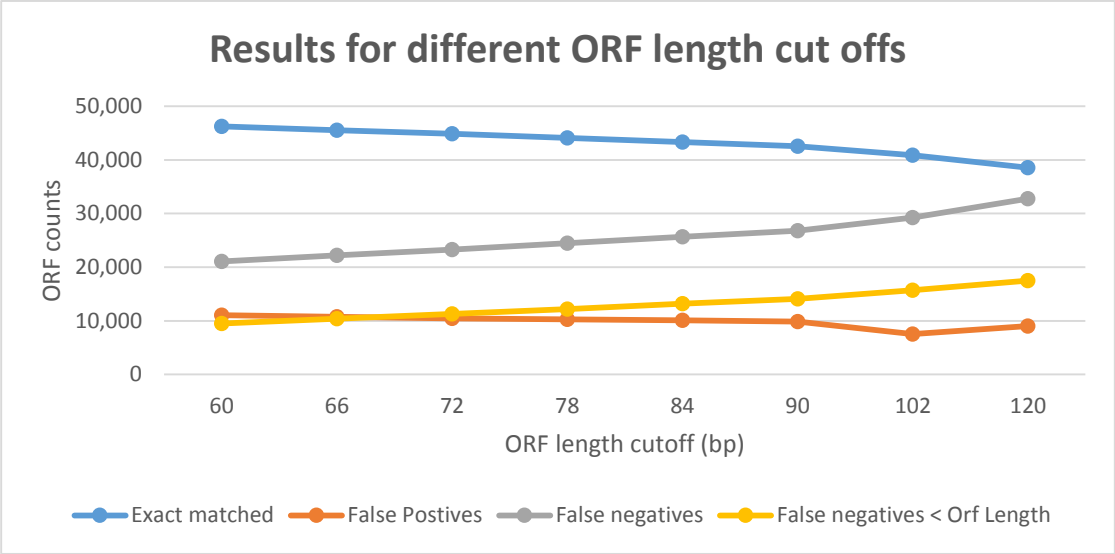


Figure 3.13(a) Results for different ORF length cutoffs. In each case, ORF lengths above the specified cutoff values - 60, 66, 72, 78, 102 were considered. The exact matched, false positives, false negatives and false negatives less than ORF length cutoff count are shown.

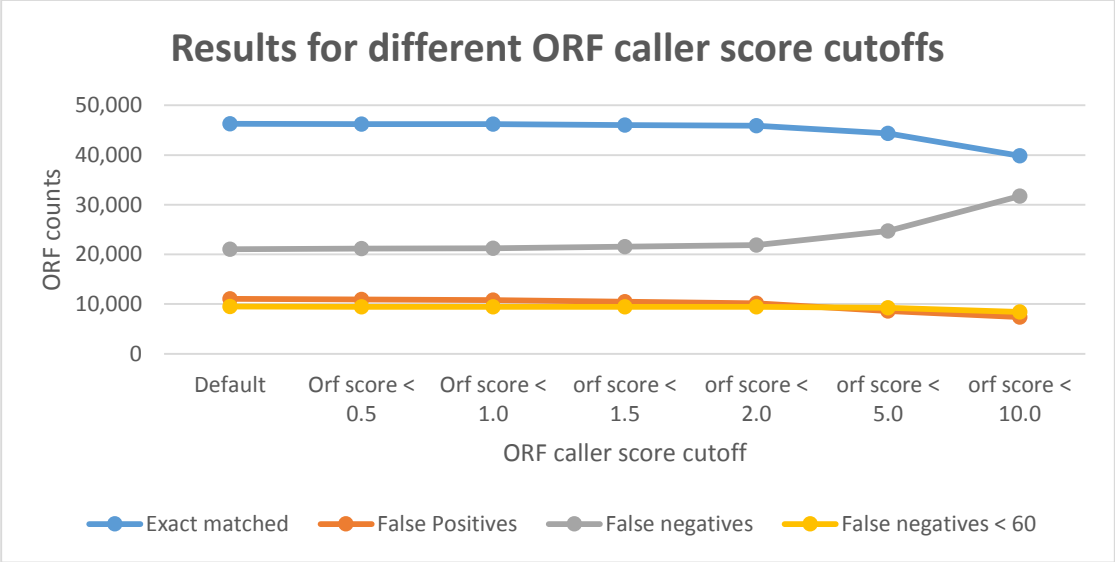


Figure 3.13 (b) Results for different ORF caller score cutoffs. In each case, ORF scores above the specified cutoff values – 0.5, 1.0, 1.5, 2.0, 5.0, 10.0 were considered. The exact matched, false positives, false negatives and false negatives less than ORF length cutoff counts are shown.

Chapter 4

DISCUSSION AND CONCLUSION

We considered a set of predicted ORFs by the ORF caller of the VIROME pipeline, which belonged to five environmental viral libraries. Initially a BLAST was carried out against the reference and environmental database to check for any changes or improvements over the years for these libraries. The BLAST database and hit quality results indicated that the number of ORFs hitting the database have steadily increased over time. This is due to the fact that more and more proteins have been accounted for in the database hence more ORFs find homologs. Even though this was true, we could see that even after doing a BLAST against the metagenomic database, a huge number of ORFans remain. The question to be answered was "are the ORFans really unknown proteins or are they methodological artifacts?" Going further we analyzed if there can be any characteristics found, which distinguishes ORFans and non-ORFans. In general, the ORF caller score and length of ORFans are lower than non- ORFans. ORF callers tend to miss ORFs below certain lengths (false negatives) and at the same time predict ORFs that are not real (false positives). They are not suited to detect such short, fragmentary ORFs as these ORFs provide little evidence to its validity or lack thereof. Thus the OR callers behavior becomes more inconsistent contributing to both false negatives and false positives in this size range. This tends to indicated that setting a minimum ORF caller score and/or minimum ORF length would seemingly reduce the number of ORFans, by reducing the number of artifactual

ORFs (false positives) identified, this does come at the expense of missing almost as many actual ORFs below that cutoff.

Further, we checked if any sequence level differences between ORFans and non-ORFans. It can be seen that maximum range of ORFans/non-ORFans ratio is considerably greater than the maximum ratio of non-ORFans/ORFans (Table 3.1). A large number of homopolymers can be seen in ORFans. Most of these libraries are sequenced using 454 which is known to produce extra homoploymers, there are indeed more overrepresented homopolymers in 454 vs. Sanger libraries (Table 3.2) so this maybe a contributing factor. However, the fact that there are all still significant numbers of overrepresented kmers in Sanger (in fact the most common kmer in the Sanger libraries is also a homopolymer) may indicate an artifact of the ORF caller.

We next analyzed the prediction efficiency of the ORF callers itself. Are the ORFs being wrongly predicted by the ORF caller tools? Three were selected to be run on our dataset of eleven simulated viral metagenomes, MetaGeneAnnotator (MGA), MetaGeneMark (MGM) and Orphelia. The viral metagenomes consisted of both whole and shredded genomes. From the results, we can say that MGA has the highest sensitivity (0.74) and precision (0.82). Next, we concentrate on MGA's results and try to find if there is any particular ORF length range or ORF caller score range that ORFans and non-ORFans are most likely to fall into. We generated results for different cutoffs of ORF length and ORF caller score and looked at how these cutoffs affected the number of false negatives and false positives. The precision can be seen increasing (false positives decreases) but the sensitivity decreases (false negatives

increases) as the cutoffs are increased. Here the question arises that whether fixing a particular cutoff for score and length is able to decrease the number of ORFans or no, or there are other reasons contributing to ORFans. Event though, the precision comparatively can be seen to increase more rapidly than sensitivity, the changes for both over different cutoffs is not steep, rather is gradual. The false positives rate of decrease can be seen to be faster than increase in rate of false negatives. A sharp decline in the false positives can be seen when ORF caller score cutoff is 2 (14.5%) as compared to a score of 5 (13.2%). Thus, a cutoff of score 2 might be a good possibility. In case of ORF lengths, it can be seen that the false positives are decreasing slowly up till 84 (15.38%), while they decline steeply at 102 (12.39%). Also, length cutoff looks like a good option for lowering ORFans as compared to score, as it helps in eliminating shorter false negatives and increases longer false negatives. A next step for this analysis should be modelling combinations of ORF caller score and ORF lengths. A particular combinatorial cutoff approaches of ORF caller score and ORF lengths, and potentially also reducing filters for ORFs containing long homopolymers. A particular combination might be the most suitable giving the least number of ORFans. Another good analysis would be look for to viral proteins from GeneBank and determine how many fall below the different length cutoffs. From our study, it can be seen that the false positives are the major contributors to ORFans. Thus, we can conclude that ORFans are most likely artifacts of the sequencing platform and ORF callers. The false-positives can be managed to a degree by applying the right length and/or score cutoffs (e.g. > 90 bp and score > 2), but it must be kept in mind that sensitivity is sacrificed. And the balance of these two factors must be considered in applying these cutoffs.

REFERENCES

Allen, J. E., Majoros, W. H., Pertea, M., & Salzberg, S. L. (2006). JIGSAW, GeneZilla, and GlimmerHMM: puzzling out the features of human genes in the ENCODE regions. *Genome Biology*, *7 Suppl 1*(Suppl 1), S9.1–13. doi:10.1186/gb-2006-7-s1-s9

Bergh, O., Børsheim, K. Y., Bratbak, G., & Heldal, M. (1989). High abundance of viruses found in aquatic environments. *Nature*, *340*(6233), 467–468.

Besemer, J., & Borodovsky, M. (1999). Heuristic approach to deriving models for gene finding. *Nucleic Acids Research*, *27*(19), 3911–3920.

Bhasvar, J., Polson, S., Dhankar, S., & Wommack, E. (2009). VIROME: a resource for analysis of viral metagenomes. *General Meeting of the American …*. Retrieved from http://scholar.google.com/scholar?cluster=6749875198543088646&hl=en&oi=scholarr#0

Borodovsky, M., & McIninch, J. (1992). GeneMark_parallel_gene___Article.pdf.

Brussaard, C. P. D., Wilhelm, S. W., Thingstad, F., Weinbauer, M. G., Bratbak, G., Heldal, M., … Wommack, K. E. (2008). Global-scale processes with a nanoscale drive: the role of marine viruses. *The ISME Journal*, *2*(6), 575–578.

Caranta, C., Aranda, M., Tepfer, M., & Lopez-Moya, J. (Eds.). (2011). *Recent Advances in Plant Virology*. Caister Academic Press.

Delcher, A. (1999). Improved microbial gene identification with GLIMMER. *Nucleic Acids Research*, *27*(23), 4636–4641. doi:10.1093/nar/27.23.4636

Delcher, A. L., Bratke, K. A., Powers, E. C., & Salzberg, S. L. (2007). Identifying bacterial genes and endosymbiont DNA with Glimmer. *Bioinformatics (Oxford, England)*, *23*(6), 673–9. doi:10.1093/bioinformatics/btm009

Edwards, R., & Rowher, F. (2005). Viral metagenomics, *3*(June), 801–805.

El Allali, A., & Rose, J. R. (2013). MGC: a metagenomic gene caller. *BMC Bioinformatics*, *14 Suppl 9*(Suppl 9), S6. doi:10.1186/1471-2105-14-S9-S6

Goris, N., Vandenbussche, F., & De Clercq, K. (2008). Potential of antiviral therapy and prophylaxis for controlling RNA viral infections of livestock. *Antiviral Research*, *78*(1), 170–8. doi:10.1016/j.antiviral.2007.10.003

Handelsman, J. (2004). Metagenomics: application of genomics to uncultured microorganisms. *Microbiology and Molecular Biology Reviews : MMBR*, *68*(4), 669–85. doi:10.1128/MMBR.68.4.669-685.2004

Hoff, K. J., Lingner, T., Meinicke, P., & Tech, M. (2009a). Orphelia: Predicting genes in metagenomic sequencing reads. *Nucleic Acids Research*, *37*(SUPPL. 2).

Hoff, K. J., Lingner, T., Meinicke, P., & Tech, M. (2009b). Orphelia: predicting genes in metagenomic sequencing reads. *Nucleic Acids Research*, *37*(Web Server issue), W101–5. doi:10.1093/nar/gkp327

Hyman, P., & Abedon, S. T. (2012). Smaller Fleas: Viruses of Microorganisms. *Scientifica*.

Kelley, D. R., Liu, B., Delcher, A. L., Pop, M., & Salzberg, S. L. (2012a). Gene prediction with Glimmer for metagenomic sequences augmented by classification and clustering. *Nucleic Acids Research*, *40*(1).

Kelley, D. R., Liu, B., Delcher, A. L., Pop, M., & Salzberg, S. L. (2012b). Gene prediction with Glimmer for metagenomic sequences augmented by classification and clustering. *Nucleic Acids Research*, *40*(1), e9. doi:10.1093/nar/gkr1067

Koonin, E. V, Senkevich, T. G., & Dolja, V. V. (2006). The ancient Virus World and evolution of cells. *Biology Direct*, *1*, 29. doi:10.1186/1745-6150-1-29

Lawrence, C. M., Menon, S., Eilers, B. J., Bothner, B., Khayat, R., Douglas, T., & Young, M. J. (2009). Structural and functional studies of archaeal viruses. *Journal of Biological Chemistry*.

Luo, C., Tsementzi, D., Kyrpides, N., Read, T., & Konstantinidis, K. T. (2012). Direct comparisons of Illumina vs. Roche 454 sequencing technologies on the same microbial community DNA sample. *PloS One*, *7*(2), e30087. doi:10.1371/journal.pone.0030087

Makarova, K. S., Grishin, N. V, Shabalina, S. A., Wolf, Y. I., & Koonin, E. V. (2006). A putative RNA-interference-based immune system in prokaryotes:

computational analysis of the predicted enzymatic machinery, functional analogies with eukaryotic RNAi, and hypothetical mechanisms of action. *Biology Direct*, *1*, 7.

Marine, R., McCarren, C., Vorrasane, V., Nasko, D., Crowgey, E., Polson, S. W., & Wommack, K. E. (2014). Caught in the middle with multiple displacement amplification: the myth of pooling for avoiding multiple displacement amplification bias in a metagenome. *Microbiome*, *2*(1), 3. doi:10.1186/2049-2618-2-3

Marine, R., Polson, S. W., Ravel, J., Hatfull, G., Russell, D., Sullivan, M., … Wommack, K. E. (2011). Evaluation of a transposase protocol for rapid generation of shotgun high-throughput sequencing libraries from nanogram quantities of DNA. *Applied and Environmental Microbiology*, *77*(22), 8071–9. doi:10.1128/AEM.05610-11

Mojica, F. J. M., Díez-Villaseñor, C., García-Martínez, J., & Soria, E. (2005). Intervening sequences of regularly spaced prokaryotic repeats derive from foreign genetic elements. *Journal of Molecular Evolution*, *60*(2), 174–182.

Nelson, K. E., Weinstock, G. M., Highlander, S. K., Worley, K. C., Creasy, H. H., Wortman, J. R., … Zhu, D. (2010). A catalog of reference genomes from the human microbiome. *Science (New York, N.Y.)*, *328*(5981), 994–999.

Noguchi, H., Park, J., & Takagi, T. (2006). MetaGene: prokaryotic gene finding from environmental genome shotgun sequences. *Nucleic Acids Research*, *34*(19), 5623–30. doi:10.1093/nar/gkl723

Noguchi, H., Taniguchi, T., & Itoh, T. (2008). MetaGeneAnnotator: detecting species-specific patterns of ribosomal binding site for precise gene prediction in anonymous prokaryotic and phage genomes. *DNA Research : An International Journal for Rapid Publication of Reports on Genes and Genomes*, *15*(6), 387–96. doi:10.1093/dnares/dsn027

Pavlović, V., Garg, A., & Kasif, S. (2002). A Bayesian framework for combining gene predictions. *Bioinformatics (Oxford, England)*, *18*(1), 19–27.

Polson, S. W., Wilhelm, S. W., & Wommack, K. E. (2011). Unraveling the viral tapestry (from inside the capsid out). *The ISME Journal*, *5*(2), 165–8. doi:10.1038/ismej.2010.81

Prangishvili, D., Forterre, P., & Garrett, R. A. (2006). Viruses of the Archaea: a unifying view. *Nature Reviews. Microbiology*, *4*(11), 837–848.

Rho, M., Tang, H., & Ye, Y. (2010a). FragGeneScan: Predicting genes in short and error-prone reads. *Nucleic Acids Research*, *38*(20).

Rho, M., Tang, H., & Ye, Y. (2010b). FragGeneScan: predicting genes in short and error-prone reads. *Nucleic Acids Research*, *38*(20), e191. doi:10.1093/nar/gkq747

Sakowski, E. G., Munsell, E. V, Hyatt, M., Kress, W., Williamson, S. J., Nasko, D. J., … Wommack, K. E. (2014). Ribonucleotide reductases reveal novel viral diversity and predict biological and ecological features of unknown marine viruses. *Proceedings of the National Academy of Sciences of the United States of America*, *111*(44), 15786–91. doi:10.1073/pnas.1401322111

Salzberg, S. L., Delcher, A. L., Kasif, S., & White, O. (1998). Microbial gene identification using interpolated Markov models. *Nucleic Acids Research*, *26*(2), 544–548. doi:10.1093/nar/26.2.544

Schmidt, H. F., Sakowski, E. G., Williamson, S. J., Polson, S. W., & Wommack, K. E. (2014). Shotgun metagenomics indicates novel family A DNA polymerases predominate within marine virioplankton. *The ISME Journal*, *8*(1), 103–14. doi:10.1038/ismej.2013.124

Shah, S. P., McVicker, G. P., Mackworth, A. K., Rogic, S., & Ouellette, B. F. F. (2003). GeneComber: combining outputs of gene prediction programs for improved results. *Bioinformatics*, *19*(10), 1296–1297. doi:10.1093/bioinformatics/btg139

Shors, T. (2011a). Understanding Viruses. In *Understanding of Viruses* (2nd ed.). Jones & Bartlett Learning.

Shors, T. (2011b). Understanding Viruses. In *Understanding Viruses*. Jones & Batlett Learning.

Stanke, M., & Waack, S. (2003). Gene prediction with a hidden Markov model and a new intron submodel. *Bioinformatics*, *19*(Suppl 2), ii215–ii225. doi:10.1093/bioinformatics/btg1080

Turnbaugh, P. J., Ley, R. E., Hamady, M., Fraser-Liggett, C. M., Knight, R., & Gordon, J. I. (2007). The human microbiome project. *Nature*, *449*(7164), 804–10. doi:10.1038/nature06244

Virus Taxonomy: Ninth Report of the International Committee on Taxonomy of Viruses. (2012) (p. 1327). Elsevier. Retrieved from http://books.google.com/books?hl=en&lr=&id=KXRCYay3pH4C&pgis=1

Whitman, W. B., Coleman, D. C., & Wiebe, W. J. (1998). Prokaryotes: the unseen majority. *Proceedings of the National Academy of Sciences of the United States of America*, *95*(12), 6578–6583.

Wommack, K. E., Bhavsar, J., Polson, S. W., Chen, J., Dumas, M., Srinivasiah, S., … Nasko, D. J. (2012). VIROME: a standard operating procedure for analysis of viral metagenome sequences. *Standards in Genomic Sciences*.

Wommack, K. E., & Colwell, R. R. (2000). Virioplankton: Viruses in Aquatic Ecosystems. *Microbiology and Molecular Biology Reviews*, *64*(1), 69–114. doi:10.1128/MMBR.64.1.69-114.2000

Yada, T., Takagi, T., Totoki, Y., Sakaki, Y., & Takaeda, Y. (2003). DIGIT: a novel gene finding program by combining gene-finders. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 375–387.

Yok, N. G., & Rosen, G. L. (2011). Combining gene prediction methods to improve metagenomic gene annotation. *BMC Bioinformatics*, *12*(1), 20. doi:10.1186/1471-2105-12-20

Zhu, W., Lomsadze, A., & Borodovsky, M. (2010). Ab initio gene identification in metagenomic sequences. *Nucleic Acids Research*, *38*(12), e132. doi:10.1093/nar/gkq275

Appendix A

LIST OF FILES AND SCRIPTS USED

a. annotationParser.py: This file converts the annotations downloaded from NCBI for the eleven whole genomes into a common Annotation Standard Output format (ASOF)
b. virusName complete genome_annotations: This file contains the Annotations for each of the viruses downloaded directly from NCBI.
c. virusName_coordinates.fasta: This file contains the downloaded annotations converted to a common ASOF format by annotationParser by parsing file - virusName complete genome_annotations.
d. finalParser_for_whole_genomes.py: This file compares the predicted annotations by NCBI and the predictions made by ORF callers for the eleven whole genomes and also generates the statistics. This script produces tow output files – results_file_whole_toolName.tab and statistics_file_whole_toolName.tab
e. results_file_whole_toolName.tab: This file contains the results of whole genome comparison between the annotation and ORF caller prediction file.
f. Statistics_file_whole_toolName.tab: This file takes results_file_whole_toolName.tab and generates statistics for whole genomes.
g. MGA_parser.py: This script converts the whole genomes as wells as shredded genomes predictions made by MGA into a common Tool Standard Output format (TSOF).

h. virusName_whole_genome_mga: This file contains the output predictions made by MetaGeneAnnotator for the particular whole genome.
i. virusName_whole_genome_output_mga.fasta: This file contains the output in TSOF format parsed by MGA_parser.py obtained from virusName_whole_genome_mga.
j. virusName_shredded_mga: This file contains the output predictions made by MetaGeneAnnotator for the particular shredded virus.
k. virusName_shredded_output_mga.fasta: This file contains the output in TSOF format parsed by MGA_parser.py obtained from virusName_shredded_mga.
l. MGM_parser.py: This script converts the whole genomes as wells as shredded genomes predictions made by MGM into a common Tool Standard Output format (TSOF).
m. virusName_whole_genome_mgm.gff: This file contains the output predictions made by MetaGeneMark for the particular whole genome.
n. virusName_whole_genome_output_mgm.fasta: This file contains the output in TSOF format parsed by MGM_parser.py obtained from virusName_whole_genome_mgm.
o. virusName_shredded_mgm.gff: This file contains the output predictions made by MetaGeneMark for the particular shredded virus.
p. virusName_shredded_output_mgm.fasta: This file contains the output in TSOF format parsed by MGM_parser.py obtained from virusName_shredded_mgm.gff file.

q. Orphelia_parser.py: This script converts the whole genomes as wells as shredded genomes predictions made by Orphelia into a common Tool Standard Output format (TSOF).
r. virusName_whole_genome_orphelia.pred: This file contains the output predictions made by Orphelia for the particular whole genome.
s. virusName_whole_genome_output_mga.fasta: This file contains the output in TSOF format parsed by Orphelia_parser.py obtained from virusName_whole_genome_orphelia.
t. virusName_shredded_orphelia.pred: This file contains the output predictions made by Orphelia for the particular shredded virus.
u. virusName_shredded_output_mgm.fasta: This file contains the output in TSOF format parsed by Orphelia_parser.py obtained from virusName_shredded_orphelia.pred file.
v. samParser.py: This script maps the shredded sequences (metagenomes) to the whole genomes and gives information on the exact position where they match.
w. finalParser_for_metagenomes.py: This script compares the Mapped metagenomes to reference annotation file with the prediction file of metagenomes from ORF callers. This script produces
x. generateStatistics.py: This script takes the finalParser_for_metagenomes.py and other generated files resulted from comparison and generates statistics (i.e. count of each category like exact matched, partial matched, etc.) for metagenomes prediction.
y. results_file_toolName.tab: This file contains the results for metagenomes comparison like Exact matched and few partial matched cases.

z. 3_matched_5!_diff_toolName.tab: This script contains one of the partial matched cases.

aa. 5_matched_3!_diff_toolName.tab: This script contains one of the partial matched cases.

bb. in_tool_not_annotation_toolName.tab: This file contains predictions made by ORF caller but not by annotation file.

cc. in_annnotation_not_tool_toolName.tab: This file contains predictions made by annotation file but not by ORF caller.

dd. orf_length_generator.py: This script takes the ORFs predicted by annotation but not by tool and gives the number of reads above or below a given cutoff ORF length.

ee. scatter_plot_parser.py: This script generates a file with read names, their lengths and scores to generate scatter plot

ff. mga_parser_with_len_cutoff: This is similar to original MGA parser script but also takes length as an arguments and outputs reads only above the specified cutoff.

gg. mga_parser_with_score_cutoff: This is similar to original MGA parser script but also takes ORF caller score as an arguments and outputs reads only above the specified cutoff.

hh. score_len_genrator.py: This script gives an output with ORFs and their corresponding lengths.

.

Appendix B

WHOLE GENOME STATISTICS FOR ALL VIRUSES

Table B.1 Statistics for the virus Enterobacteria Phage T4 for ORF callers – MGA, MGM, Orphelia and Glimmer-MG

Enterobacteria Phage T4 (168,903 bp)				
Type	**MGA**	**MGM**	**Orphelia**	**Glimmer-MG**
Total annotation count	278	278	278	278
Total tool count	264	263	244	247
Exact matched	240	225	206	219
Partial matched	14	28	31	22
Frame shift	0	0	0	0
Found in tool but not in annotation (false positives)	10	10	7	6
Found in annotation but not in tool (false negatives)	24	25	41	37

Table B.2 Statistics for the virus Sulfitobacter phage pCB2047-C for ORF callers – MGA, MGM and Orphelia

Sulfitobacter phage pCB2047-C (40, 931 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	73	73	73
Total tool count	77	70	59
Exact matched	57	53	44
Partial matched	11	12	7
Frame shift	0	0	0
Found in tool but not in annotation (false positives)	9	5	8
Found in annotation but not in tool (false negatives)	5	8	22

Table B.3 Statistics for the virus Pseudomonas Phage tf for ORF callers – MGA, MGM and Orphelia

Pseudomonas Phage tf (46,271 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	72	72	72
Total tool count	68	65	55
Exact matched	57	44	40
Partial matched	7	16	8
Frame shift	0	0	0
Found in tool but not in annotation (false positives)	4	5	7
Found in annotation but not in tool (false negatives)	8	12	24

Table B.4 Statistics for the virus Cyanophage KBS P1A for ORF callers – MGA, MGM and Orphelia

Cyanophage KBS P1A (45,730 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	57	57	57
Total tool count	57	50	41
Exact matched	36	35	23
Partial matched	14	10	13
Frame shift	0	0	0
Found in tool but not in annotation (false positives)	7	5	5
Found in annotation but not in tool (false negatives)	7	12	21

Table B.5 Statistics for the virus Cyanophage S-TIM5 for ORF callers – MGA, MGM and Orphelia

Cyanophage S-TIM5 (161,440 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	180	180	180
Total tool count	170	169	95
Exact matched	156	157	73
Partial matched	8	4	13
Frame shift	0	0	0
Found in tool but not in annotation (false positives)	6	8	7
Found in annotation but not in tool (false negatives)	16	19	92

Table B.6 Statistics for the virus Klebsiella Phage JD001 for ORF callers – MGA, MGM and Orphelia

Klebsiella Phage JD001 (48,814 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	68	68	68
Total tool count	66	64	57
Exact matched	55	55	41
Partial matched	8	7	13
Frame shift	0	0	0
Found in tool but not in annotation (false positives)	3	2	3
Found in annotation but not in tool (false negatives)	5	6	14

Table B.7 Statistics for the virus Pelagibacter Phage HTVC011P for ORF callers – MGA, MGM and Orphelia

Pelagibacter Phage HTVC011P (39,921 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	45	45	45
Total tool count	46	46	22
Exact matched	36	33	45
Partial matched	4	5	6
Frame shift	0	0	0
Found in tool but not in annotation (false positives)	6	8	1
Found in annotation but not in tool (false negatives)	5	7	24

Table B.8 Statistics for the virus Puniceispirillum Phage HMO-2011 for ORF callers – MGA, MGM and Orphelia

Puniceispirillum Phage HMO-2011 (55,282 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	74	74	74
Total tool count	88	80	62
Exact matched	59	51	42
Partial matched	12	14	8
Frame shift	0	0	0
Found in tool but not in annotation (false positives)	17	15	12
Found in annotation but not in tool (false negatives)	3	9	24

Table B.9 Statistics for the virus Prochlorococcus Phage P-SSM2 for ORF callers – MGA, MGM and Orphelia

Prochlorococcus Phage P-SSM2 (252,401 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	334	334	334
Total tool count	324	322	147
Exact matched	290	295	114
Partial matched	6	5	20
Frame shift	0	0	0
Found in tool but not in annotation (false positives)	28	22	12
Found in annotation but not in tool (false negatives)	38	34	200

Table B.10 Statistics for the virus Cellulophaga Phage phi14:2 for ORF callers – MGA, MGM and Orphelia

Cellulophaga Phage phi14:2(100,418 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	133	133	133
Total tool count	117	115	43
Exact matched	107	103	32
Partial matched	7	10	8
Frame shift	0	0	0
Found in tool but not in annotation (false positives)	3	2	3
Found in annotation but not in tool (false negatives)	19	20	93

Table B.11 Statistics for the Cyanophage NATL2A-133 for ORF callers – MGA, MGM and Orphelia

Cyanophage NATL2A-133 (47,536 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	62	62	62
Total tool count	63	62	38
Exact matched	46	44	25
Partial matched	9	10	9
Frame shift	0	0	0
Found in tool but not in annotation (false positives)	8	8	4
Found in annotation but not in tool (false negatives)	7	8	28

Appendix C

STATISTICS FOR ALL VIRUSES FOR SHREDDED GENOMES

Table C.1 Statistics for the Enterobacteria Phage T4 for ORF callers – MGA, MGM and Orphelia

Enterobacteria Phage T4 (168,903 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	79,178	79,178	79,178
Total tool count	72,239	64,698	51,276
Exact matched	46,278	28,546	19,594
Exact Matched - 3'matched 5' end off by 1 / 2	838	8,175	124
Exact Matched - 5' matched 3' end off by 1 / 2	364	7,148	147
Exact matched – both 3' and 5' end off by 1/2	12	76	1
Partial Match	9508	6039	2626
ORFs in tool not in annotation (false positives)	11038	14315	28625
ORFs in annotation not tool (false negatives)	21,070	29,544	56,570

Table C.2 Statistics for the Sulfitobacter phage pCB2047-C for ORF callers – MGA, MGM and Orphelia

Sulfitobacter phage pCB2047-C (40,931 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	19,917	19,917	19,917
Total tool count	17268	14902	13555
Exact matched	11605	6789	9772
Exact Matched - 3'matched 5' end off by 1 / 2	252	2096	178
Exact Matched - 5' matched 3' end off by 1 / 2	32	1590	28
Exact matched – both 3' and 5' end off by 1/2	3	8	1
Partial Match	2046	1267	1375
ORFs in tool not in annotation (false positives)	2619	3171	2057
ORFs in annotation not tool (false negatives)	5820	8305	8516

Table C.3 Statistics for the Pseudomonas Phage tf for ORF callers – MGA, MGM and Orphelia

Pseudomonas Phage tf (46,271 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	21,570	21,570	21,570
Total tool count	19,261	15462	15,298
Exact matched	12,326	7092	10,800
Exact Matched - 3'matched 5' end off by 1 / 2	448	1746	303
Exact Matched - 5' matched 3' end off by 1 / 2	96	1348	37
Exact matched – both 3' and 5' end off by 1/2	1	21	1
Partial Match	2,139	1344	1,357
ORFs in tool not in annotation (false positives)	3,216	3842	2,598
ORFs in annotation not tool (false negatives)	6,283	10043	8,962

Table C.4 Statistics for the Cyanophage KBS P1A for ORF callers – MGA, MGM and Orphelia

Cyanophage KBS P1A (45,730 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	19,537	19,537	19,537
Total tool count	18,067	15,330	14661
Exact matched	11,519	8,134	10,782
Exact Matched - 3'matched 5' end off by 1 / 2	277	1,376	72
Exact Matched - 5' matched 3' end off by 1 / 2	32	937	28
Exact matched – both 3' and 5' end off by 1/2	0	3	0
Partial Match	2,152	1,321	1,386
ORFs in tool not in annotation (false positives)	3,102	3,482	2,252
ORFs in annotation not tool (false negatives)	5,337	7,825	7,206

Table C.5 Statistics for the Cyanophage S-TIM5 for ORF callers – MGA, MGM and Orphelia

Cyanophage S-TIM5 (161,440 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	69,022	69,022	69,022
Total tool count	62,979	56,068	47,280
Exact matched	41,432	30,970	36,719
Exact Matched - 3'matched 5' end off by 1 / 2	662	4,956	224
Exact Matched - 5' matched 3' end off by 1 / 2	129	4,303	99
Exact matched – both 3' and 5' end off by 1/2	10	31	1
Partial Match	7,028	4,554	3,792
ORFs in tool not in annotation (false positives)	10,553	10,917	6,108
ORFs in annotation not tool (false negatives)	18,928	24,382	28,065

Table C.6 Statistics for the Cellulophaga Phage phi14:2 for ORF callers – MGA, MGM and Orphelia

Cellulophaga Phage phi14:2 (100,418 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	45,230	45,230	45,230
Total tool count	41,218	36198	27,927
Exact matched	26,734	18629	22,186
Exact Matched - 3'matched 5' end off by 1 / 2	422	3719	100
Exact Matched - 5' matched 3' end off by 1 / 2	84	3209	63
Exact matched – both 3' and 5' end off by 1/2	3	15	0
Partial Match	5,201	3389	2,506
ORFs in tool not in annotation (false positives)	10,672	7096	2,980
ORFs in annotation not tool (false negatives)	7,671	16544	20,323

Table C.7 Statistics for the Cyanophage NATL2A-133 for ORF callers – MGA, MGM and Orphelia

Cyanophage NATL2A-133 (47,536 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	18,758	18,758	18,758
Total tool count	19,209	16524	13,994
Exact matched	10817	6848	8,932
Exact Matched - 3'matched 5' end off by 1 / 2	163	1626	47
Exact Matched - 5' matched 3' end off by 1 / 2	107	1375	56
Exact matched – both 3' and 5' end off by 1/2	5	17	1
Partial Match	2,197	1416	1,274
ORFs in tool not in annotation (false positives)	5,075	5151	3,599
ORFs in annotation not tool (false negatives)	5,250	7525	8,401

Table C.8 Statistics for the Klebsiella Phage JD001 for ORF callers – MGA, MGM and Orphelia

Klebsiella Phage JD001 (48,814 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	21,137	21,137	21,137
Total tool count	19,135	16,308	15,400
Exact matched	12,269	8,049	11,264
Exact Matched - 3'matched 5' end off by 1 / 2	353	1,944	172
Exact Matched - 5' matched 3' end off by 1 / 2	72	1,635	59
Exact matched – both 3' and 5' end off by 1/2	2	21	2
Partial Match	2,267	1,300	1,596
ORFs in tool not in annotation (false positives)	3,219	3,323	2,143
ORFs in annotation not tool (false negatives)	5,989	8,262	7,995

Table C.9 Statistics for the Pelagibacter Phage HTVC011P for ORF callers – MGA, MGM and Orphelia

Pelagibacter Phage HTVC011P (39,921 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	16,756	16,756	16,756
Total tool count	16,420	14961	11,207
Exact matched	9,999	7181	8,561
Exact Matched - 3'matched 5' end off by 1 / 2	153	1353	56
Exact Matched - 5' matched 3' end off by 1 / 2	36	1041	26
Exact matched – both 3' and 5' end off by 1/2	2	9	1
Partial Match	2,191	1479	1,049
ORFs in tool not in annotation (false positives)	2,952	3490	1,448
ORFs in annotation not tool (false negatives)	4,052	5762	7,033

Table C.10 Statistics for the Puniceispirillum Phage HMO-2011 for ORF callers – MGA, MGM and Orphelia

Puniceispirillum Phage HMO-2011 (55,282 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	24,497	24,497	24,497
Total tool count	23,574	19803	17,846
Exact matched	15,054	9682	13,278
Exact Matched - 3'matched 5' end off by 1 / 2	346	2332	95
Exact Matched - 5' matched 3' end off by 1 / 2	54	1657	28
Exact matched – both 3' and 5' end off by 1/2	1	9	0
Partial Match	2,793	1448	159
ORFs in tool not in annotation (false positives)	3,963	4587	2,548
ORFs in annotation not tool (false negatives)	5,939	9441	9,385

Table C.11 Statistics for the Prochlorococcus Phage P-SSM2 for ORF callers – MGA, MGM and Orphelia

Prochlorococcus Phage P-SSM2 (252,401 bp)			
Type	**MGA**	**MGM**	**Orphelia**
Total annotation count	113,439	113,439	113,439
Total tool count	104,882	94771	73,572
Exact matched	67,250	48041	57,599
Exact Matched - 3'matched 5' end off by 1 / 2	1,050	10028	267
Exact Matched - 5' matched 3' end off by 1 / 2	237	9064	172
Exact matched – both 3' and 5' end off by ½	7	46	2
Partial Match	12,827	8823	6,704
ORFs in tool not in annotation (false positives)	17,964	18197	8,419
ORFs in annotation not tool (false negatives)	27,791	38008	48,509

Appendix D

PICTORIAL REPRESENTATION OF WHOLE GENOME STATISTICS FOR SOME REPRESENTATION INDIVIDUAL VIRUSES

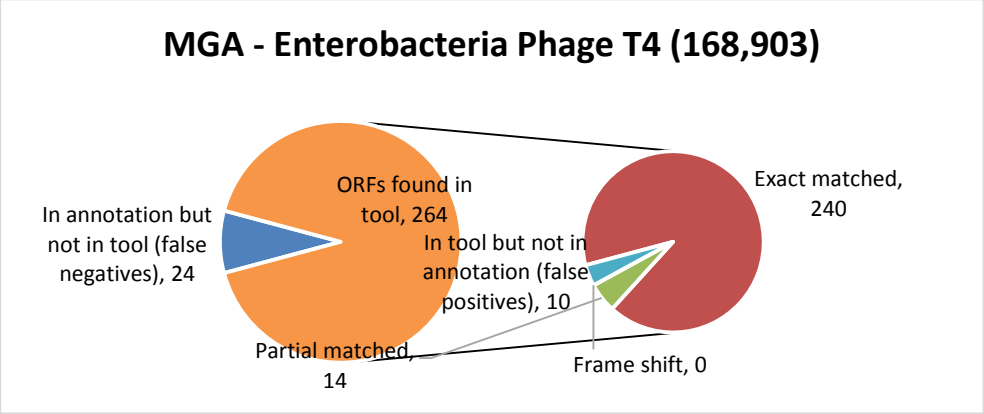


Figure D.1 Pictorial representation of statistics for Enterobacteria Phage T4 – MetaGeneAnnotator (MGA)

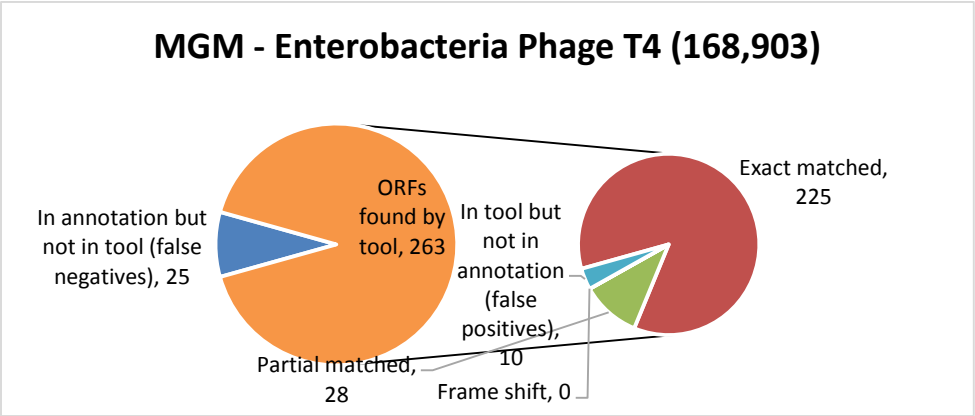


Figure D.2 Pictorial representation of statistics for Enterobacteria Phage T4 – MetaGeneMark (MGM)

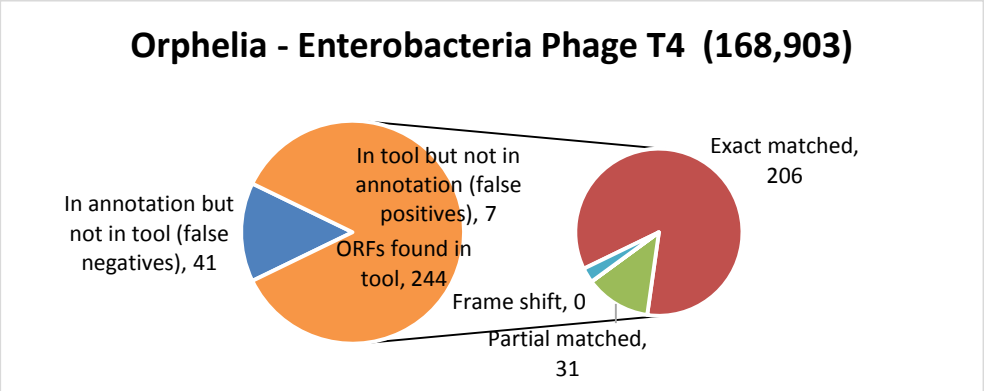


Figure D.3 Pictorial representation of statistics for Enterobacteria Phage T4 – Orphelia

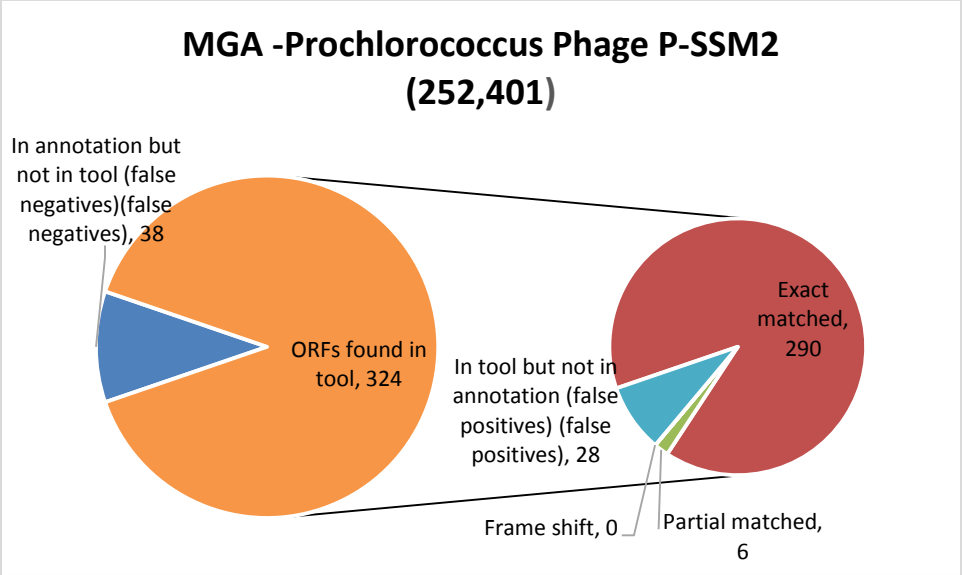


Figure D.4 Pictorial representation of statistics for Prochlorococcus Phage P- SSM2 – MGA

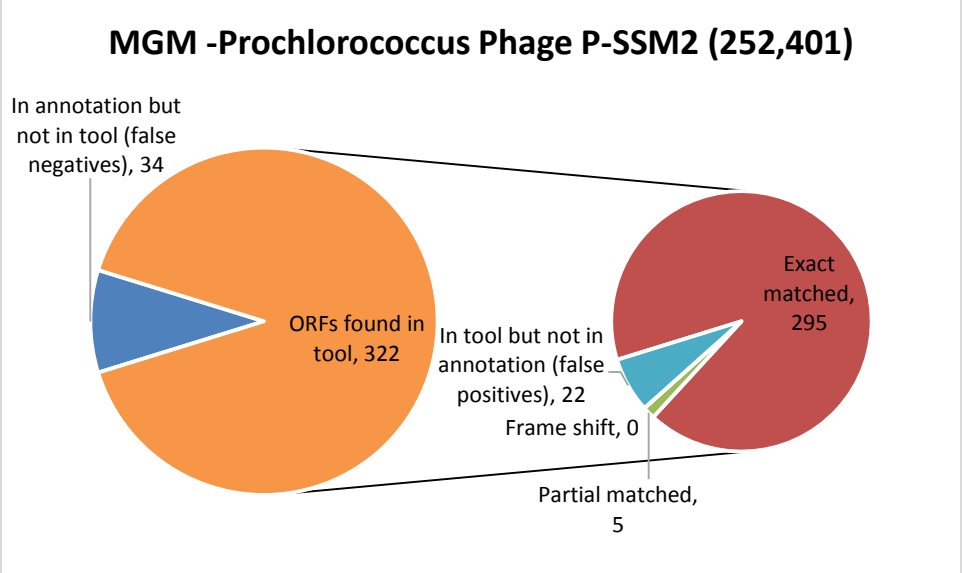


Figure D.5 Pictorial representation of statistics for Prochlorococcus Phage P-SSM2 – MGM

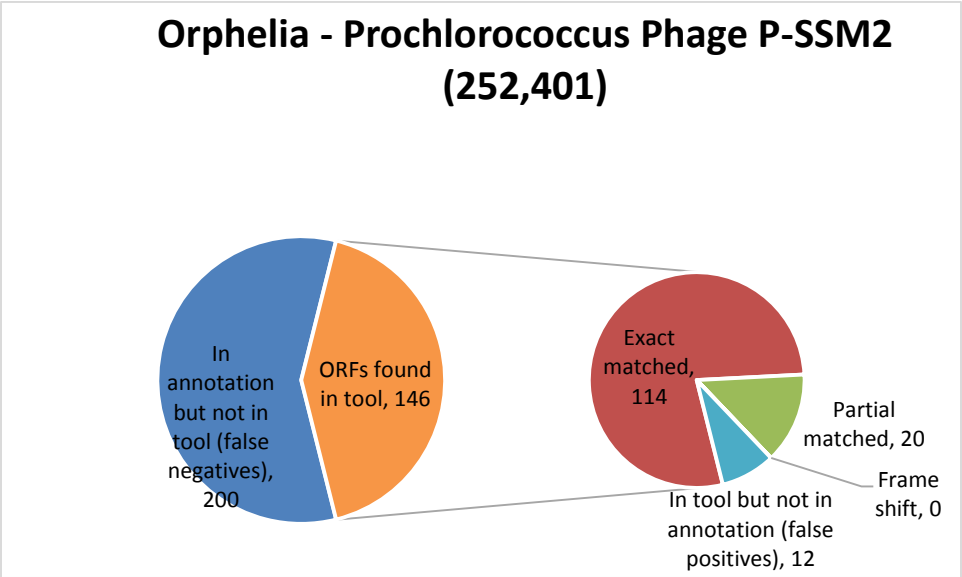


Figure D.6 Pictorial representation of statistics for Prochlorococcus Phage P-SSM2 – Orphelia

Appendix E

PICTORIAL REPRESENTATION OF SHREDDED GENOME STATISTICS FOR A REPRESENTATIVE VIRUS

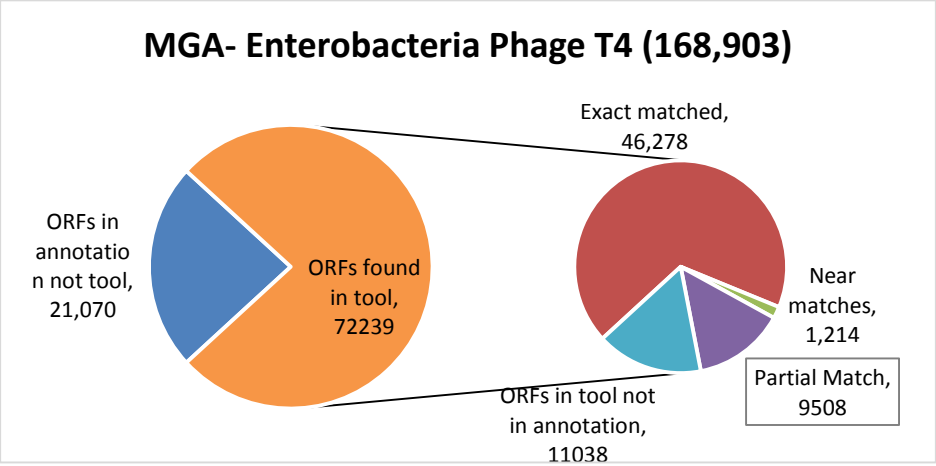


Figure E.1 (a) Pictorial representation of statistics for Enterobacteria Phage T4 –MGA

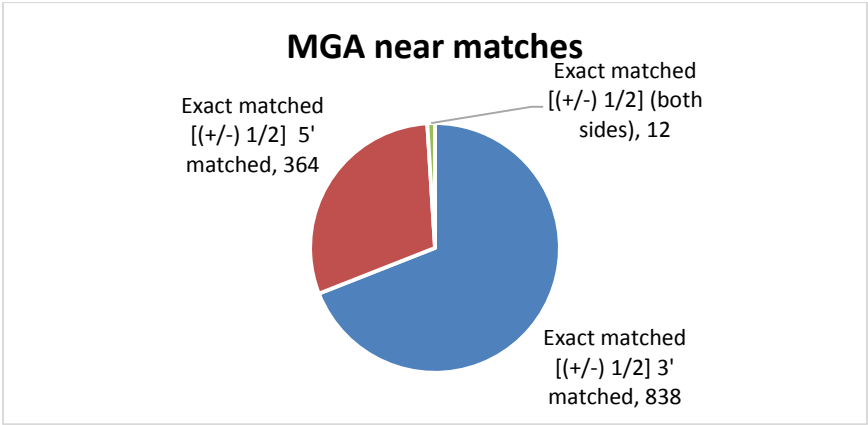


Figure E.1 (b) Pictorial representation of Near matches statistics for Enterobacteria PhageT4 – MGA

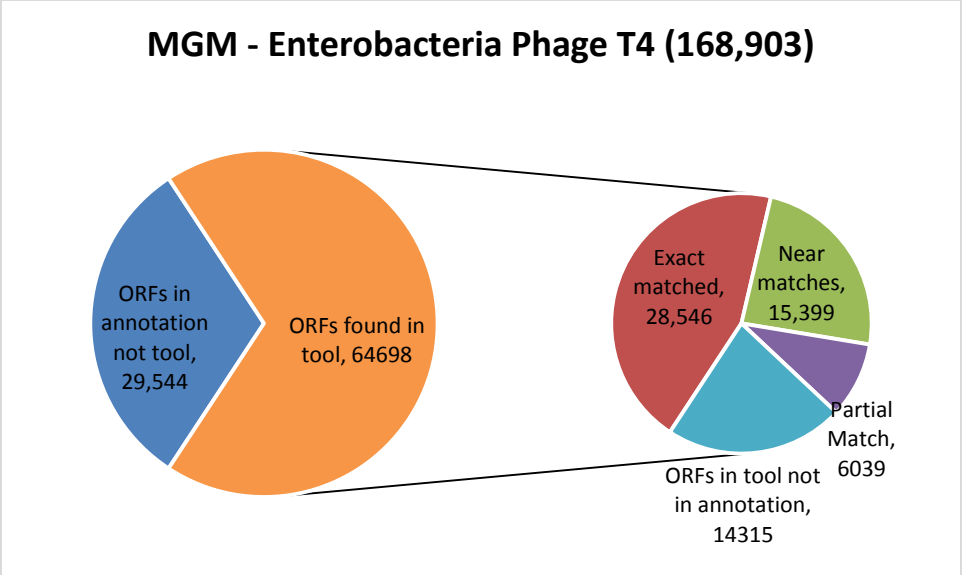


Figure E.2 (a) Pictorial representation of statistics for Enterobacteria Phage T4 – MGM

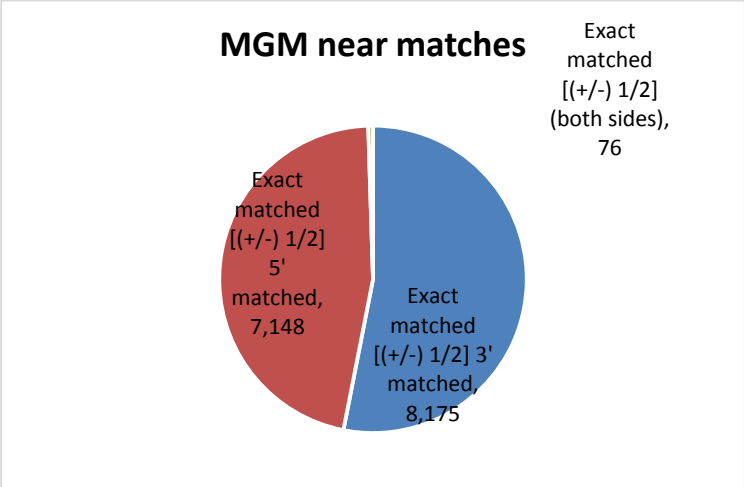


Figure E.2 (b) Pictorial representation of Near matches statistics for Enterobacteria PhageT4 – MGM

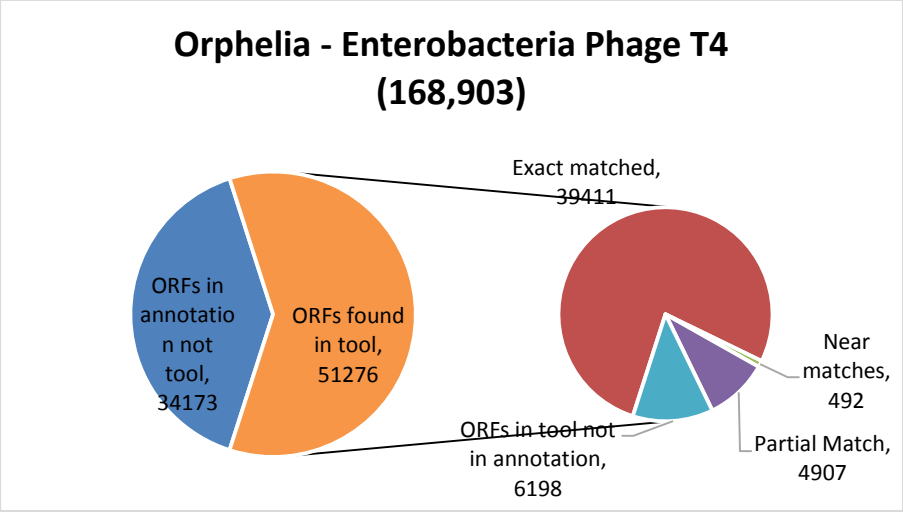


Figure E.3 (a) Pictorial representation of statistics for Enterobacteria Phage T4 – Orphelia

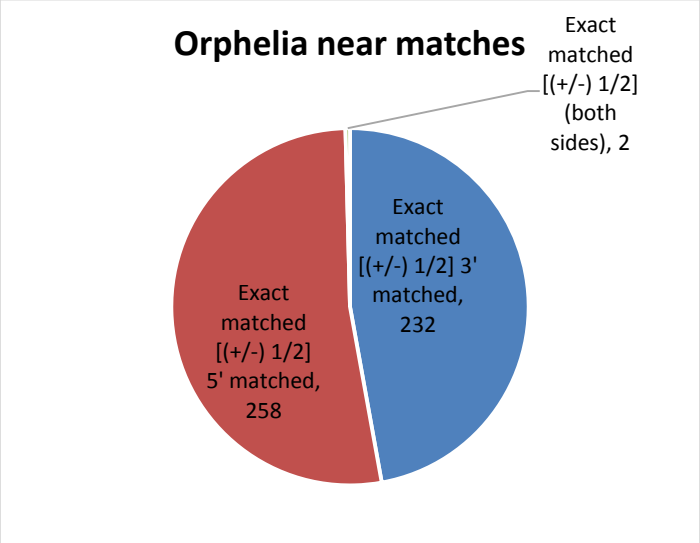


Figure E.3 (b) Pictorial representation of Near matches statistics for Enterobacteria PhageT4 - Orphelia