UNDERSTANDING BIAS IN NEXT-GENERATION SEQUENCING TECHNOLOGIES AND ANALYSES

by

Adam Ryan Price

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Bioinformatics and Computational Biology

Charlotte

2017

Approved by:

____
Dr. Cynthia Gibas

____
Dr. Anthony Fodor

____
Dr. Jennifer Weller

____
Dr. Xinghua Shi

____
Dr. Ian Marriott

ABSTRACT

ADAM RYAN PRICE. Understanding bias in next-generation sequence technologies and analyses. (Under the direction of DR. CYNTHIA GIBAS)

Accurate and unbiased measurements are critical for a wide variety of studies that endeavor to understand the function and basis of biological features. Next-generation sequencing has made it necessary to increase the pace of development for analysis methodologies, creating opportunities for bias to enter the system and influence downstream interpretation unless appropriate measures are taken. This dissertation addresses several such points where bias can be introduced into analysis pipelines, and offers guidelines and tools for addressing and investigating bias in next-generation sequencing analysis. First, we investigated the impact of GC bias in the Illumina sequencing platform, identifying RNA secondary structure as a large contributing factor. By identifying and quantifying this effect, assays that take structure into account can attempt to minimize the resulting GC bias when performing next-generation sequencing. Next, we examined the issue of using non-native reference genomes for read alignment. Using both in-house and publicly available data sets with different properties, and additionally simulating reference and read data with specific properties, we were able to show what factors introduce read alignment loss and misalignments, and outline what steps can be taken to avoid these biases when performing studies using a non-native reference genome. Finally, we developed a powerful and user-friendly tool, Simulome, for simulating genomes and variants with specific properties. Simulome makes it possible to control for variables in data such that the efficacy of analysis methodologies can be studied with regard to specific variations in data. Using this tool we were able to

model the influence of specific causes behind false positives individually in the previous study of non-native reference genomes.

ACKNOWLEGMENTS

My greatest appreciation goes to my advisor, Dr. Cynthia Gibas, for her support, guidance, encouragement and patience over the years. She has been a distinguished mentor who has always encouraged me to push myself and my boundaries, and has believed in me in those times that I did not believe in myself. This dissertation would have not been possible without her invaluable guidance and help.

I most sincerely thank my committee members Dr. Anthony Fodor, Dr. Jennifer Weller, Dr. Xinghua Shi, and Dr. Lixia Yao for their valuable help and kind guidance throughout the years I have had the honor of knowing them. I would like to thank Dr. Jennifer Weller specifically, for often taking time from her busy schedule to speak with me at great length and with heroic patience, and for her constant support and guidance. Dr. Weller has been incredibly kind and helpful, and I consider myself fortunate to have had her as a mentor.

I would like to thank my friends and colleagues, Ehsan Tabari and Meng Niu. These two have been amazing friends during my years at UNC-Charlotte, as well as great colleagues. I am grateful for the conversation, company, and their aid in reflecting on difficult challenges. It has been an honor to know and work with them.

Additionally, I would like to express my deepest gratitude to my family for their unconditional love and support. My wife, Ai Kamei, especially has helped me through this PhD. Her support has meant to me more than I can ever express, and without a doubt is the greatest contributing factor to my ability to complete my PhD.

Finally, I am very grateful to my advisor, the Department of Bioinformatics and Genomics, the UNC-Charlotte graduate school, and the Giles foundation for providing me financial support in the past years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BLAST	Basic local alignment search tool
FRT-seq	Flowcell Reverse Transcription Sequencing
GB	Gigabytes
HPC	High performance computing
MPI	Message passing interface
NPPH	Number of peptides per hundred amino acids
ORF	Open reading frame
PARS	parallel analysis of RNA structure
PCR	polymerase chain reaction
RPK	Reads per kilo-base
rRNA	ribosomal RNA
SQL	Structured query language
TPM	Transcripts per million
TU	Transcription unit

INTRODUCTION

The problem of bias resulting from biological and technological sources is one that must be consistently reexamined alongside the development of new technologies and methods of analysis. While it is common practice, and necessary for the progress of science, to trust the tools that we use to make deductions and inferences about the complex and often subtle nature of biology, it is also imperative that we vigilantly monitor the efficacy and accuracy of our technologies and methods. In recent years, next-generation sequencing technologies have been producing unprecedented amounts of data, which has subsequently lead to the development and application of many new algorithms and analysis methodologies. A consequence of this rapid growth is that bias caused by biological and technological sources has the potential to propagate into future research unless the sources of bias can be identified and corrected for.

High-throughput sequencing is subject to sequence dependent bias. A widely-studied source of bias in sequencing is the GC content bias, in which levels of GC content in a genomic region effect the number of reads produced during sequencing. Single-stranded RNA molecules are known to fold on themselves due to free energy interactions to form complex three dimensional structures that will vary depending on molecular sequence (Brierley, Pennell and Gillbert 2007). These structures are more than simply an artifact that has no relationship to biological function, and have been shown in some cases to be necessary for function (Buratti, et al. 2004; Lyubetsky, et al. 2005). While there is a distinct set of biological functionality for the formation of secondary structure in mRNA, this folding has the additional unintended consequence of introducing bias when mRNA strands are subject to sequencing in next-generation sequencing technologies. This effect has been observed in high throughput sequencing technologies, such as Illumina sequencers, that use PCR as a method of read replication (Quail, et al. 2012). It has also been observed that regions of RNA with higher GC content have more stable secondary structures than RNA strands with lower GC content (Chan, et al. 2009). Some research has been performed to correct for GC bias, but there has been so far been little effort to explicitly model the underlying mechanism. Recently, a method for detecting secondary structure across the entire transcriptome has been developed called PARS. PARS uses a multiple enzyme digestion protocol to identify the specific location of single and double stranded structure in nucleic acid molecules and makes it possible to investigate the underlying molecular origin of observed GC bias in sequencing (Kertesz, et al. 2010). Specifically, the PARS method makes it possible to identify folded and non-folded regions in mRNA molecules, which can then be compared with sequence read levels as produced by next-generation sequencing to quantify the bias that is caused by mRNA secondary structure.

Sequence read alignment is currently the basis of many biological studies, including analysis of RNA-Seq transcriptome data. Many common analysis pipelines rely on proper alignment of reads to a corresponding reference genome for the target organism under investigation. Differential expression studies, for example, typically proceed by aligning transcriptome reads to a reference and extracting count data based on the alignment to examine the differences in transcript read levels for the genome under study. In cases where multiple related organisms are being studied, it is not uncommon to map reads from all organisms to a common reference genome. The assumption is that the differences between these organisms are small enough that they will not influence analysis. In these sorts of experiments, it is generally assumed that genes that do not map in one sample or the other can be excluded, and other genes that do show seemingly reasonable levels of alignment in both organisms can be trusted and used in differential expression analysis. Comparison of real bacterial genomes from closely related strains, however, calls these assumptions into question. While we have demonstrated that exclusion of peripheral genes from the analysis does not have a large impact on the overall differential expression results in the core genome, we show here that misalignment can lead to inaccuracies in reference alignment and subsequently read counts, especially when tolerance for mismatches is set low to increase specificity. This can result in false positive identification of genes as differentially expressed. Comparative investigations of microbial organisms since the data explosion of high-throughput sequencing have shown that prokaryotic genomes are very dynamic and diverge rapidly, even for closely related strains of the same organism. The dynamic nature of prokaryotes allows for a significant degree of gene content and sequence variation. The genomes used in this our comparative analysis differ from their near relatives by approximately 4% overall, with some homologous genes showing even greater divergence. The practice of using common reference genomes as a basis of comparison will only increase as even more data becomes available. Available data is not always sufficiently complete for use as a reference genome in an RNA-Seq experiment, and reference-free clustering methods for transcriptomics have their own set of interpretation challenges. As a result, researchers often resort to using readily available, fully closed reference genomes in their studies, even when their data was produced from a heterologous strain. This approach fails to consider factors that can influence read counts, such the frequency and density of mismatches due to natural divergence between strains, and how alignment algorithms handle reads with multiple possible mapping positions, especially when mutations decrease mapping position certainty. By examining how this bias functions under controlled conditions and parameter selection, the bias caused by heterologous reference strain alignment can be quantified and corrected for in future studies.

Many forms of bias are difficult to detect and their identification can often be compounded by many overlapping factors, making precise quantification and identification of the source of bias an extremely difficult task. Simulation tools are becoming increasingly relevant to the development, testing, and benchmarking of bioinformatics research, by making it possible to separate factors and control conditions precisely. They provide a valuable control case for a variety of topics, such as the identification of read mapping bias (Degner, *et al.,* 2009*),* correction of read bias in RNA-seq mapping (Satya, *et al.,* 2012), and analysis of the accuracy of gene expression profiling (Hirsch, *et al.,* 2015).

In this dissertation, we address the complexities of bias identification, quantification, and correction in RNA-Seq studies that rely on next generation sequencing technologies and associated analysis methodologies, although the same considerations may also apply to other types of genome-scale sequencing experiments. First, we identify the cause of GC bias resulting from mRNA secondary structure formation in next generation sequencing platforms and quantify levels of bias in three bacteria spanning low, medium, and high GC content. Next, we examine the problem of heterologous reference genome usage when comparing closely related bacterial strains, providing multiple analyses of both real and simulated data, to provide a set of best practices for the use of non-native reference genome comparison. Finally, we introduce Simulome, a simulation tool to generate synthetic reference genomes by sampling and restructuring data from existing genomic sequence data. Simuome makes it possible to separate sources of bias and control for very specific factors by creating reference genomes with specific features and variants of the simulated genome containing controlled mutation events. This functionality makes it possible to analyze the effect of specific mutation types on a large scale, providing researchers with the ability to investigate the efficacy of analysis methodologies and data integrity on many genes that contain similar mutation events, while providing a control genome by which comparisons can be made.

CHAPTER 1: THE IMPACT OF RNA SECONDARY STRUCTURE ON READ START LOCATIONS ON THE ILLUMINA SEQUENCING PLATFORM

1.1 Background

Single-stranded RNA molecules are known to fold into complex three dimensional structures that vary depending on the molecular sequence (Brierley, et al. 2007). It has been shown that these structures are more than simply an artifact of free energy interactions occurring on an unstable single-stranded molecule, and that they are in some cases necessary for function (Buratti, et al. 2004; Lyubetsky, et al. 2005). Many methods have been developed to predict the folded conformations of RNA molecules, and several computational methods have become popular in recent years, such as MFold (Zuker, et al. 2003) and Vienna RNA (Lorenz, et al. 2011), which make predictions of RNA folding conformations based on free energy calculations. An experimental method for detecting secondary structure across the entire transcriptome, called PARS, has also been developed recently (Kertesz, et al. 2010). These technologies and methods make it possible to further investigate the role and effects of secondary structure.

Here, we investigate the effect RNA secondary structure has on gene expression data that is generated through modern sequencing technologies. It has been previously shown that there is a detectable dependence of read depth on GC content (Dohm, et al. 2008). This effect has been observed in high throughput sequencing technologies, such as Illumina sequencers, that use PCR as a method of read replication (Quail, et al. 2012). It has been observed that regions of RNA with higher GC content have more stable secondary structures than RNA strands with lower GC content (Chan, et al. 2009). It has also been shown that the speed at which polymerase moves along an associated RNA strand is dependent on the secondary structure the polymerase encounters, and that polymerase works at a slower pace when confronted with more secondary structure elements (Lyubetsky, et al. 2005). Because the frequency of stable secondary structure increases as GC content increases, the relative abundance of high GC reads produced by next-gen sequencing is likely to be lower due to intermittent pausing by polymerase during fragment amplification. Additionally, amplification methods that rely on single stranded DNA with an associated primer correctly annealing to an oligo, such as flow cell cluster amplification in Illumina sequencing, may be subject to additional bias due to the initial single stranded fragments forming stable structures at the end of the strand.

In our study, we hypothesize that RNA secondary structure formation is the underlying cause of GC bias. We use the PARS assay to measure RNA secondary structure for three bacterial strains with varied levels of GC content (low, medium and high GC content). We show that secondary structure is the GC-correlated molecular property that impacts apparent gene expression levels as measured by transcript abundance. We identify the extent of this effect and use that information to statistically model the relationship between GC content, RNA secondary structure, and the abundance of reads produced by Illumina sequencers.

1.2 Materials and Methods

The goal of this study was to examine the relationship between secondary structure in RNA transcripts and gene expression levels, with particular regard to GC content. To achieve this, it was first necessary to determine the presence or absence of secondary structure for the entire transcriptome at a nucleotide-level resolution. We employed the PARS method, a procedure for measuring transcriptome wide secondary structure (Kertesz, et al. 2010) which has previously been demonstrated in Saccharomyces cerevisiae. PARS works by exposing RNA molecules to enzymes that selectively cleave them depending on their folded state. A sample of RNA was divided and one part was exposed to RNase V1 and another to RNase S1. RNase V1 randomly fragments double-stranded RNA, leaving behind a 5' phosphate at the cleavage site. RNase S1 works similarly but targets single-stranded RNA. Adaptors that can ligate to these 5' phosphoryl-terminated RNAs were then used to select against random fragmentation. The result is that for the RNase V1 sample, the starting position of the

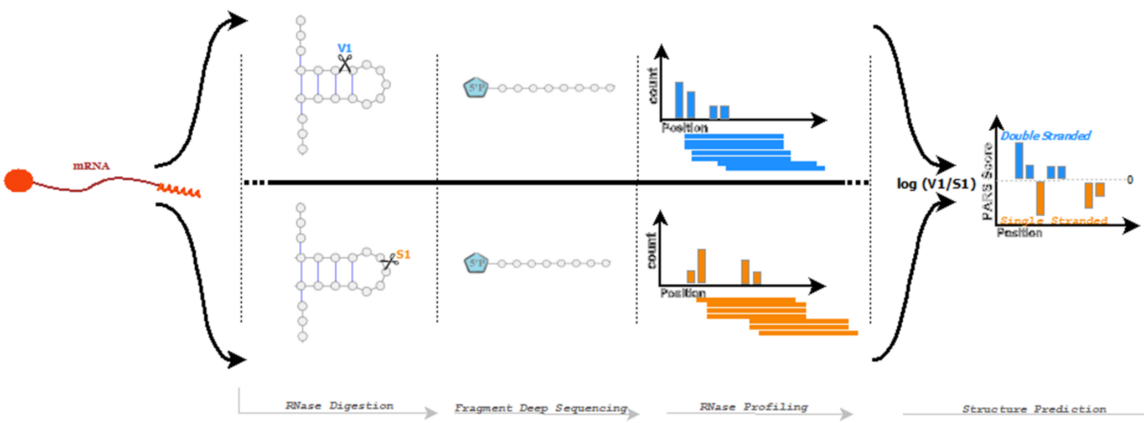

Figure 1.1 Selective enzyme digestion is performed on each organism using S1 RNase and V1 RNase. V1 RNase cuts selectively at double stranded positions, while S1 cuts at single stranded positions. A 5' phosphate is attached to the fragment and deep sequencing is performed. The resulting data is then processed into profiles based on read start positions for each condition and then combined to create structural predictions.

aligned read corresponds to a double-stranded region of RNA, and similarly RNase S1 read start sites corresponds to a single-stranded region. In order to convert this information into a quantitative measure at single nucleotide resolution, the log ratio of the number of reads starting at any given position is calculated based on the RNase V1 and RNase S1 read sets. A higher log ratio, also referred to as a PARS score, indicates a higher probability that nucleotides at a given position are in double-stranded conformations, while a lower score indicates a higher probability of single-stranded conformations (Kertesz, et al. 2010). An overview of this process can be seen in figure 1.1.

1.2.1 Bacterial Sample Selection

Three gram positive bacterial strains with varying levels of genome-wide GC content were chosen from the from NCBI database for the PARS assay. Staphylococcus epidermidis ATCC 12228, with 32% GC, was chosen as a representative strain for low GC content, Exiguobacterium sp. AT1b ATCC-BAA 1283 was chosen as a medium GC content strain with 48.5% GC, and Micrococcus luteus ATCC-4698 was chosen as a representative strain for high GC with 73% GC content. These strains were selected because they had similar genomic sizes and fairly simple culture requirements.

1.2.2 PARS Assay and Sample Preparation

S. epidermidis was cultured in LB, while M. luteus and Exiguobacterium sp. AT1b were cultured in trypticase soy agar. All three cultures were grown to log phase and two volumes of RNA Protect were added to the culture. The cells were pelleted and the resulting pellet was either used directly for RNA extraction or saved at -80C for no longer than twelve hours following extraction. Cell lysis was carried out by freshly made ultrapure lysostaphin, or 125 µl lysostaphin + 200 µl TE NaCl + 5 µl proteinase K in 15 ml cell pellets. The cell pellet mix was vortexed with RLT and 0.1 mm for fifteen to thirty minutes using the disruptor genie vortexer. Similar pretreatment was done for Exiguobacterium sp. AT1b, except that ultrapure Lysozyme was used in place of Lysostaphin during the lysis phase. For M. luteus an incubation period of thirty minutes was optimized for lysis with lysozyme.

The RNAeasy midi kit (Qiagen) was used for final extraction of total RNA from the bacterial cell’s DNA contamination was then removed by treating the total RNA with 4-8 units of DNAse I twice at 37C for thirty minutes in a total volume of 50 µl, with a total nucleic acid concentration of 10 µg. Total RNA was checked for DNA contamination and integrity before proceeding with mRNA enrichment and PCR was subsequently carried out to check for DNA contamination after the DNAse treatment. Agilent 2100 bioanalyzer RNA 600 nano chip was used for total RNA integrity and quantification. Total RNA with RIN equal or greater than 9 was only further used for further experiments. mRNA enrichment was carried out using MICROBExpress kit (Ambion, Thermofisher Scientific) followed by QC on agilent bioanalyzer. mRNA was used to prepare directional paired end libraries of size 300 bp using modified methodology of directional library preparation and the TruSeq small RNA library preparation kit protocol (Illumina Inc.). Briefly, after fragmentation by S1/V1 enzymes, the 5' P end was capture by ligating to 5' adapters and only those fragments which have 5'P end were captured in the cDNA and thus captured in the library. The final libraries were Ampure cleaned and ran on DNA 100 chip for QC before 200bp sequencing.

Three replicates of each sample and condition were prepared and sequenced using the Illumina HiSeq 2500 sequencer. In addition, a control condition was also performed for each organism, in which RNA-seq was performed using standard protocols. Three replicates were also used for the control condition.

1.2.3 Data Processing and Selection

Raw sequence reads were aligned to their respective reference genomes using bowtie2 (Langmead, et al. 2012), with all samples showing a high rate of overall alignment. The resulting files generated by bowtie2 were then filtered for quality and converted to sorted BAM format using samtools (Li, et al. 2009). This process was performed for each of the three conditions: S1 RNase digestion, V1 RNase digestion, and a control sample, for each of the three bacteria used in the experiment. This process was performed on all available replicate data sets, and replicate data were then merged. The data were subsequently used to create a file tallying the number of read start sites for each position along the transcriptome. These positional values were converted into PARS scores for each position along the transcriptome.

The data were then filtered to select a subset of genes that had both the highest confidence PARS scores, due to having adequate data from each experimental condition, and few positions with unknown conformations. To select the most reliable data, data were filtered using a method based on the previously calculated positional values for all three organisms. For each position, in cases where positional data was absent in an experimental condition or in the control condition, a value of 0 was assigned. If data were present in both experimental conditions and in the control condition, PARS scores, the log ratio of the positional data for the two experimental conditions, were calculated for that position. Then, if the PARS score for the position had an absolute value of 3 or greater, a value of 1 was assigned to the position. Positions assigned 1 were considered to be high quality positions, and positions assigned 0 were considered to be poor quality positions. Finally, the percentage of high quality positions for each coding region was calculated, and regions with at least 80% high quality positions were retained for analysis. After this filtering step, fifty-nine genes were selected from Staphylococcus epidermidis, sixty-one genes were selected from Exiguobacterium sp., and eighty-one genes were selected from Micrococcus luteus.

1.3 Results

1.3.1 Secondary Structure and Read Depth

In order to test one of the main hypotheses of this paper, that secondary structure is a possible cause of bias in the measurement of gene expression via high throughput sequencing, we first performed correlation testing. We compared read depth in a standard RNA-seq assay for each organism, to structural predictions at single nucleotide resolution generated in the PARS assay. The 23S rRNA genes for each organism were selected for initial inspection, as these genes were identified as having high quality data for all three organisms in the filtering step. We used Pearson's product-moment correlation testing on these genes to examine the relationship between read depth in the control experiment, and PARS scores. In all three cases, significant positive correlation was found, such that higher levels of read depth corresponded to regions predicted to be in single stranded conformations, with Exiguobacterium sp. having a p-value of 8.944e-05, Staphylococcus epidermidis having a p-value of 0.04939, and Micrococcus luteus having a p-value less than 2.2e-16. This result supports the hypothesis that RNA secondary structure contributes to bias in read depth.

1.3.2 Secondary Structure and GC Bias

We next examined the relationship between GC bias and RNA secondary structure. It has been shown previously that Illumina sequencers exhibit a bias in which A-T residues are sequenced with higher frequency than are G-C residues (Dohm, et al. 2008). It has been further shown that elevated GC content negatively influences sequencing coverage overall, with regions having GC content greater than 70% becoming increasingly read sparse (Sendler, et al. 2011). As such, our investigation of GC content and secondary structure first attempted to confirm this finding, in order to validate subsequent findings in our data. Using the same subset of high-quality genes selected in the analysis of secondary structure's effect on read depth, read depth as generated from the standard RNA-seq assays was contrasted with GC content. The results, shown in figure 1.2, show regions of lower GC content having a more widely varied range of read depth, and regions of higher GC content showing a much more narrow range of read depth, with a tendency toward the lower end relative to the dataset as a whole. Pearson's product-moment correlation test was performed on these data as a whole to characterize this relationship between GC content

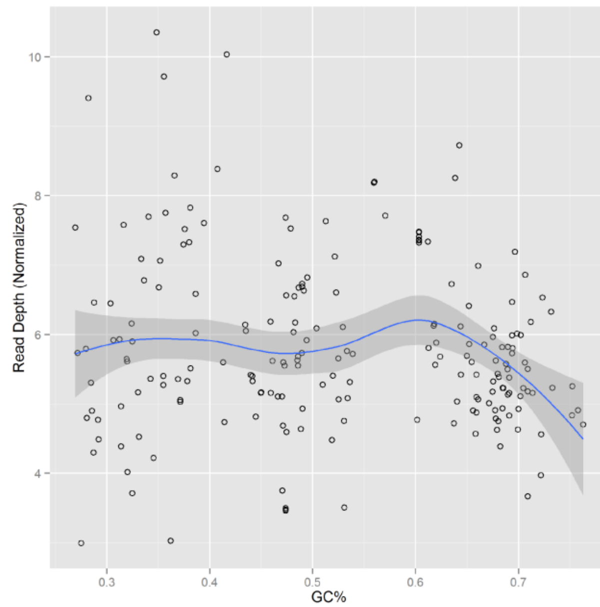

Figure 1.2: Normalized read depth in relation to GC content for 201 genes across all three organisms.

and read depth. The results of this test showed significant correlation between GC content and read depth with a p-value of 0.01436, which is consistent with previous research. A closer examination of the data shown in figure 2 shows that regions with low GC content, less than 40%, is not correlated with read depth. However, as GC content increases, an increasingly strong relation between read depth and GC content can be identified. For regions with medium GC content, between 35% and 60%, a modest correlation between read depth and GC percentage is identified with a p-value of .02722. For regions of high GC content, however, the relationship is much stronger. Regions with greater than 60% GC content showed a strong inverse correlation with read depth with a p-value of 0.000004638. This result is consistent with previous research which has shown that regions with high GC content are more read sparse than medium and low GC content regions (Sendler, et al. 2011).

Next, data for each strain were individually analyzed using linear regression analysis. GC content in Staphylococcus epidermidis, the low GC strain, was not significantly correlated with read depth. However, Exiguobacterium sp., the medium GC strain, showed significant correlation between read depth and GC content with a p-value of 0.03323. Micrococcus luteus, the high GC strain, showed the strongest relationship between GC and read depth with a p-value of 0.0000130. Again, we observe that the relationship between GC content and read depth strengthens as the GC content of a strain increases. The discrepancy between the significance levels of these tests is also to be expected, as the magnitude of GC bias correspondingly varies with the composition of each organism. Staphylococcus epidermidis has an overall GC content of 32% genome-wide, which should not be strongly influenced by GC bias. The lack of a significant

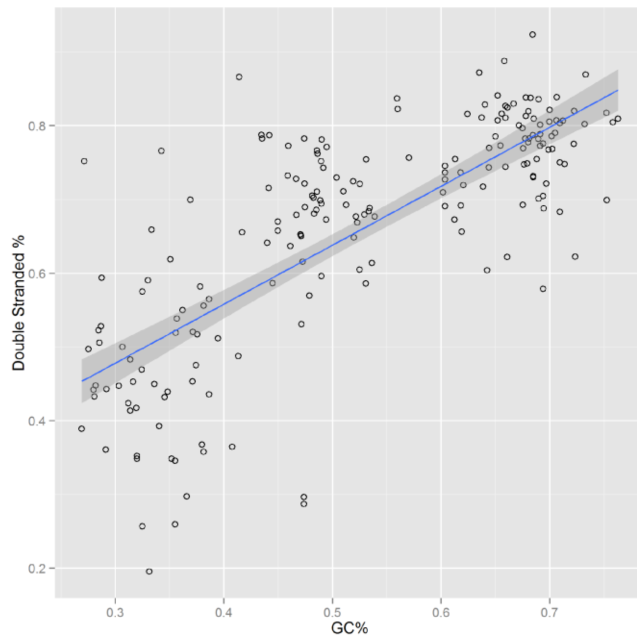

Figure 1.3: Percent of double-stranded predictions by PARS in relation to GC content for 201 genes across all three organisms.

relationship between GC content and read depth in this organism shows that without the influence of GC bias that reads are more evenly distributed across a wider range of depth. Micrococcus luteus, having an average GC level of 48.5%, has a modest, though significant correlation between read depth and GC content. This level of correlation again shows the relationship between GC content and read depth, as some regions are beginning to be influenced by GC bias. In the case of this organism, the level of GC content is just high enough that GC bias becomes apparent, though low enough that the effect is not excessive. Exiguobacterium sp., with a 73% GC content level shows the strongest relation. Here, the effects of GC bias are apparent with a strongly significant correlation indicated. The reads are sparser and with increasing GC the read depth is correspondingly lower. The combination of these results confirms the presence of GC bias in our data and provides a basis for understanding the strength of the effect.

Finally, correlation testing was performed to compare the percentage of positions predicted to be in a folded conformation with the percentage of GC content for each respective gene. The result of this analysis, shown in figure 1.3, was highly significant, with a correlation of 74.1% and a p-value of less than 2.2e-16. This strong relationship between read depth and GC content, in combination with the previously shown capacity of read depth as a predictor of secondary structure conformation, indicate RNA secondary structure as a strong contributing factor to GC bias.

1.3.3 Secondary Structure and Read Start Bias

Next, we investigated the hypothesis that bias is introduced due to secondary structure formation at fragment ends. To do this, binomial logistic regression was chosen to model the relationship between experimentally predicted secondary structure and the number of reads starting at corresponding positions in the control data. Binomial logistic regression is used to model dichotomous output variables as a function of predictor variables, and makes it possible to measure if a predictor variable affects an outcome variable and to what extent (Long, et al. 1997). The outcome variable is this case is binary: a position is in a secondary structure conformation or a position is in a single stranded conformation. The predictor variable is the number of reads starting at the same position as measured by the control data. In this way, we are able to ask not only if there is a significant relationship between levels of reads produced and the presence of secondary structure, but to what extent it is expected that secondary structure exists at a position based on the number of reads starting at that position.

Secondary structure conformational predictions were first calculated using PARS scores representing folded states or single-stranded states for each position of each gene. In this step, positions identified by PARS scores as being in secondary structure conformations were assigned a value of 1, and positions predicted to be single stranded were assigned a value of 0. Logistic regression was then performed, wherein the logistic regression coefficients calculated represented a change in the log odds for a one unit increase in the predictor variable. In this case, for an increase of one in our predictor variable, the number of reads starting at this position in the control experiment, the log

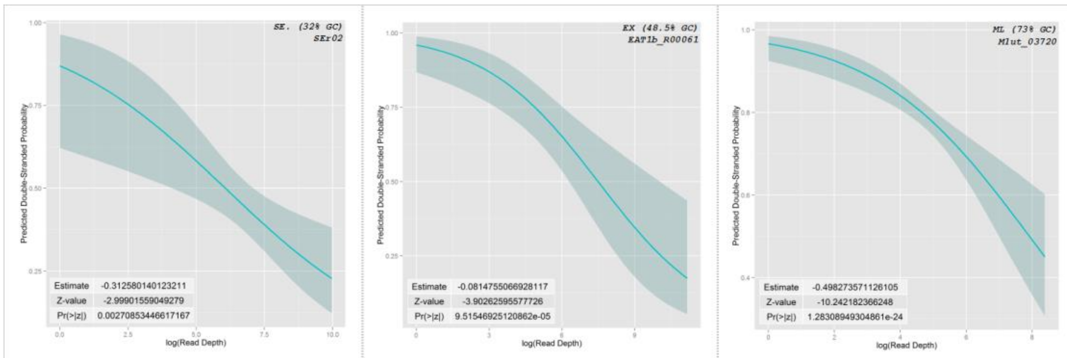

Figure 1.4: PARS predicted probabilities of secondary structure with regard to read starts in 23S rRNA genes for each organism.

odds of that position being in a secondary structure conformation, increased by a factor equal to computed logistic regression coefficient. Using these logistic regression coefficients, confidence intervals and predicted probabilities were calculated across the range of possible values of for read starts. This made it possible to model the relationship between read starts and RNA secondary structure at a nucleotide resolution.

Figure 1.4 shows these predictions for the 23S rRNA genes of each organism. This result showed a clear pattern for all three organisms, in which the predicted probability of a position being in secondary structure decreased as the number of reads starting at that position increased. This result indicates that positions that are known to fold into secondary structure conformations typically have fewer associated read starts.

1.3.4 Summary of Results

A summary of the analyses performed for each of the 207 genes that passed the quality criterion previously discussed and a summary of the results can be seen in figure 1.5. The PARS/Control subheading (pink), shows the distribution of p-values for correlation testing between experimentally predicted secondary structure by PARS and read depth in the control experiment. The PARS/Vienna subheading (green) shows the

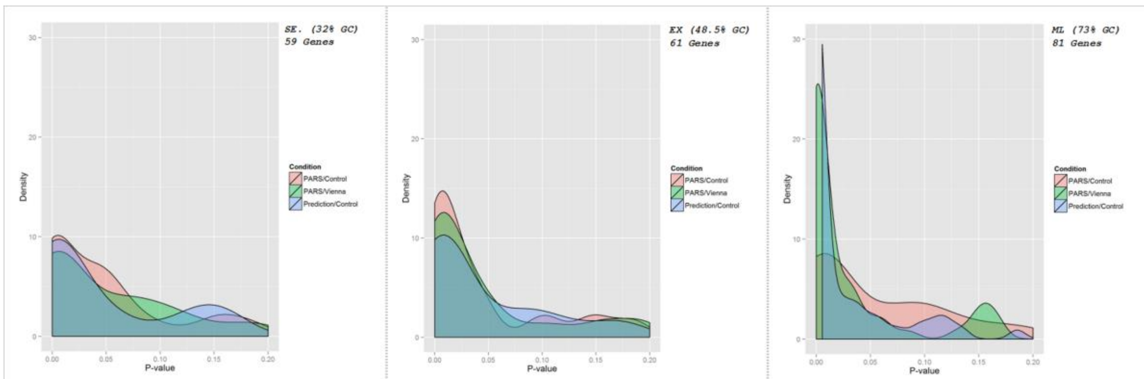

Figure 1.5: Distribution of correlation and regression testing results for all genes passing quality threshold. Read depth and experimentally predicted PARS score correlation is represented as PARS/Control (red), PARS/Vienna shows correlation results from computationally modelled secondary structure predictions and experimentally computed structures (green), and Prediction/Control shows correlation significance based on Logistic Regression (blue).

results of correlation testing between experimentally measured secondary structure and free energy secondary structure predictions as calculated by the software package, Vienna2 (Lorenz, et al. 2011). The prediction/control subheading (blue) indicates the distribution of p-values from the binomial logistic regression analysis. In all cases, a majority of the genes studied showed significance across all three tests, with the significance becoming increasingly apparent with higher levels of GC content.

1.4 Discussion

GC content bias in has been shown to exist predominately in the stages before sequencing, specifically at the PCR stage (Aird, et al. 2011). However, the cause of GC bias exists as a combination of factors. It has been shown that polymerase pauses at the base of, or slightly before, secondary structure formation (Hillebrand, et al. 1985). Furthermore, changes in RNA secondary structures that occur after the structure is only partially unwound lead to the formation of new secondary structural elements downstream from the first, which leads to slower progress overall by polymerase (Suo, et al. 1997). Other research has shown that sequences leading to stable secondary structure at fragment ends can cause adapter ligation to fail, leading to the failure of reverse transcription and introducing bias between the relative expression levels of fragments with secondary structure and those without (Tian, et al. 2010). The combination of these behaviors lead to a reduction in amplification efficiency, particularly in PCR based systems where relatively small differences before amplification may be represented exponentially afterwards.

Efforts to correct for GC bias involving methods of eliminating secondary structure formation have proven to be effective. Betaine is a zwitteronic osmoprotectant that alters DNA stability in such a way that stable secondary structure in GC rich regions melt at temperatures similar to those required to melt AT rich regions, and the introduction of betaine to PCR assays has been shown to suppress replication pausing by polymerase (Schwartz, et al. 2009). It has also been observed that increasing denaturing time in combination with the introduction of betaine substantially increases read coverage and depth in GC rich regions (Aird, et al. 2011). These findings, in conjunction with the findings of this study, indicate that the formation of RNA secondary structure is the primary cause of GC bias.

Although reduction in levels of secondary structure addresses a substantial portion of GC bias, it does not eliminate bias in sequencing assays completely. The use of betaine, for example, while reducing secondary structure in GC rich regions, may cause early disassociation of the newly synthesized strand from AT rich templates (Aird, et al. 2011). In one study, PCR-free FRT-seq was used, in which reverse transcription occurred directly on the flow cell after adapter ligation (Mamanova, et al. 2010). Comparison of optimized PCR assays with the PCR-free experiment showed that the optimized assay performed nearly as well as eliminating the PCR phase altogether (Aird, et al. 2011). Both assays, however, still produce fewer reads in GC rich regions, indicating that correction methods are yet imperfect or that other processes contribute to GC bias in ways that are not well understood.

In this paper, we have generated a novel dataset that provides experimentally measured RNA secondary structure predictions at a nucleotide resolution and have used that data to examine the role of RNA secondary structure on GC bias as observed in modern sequencing technologies. We have demonstrated that our dataset is consistent with previous research findings, and that the relationship between GC content and read depth in three bacterial strains spans a wide range of GC content. We have described the relationships between RNA secondary structure and GC content and between RNA secondary structure and read depth. We have shown that fragment counts are significantly biased, with a lower frequency of read starts at sites which are folded into RNA secondary structure conformations. Finally, we have shown that the relationship between read depth and GC content causes increasing bias as GC content increases, with less significant biases being caused in lower GC sequences.

One possible limitation of this method is that V1 nuclease can also cleave at stacked nucleotides formed due to intramolecular interaction at positions that are not double stranded (Novikova, et al. 2012). However, investigation into the possible bias of V1 in the PARS protocol has shown that there is a very small bias towards particular regions along the transcript. However, it has been confirmed that signals generated by RNase V1 are highly distinct from those generated by RNase S1 and global inspection across all transcripts for the PARS protocol revealed that approximately 7 percent of V1 and S1 peaks are shared. These shared peaks could be the result of experimental noise introduced by nonspecific enzymatic activity, but could also correspond to dynamic RNA regions or transcripts that fold into more than one stable conformation (Kertesz, et al. 2010). We therefore believe that this is an acceptable limitation of the PARS method and by extension of this study and several others (Kertesz, et al. 2010; Wan, et al. 2013; Wan, et al. 2016).

The findings of our study point to optimization of assays with regard to RNA secondary structure as an ideal method of reducing GC bias. The identification and modeling of read fragment bias at positions predicted to be in secondary structure conformations is particularly novel and opens new avenues of research for the correction of GC bias. Future work in this area may include the computational modeling of the relationship between read start sites and RNA secondary structure, as well as the application of corrections to read counts based on adjustments for observed start site bias.

CHAPTER 2: READ MAPPING TO NON-NATIVE REFERENCE GENOMES IN RELATED BACTERIAL STRAINS

2.1 Background

Sequence read alignment to a reference genome is currently a key step in many common bioinformatics workflows. Accuracy of alignment is therefore crucial for proper interpretation of biological data. Researchers frequently encounter situations in which the most appropriate reference genome for a reference-based analysis is not available, and a homologous alternative must be used. This can lead to inaccuracies in mapping and subsequently in quantitation and interpretation. These inaccuracies skew the results of otherwise sound analysis methodologies. This study approaches the problem of non-native reference alignment by comparing the effects of read alignment to native and heterologous reference genomes. We describe a method to identify false positives caused by improper alignments to the heterologous reference, and examine the underlying causes, to provide a set of best practices for research that makes use of non-native reference genomes.

Comparative analysis of microbial genomes since the advent of high-throughput sequencing has shown that prokaryotic genomes are dynamic and can be highly diverse, even among closely related species or strains. Analysis of bacterial genomes through sequencing-based methods such as RNA-Seq has made it possible to rapidly advance our understanding of basic biological function, identify host-pathogen interactions, and engineer microbes for industrial and pharmaceutical applications (Fraser-Liggett, 2005).

It has become apparent in recent years that the highly dynamic nature of prokaryotes has led to extensive genomic diversity. In 2001, sequencing of E. coli O157:H7 identified over 1300 strain-specific genes when compared with E. coli K-12, the strain previously sequenced and thought at the time to be fairly representative of the model organism (Perna, et al. 2001). The identification of these genes, found to be involved primarily in virulence and metabolism, showed that even closely related strains can differ significantly. Since that time, the availability of sequencing data from multiple strains of the same organism has increased, but due to the vastness of biodiversity in prokaryotes, it is still not uncommon to find that the most appropriate reference genome is not available, or exists only as a draft. Researchers then must resort to using finished evolutionary neighbor reference genomes in their studies, even when the sequence reads they wish to map were produced from a heterologous strain.

Many common analysis pipelines rely on accurate alignment of reads to a corresponding reference genome. Differential expression studies, for example, rely on aligning transcriptome reads to a reference, extracting count data, and examining the differences in transcript read levels for the genome under study. In cases where two or more closely related species or strains are being studied, a common approach is to simply map reads from all organisms to a common reference genome. The assumption is that the differences between closely related microbes are insignificant enough that the results of differential expression will not be influenced, or otherwise, that genes that are absent in one sample or the other should simply be excluded from analysis, while shared genes that have seemingly reasonable read counts in both organisms can be used for differential expression analysis. For example, we previously investigated differences in gene expression in clinical strains of Vibrio vulnificus, when exposed to either artificial seawater or human serum environments, as a model for the expression changes the organism undergoes when infecting a human host. V. vulnificus CMCP6 and V. vulnificus YJ016 expression levels were compared by using the CMCP6 strain as a common reference genome (Williams, et al. 2014). Using a common reference genome to make comparisons between different strains is also common in eukaryotic systems, and similar methods were used in a comparative study of strains of Bombyx mori (Bao, et al. 2009). The approach of using a common reference genome for different strains is unable to correctly represent factors that can influence read counts in the non-homologous read set, such as the frequency and density of mismatches due to natural divergence between strains. The degree of error in these studies will be affected by how alignment algorithms handle reads with multiple possible mapping positions, especially when mutations decrease mapping position certainty. Comparing data across strains becomes increasingly less sound as evolutionary distance between read sets and the reference genome increases, and this is particularly true of prokaryotic species, where divergence occurs at an accelerated pace. In this study, we examine the potential impact of using a heterologous reference genome and the effects on read alignment, and by extension differential expression. We show how differences in reference genomes influence read alignment and gene expression results when using common analysis techniques. We then provide an approach for identifying false positives caused when comparing multiple strains or species by means of alignment to a common reference genome, and outline best practices for the use of heterologous reference genomes in cross-strain analyses.

2.2 Materials and Methods

2.2.1 Data and Heterologous Reference Distance

For this study, transcriptome data from two different organisms were used. RNA-Seq data for two strains of Vibrio vulnificus, CMCP6 and YJ016, as described by Williams et al. were used. A publicly available data set, consisting of RNA-Seq data from Escherichia coli strains K12 (MG1655), a common laboratory strain, and strain IAI1, a commensal modal strain, under three experimental conditions (Vital, et al. 2015) was also analyzed. The V. vulnificus data set consists of two experimental conditions, human serum and artificial seawater, each having two replicates, while the E. Coli data set consists of three experimental conditions, batch, chemostat, and starvation, with each condition having two replicates as well. A summary of the data used in this study can be seen in table 2.1. In all cases sequencing was performed using the Illumina HiSeq platform. The same analysis workflow was applied to each data set.

Table 2.1: Summary of read data

Species	Strain	Condition	Replicates	Coverage (Native)	Coverage (Heterologous)
V. vulnificus	CMCP6	Human Serum (HS)	2	628.5 / 671.2	678.9 / 522.9
V. vulnificus	CMCP6	Artificial Saltwater (ASW)	2	641.5 / 550.5	700.9 / 349.7
V. vulnificus	YJ016	Human Serum (HS)	2	715.8 / 544.4	595.3 / 611.8
V. vulnificus	YJ016	Artificial Saltwater (ASW)	2	709.3 / 350.1	616.1 / 531.2
E. coli	K12 (MG1655)	Batch	2	169.2 / 206.6	175.7 / 214.6
E. coli	K12 (MG1655)	Chemostat	2	194.4 / 136.8	201.9 / 142.4
E. coli	K12 (MG1655)	Starvation	2	108.6 / 107.4	115.3 / 113.9
E. coli	IAI1	Batch	2	145.9 / 121.9	154.4 / 128.4
E. coli	IAI1	Chemostat	2	135.1 / 130.9	142.9 / 138.5
E. coli	IAI1	Starvation	2	98.9 / 95.5	104.4 / 101.1

Initial comparisons were performed between reference genomes for both bacterial strains using Mauve (Darling, et al. 2004). The E. coli K12 strain is 4,641,652 base pairs in length and the IAI1 strain is 4,700,560 long, for a difference in length of 58,908 base pairs. 64,577 SNPs were identified across the span of both strains, making the approximate level of polymorphic sites 1.38%. The combination of indel and polymorphic differences between these genomes is 2.64%. Structural analysis shows

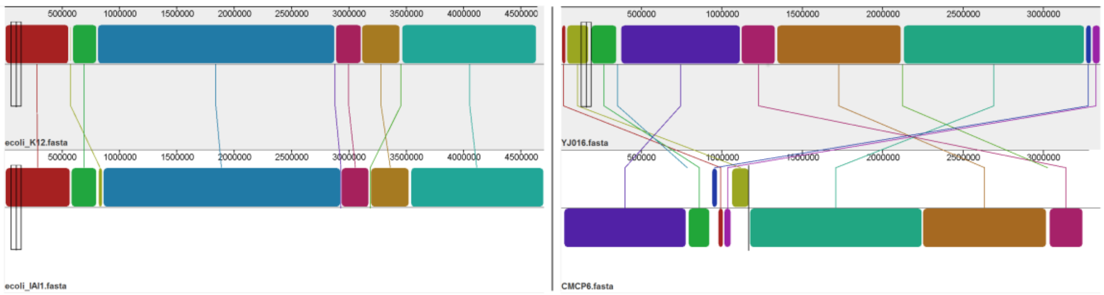

Figure 2.1: Structural differences between reference strains for E. coli (left) and V. Vulnificus (right).

relatively few rearrangement events and broad similarity between these genomes. V. Vulnificus CMCP6 and YJ016 are 3,281,866 and 3,354,505 base pairs in length respectively, for a difference in total length of 72,639 base pairs. 46,955 SNPs were identified between the two references, making the difference between the two genomes by polymorphic sites 1.42%. Combining the total differences for polymorphisms and indel events, these strains are approximately 3.61% different from one another. Structural analysis of the V. Vulnificus genomes revealed more structural changes than were observed in E. coli, although overall structural similarity is still high. Figure 2.1 shows structural differences for both organisms.

2.2.2 Orthology Mapping and Data Processing

In order to make accurate comparisons of data as aligned to heterologous reference genomes, orthology relationships between genes were first determined. All-against-all protein BLAST was used to find orthologous regions between strains.

Regions were determined to be orthologous if they showed greater than 95% identity, were at least 200 base pairs in length, and had no more than 5 mismatches at the protein level. 1570 orthologous regions, approximately 36% of annotated genes, were identified between V. vulnificus strains CMCP6 and YJ016. 2378 orthologous regions, approximately 55% of genes, were identified between E. coli strains K12 and IAI1. Annotation information for each strain was then applied to these orthologous regions, to determine correspondence in read counts between strains at a per-gene level.

Each RNA-Seq read set was aligned to both potential reference genomes for their respective species using Bowtie2 (Langmead, et al. 2012). In the case of E. coli, all replicates and conditions from both strains K12 and IAI1 were aligned to both the K12 and IAI1 reference genomes. Similarly, all read sets for V. vulnificus were aligned to both the CMCP6 and YJ016 reference genomes. All alignments were performed using Bowtie2's sensitive alignment (-M 3, -N 0, -L 22). Raw read counts were then extracted from each alignment using the featureCounts Bioconductor package (Liao, et al. 2013). Next, the previously computed orthologous gene mapping information was used to map read count data for all conditions and replicates with orthologous genes. This process was performed on all samples and replicates for both E. coli and V. vulnificus, so that each read set is counted for alignment to both their native reference genome and the heterologous reference genome for their respective species. This makes it possible to make direct comparisons of the effects of mapping identical read data to heterologous genomes. As the process was applied for all conditions to both native and heterologous alignments, cross effects can be identified to increase confidence in observations. An overview of this data processing pipeline is shown in figure 2.2.

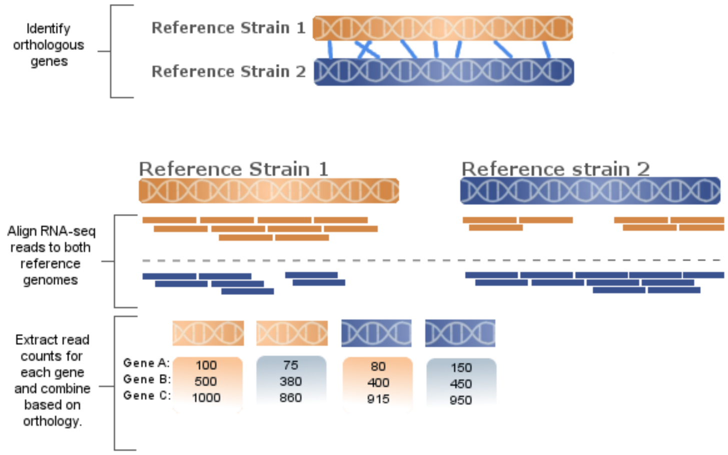

Figure 2.2: Data processing pipeline. Orthology is identified between heterologous strains and reads are aligned to both reference genomes. Using the orthology mapping information, extrapolated read alignment counts are compiled such that counts can be compared for each read set as aligned to each reference genome.

Next, differential expression analysis was performed for all strain/condition permutations for each organism. By examining the results of differential expression analysis on identical read data, with the choice of reference genome being the only differential factor, any genes that are identified as being differentially expressed for the same condition can be marked as false positives caused by reference-based factors. For example, by aligning reads from E. coli strain K12, batch condition, to both the K12 reference genome and the heterologous IAI1 reference genome, and then performing differential expression analysis, any genes that are identified as being differentially expressed can be assumed to have been incorrectly identified, as the initial read set is identical and the only differential factor is the reference genome. When examining the reciprocal condition, in which reads generated from the IAI1 strain, batch condition, are aligned to both reference genomes, another set of false positives can be identified, many of which correspond to the false positives identified previously, creating a cross-identification effect for many false positives.

Differential expression analysis was performed using DESeq2 (Love, et al. 2014). Initial investigation of each experimental condition aligned to native and heterologous reference genomes showed high similarity in all cases. Principal component analysis was performed to confirm the integrity of replicates for all read sets. Figure 2.3 shows an example of the log-fold changes for all genes for the E. coli, strain K12, batch condition. In this case, reads generated from strain K12, batch condition, were aligned to both the native reference and the heterologous reference, strain IAI1. As the reads are identical, high correspondence is naturally expected, with variation only being caused due to differences in the reference genome. High correspondence such as this was observed for all conditions and read sets. This level of correspondence indicates that the assumption that these two genomes can be used interchangeably as a reference for read alignment is reasonable. Had this data shown significant deviation, it would have indicated that heterologous alignment was not appropriate. This comparison of alignment to orthologous regions should be applied in cases when a reference genome for a read set is unavailable, but a homologous alternative

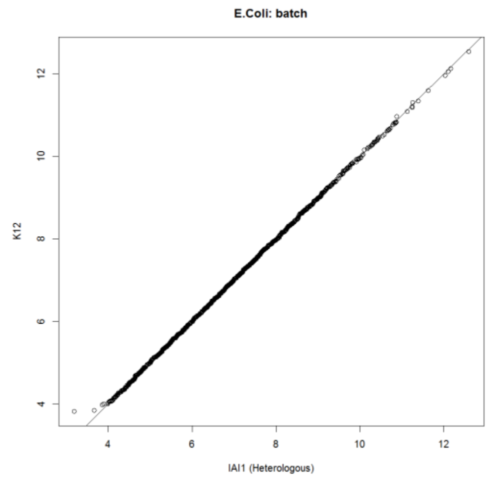

Figure 2.3: Log-fold changes of read counts for all E. coli strain K12 genes as aligned to both native and heterologous references.

exists, in order to determine if alignment to the homologous reference if viable. This point will be covered in additional detail in the discussion section of this study.

2.3 Results

2.3.1 False Positive Identification

Differential expression analysis was performed on all permutations of data sets for each organism as aligned to native and heterologous reference genomes. False positives were identified in two ways. When identical datasets were aligned to both references for a single condition, differential expression analysis was performed and the set of differentially expressed genes were taken as false positives. For example, E. coli strain K12, batch condition was aligned to both native and heterologous genomes and differential expression was performed, identifying 15 false positives. When identifying false positives across multiple conditions, differential expression for two conditions is performed with both conditions aligned to native and heterologous reference genomes, and false positives are then identified as the set difference between the two differential gene results. For example, when comparing E. coli strain K12, batch condition to the K12 chemostat condition, differential expression is performed on the batch vs chemostat reads as aligned to the K12 genome, and then as aligned to the IAI1 genome. True positives are considered to be the intersection of these two result sets, and false positives are considered to be the difference of the two sets. This method generally identifies significantly more false positives than are identified when only a single condition is examined. This compounding of false positives is to be expected as the first method relies on aligning only one read set to two references (2 replicates x 2 alignments each), and the second method must align two read sets to two references (2 replicates x 4 alignments each). Table 2.2 shows a summary of all false positives identified through both methods.

Table 2.2: Summary of false positives

Species	Native Reference	Condition	False Positives	Cross-identified
V. vulnificus	CMCP6	Human Serum (HS)	0	0
V. vulnificus	CMCP6	Artificial Saltwater (ASW)	2	1
V. vulnificus	YJ016	Human Serum (HS)	1	0
V. vulnificus	YJ016	Artificial Saltwater (ASW)	2	1
V. vulnificus	CMCP6	HS vs ASW	14	1
V. vulnificus	YJ016	HS vs ASW	14	1
E. coli	K12 (MG1655)	Batch	15	5
E. coli	K12 (MG1655)	Chemostat	6	4
E. coli	K12 (MG1655)	Starvation	1	1
E. coli	IAI1	Batch	9	5
E. coli	IAI1	Chemostat	16	4
E. coli	IAI1	Starvation	5	1
E. coli	K12 (MG1655)	Batch vs Chemostat	58	6
E. coli	K12 (MG1655)	Batch vs Starvation	17	0
E. coli	K12 (MG1655)	Chemostat vs Starvation	32	2
E. coli	IAI1	Batch vs Chemostat	61	6
E. coli	IAI1	Batch vs Starvation	40	0
E. coli	IAI1	Chemostat vs Starvation	42	2

Cross identification of false positives was also examined to determine if the same regions produce false positives across different read sets. Several false positives were identified from multiple read sets; however, cross identification is not necessarily always present due to naturally occurring differences in expression levels between different strains. Even though a reduction in read counts is typically associated with alignment to a heterologous genome, genes will not necessarily be identified as differentially expressed unless the log-fold change is significantly different with regard to the expected concentration of fragments as determined by dispersion of counts across the entire read set (Love, et al. 2014). For example, if an E. coli read set from the K12 strain is aligned to both a native and a heterologous genome, and a gene is identified as a false positive through differential expression, we can be confident that read alignment for that gene is being compromised by the reference genome. While it is likely that the same gene will be reciprocally compromised in the corresponding read set from the IAI1 strain, it may or may not be identified as differentially expressed because the overall expression levels in IAI1 may be naturally different from K12, and the log-fold change may not be extreme enough to identify the gene as differentially expressed with regard to fragment dispersion for the entire IAI1 read set. For this reason, genes are considered to be false positives if they are identified in either case, though special attention was given to genes with cross identification as representative cases of false positive causes in later analysis.

Read counts for false positives tended to be significantly higher when aligned to their native genome than their heterologous counterpart. This is to be expected, as it is likely that differences in genome cause alignment failures for non-native reads. Once false positives were identified, sequence analysis was performed. Nucleotide BLAST was performed on all ortholog pairs to examine the influence of reference sequences on read alignment. For E. coli the mean number of polymorphic sites per gene was 13 between the two reference genomes. Similarly, V. vulnificus strains showed a mean of 12 SNPs per ortholog pair. In all cases, false positives contained two to three-fold increases in SNPs, with E. coli having an average of 28 SNPs per false positive and V. vulnificus having 26. This ratio was also observed with regard to gene length, with the number of SNPs to length ratio for false positives being around three times that of true positives.

Once false positives were identified through differential expression analysis, further examination of the identified genes was conducted to identify the underlying causes for false positive identification. Two specific causes are identified as the primary contributing factors: indel/duplication events and high-density SNP windows.

2.3.2 Indel / Duplication Events

One gene that was identified as a false positive of particular interest in E. coli was cusC. This gene was cross-identified for both batch and chemostat conditions for read sets generated from both the K12 and IAI1 strains. For this reason, this gene was selected as a representative sample for the explanation of duplication based false positive identification. cusC is the first gene of an operon consisting of 4 genes. Figure 2.4 shows an overview of the operon structure (Castro-Gama, et al. 2016).

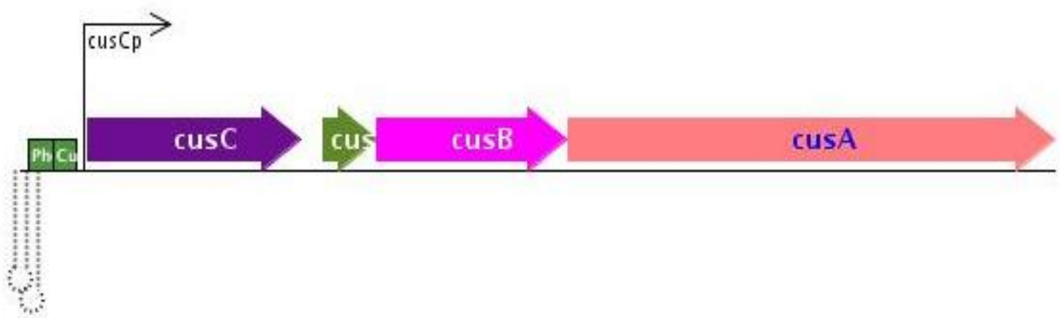

Figure 2.4: Operon structure containing the gene. cusC -> cusF -> cusB -> cusA

The cusC operon encodes a two component signal transduction system that is responsive to copper ions, acting as a regulatory system to the pco operon, which provides copper resistance for E. coli (Munson, et al. 2000). The cusC gene itself is 1373 bases long and has 21 SNPs along its length between the K12 and IAI1 strains. Overall expression for this gene is generally low relative to average expression levels for each genome, and the ratio of SNPs to gene length is also approximately half that of typically identified false positives. Other genes in the operon following cusC are not identified as false positives.

Investigation into the cause of false positive identification of cusC found that incorrect expression levels are created due to an indel event between the two genomes. Figure 2.5 shows an example of read alignment to this operon for an identical read set as aligned to both native and heterologous genomes.

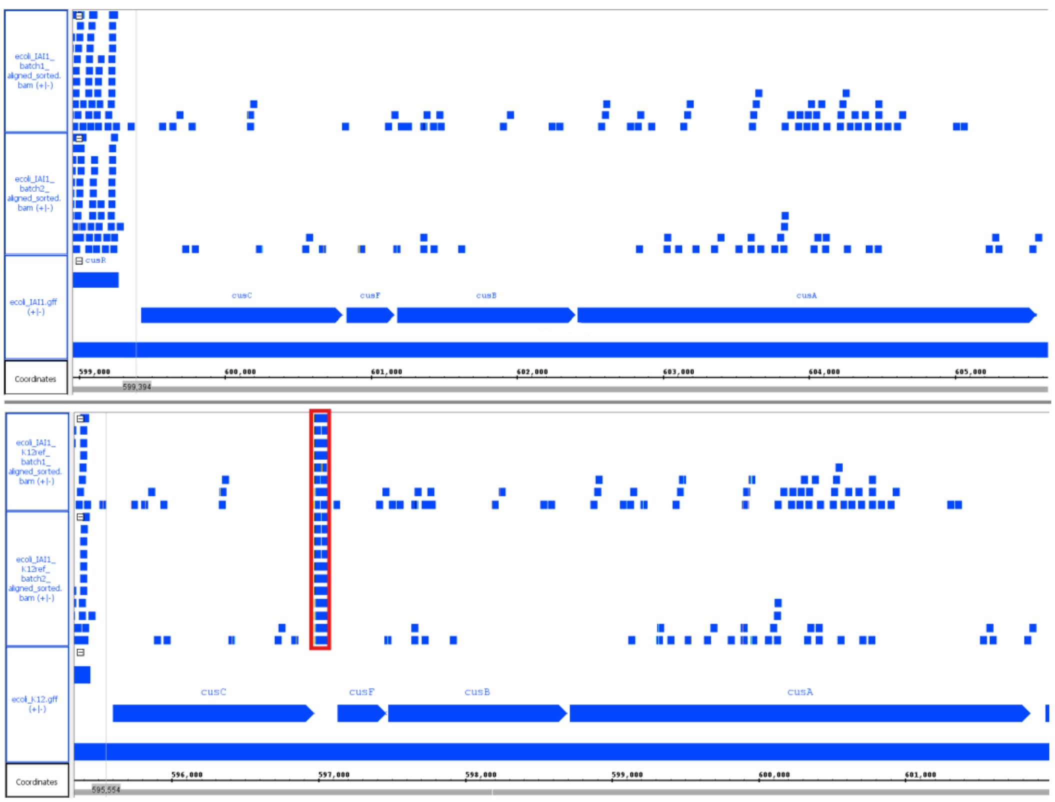

Figure 2.5: Read alignment for cusC operon for IAI1 batch condition, replicates 1 and 2. The native IAI1 genome (top) shows a continuous operon with sparsely aligned reads, while the heterologous K12 genome (bottom) shows an insertion after the cusC gene that is highly expressed (outlined in red).

The indel event that can be seen between cusC and cusF between the native (IAI1) and heterologous (K12) genomes causes reads that align to the gapped area that slightly overlap cusC to be counted as expression for the cusC gene, causing a log-fold change in expression between the true and false expression levels of approximately 3.4, a highly significant difference. Examination of the overlapping region, as shown in figure 2.6, shows that the reads map poorly to this region, especially in the area overlapping the cusC gene. In addition, the reads in this example were generated from the IAI1 strain and therefore cannot have produced reads in these positions, which implies that these genes are most likely the result of a duplication event and have been mapped to multiple locations. This suggests that elsewhere in the genome a region where these reads map accurately should exist. To investigate this possibility, BLAST was performed on the sequence from the indel point and found five matching positions in the K12 genome and only four in the IAI1 genome.

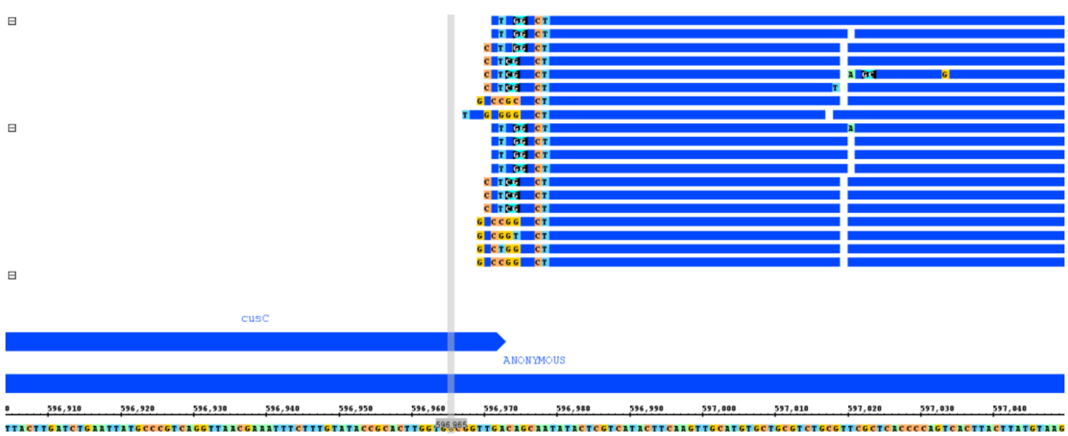

Figure 2.6: Reads in the indel region slightly overlapping cusC, with particularly poor alignment in the overlapping area.

Each of these positions were individually inspected in both genomes and appear to be orthologous across references, with the additional matching region in the K12 strain being the observed location in cusC. In both genomes, a single case was found where these reads map perfectly, with all other cases showing similarly poor alignments to that shown in figure 2.6. It is likely that a duplication occurred in E. coli, in which sequence from this section of genome, showing perfect alignment for these reads, was inserted into other parts of the genome. This specific sequence duplication has occurred in K12 in the cusC operon, but did not occur in the IAI1 strain. When alignment is performed using either the IAI1 or the K12 based reads, because the original sequence that was duplicated still exists elsewhere in both genomes and is expressed, reads from that region incorrectly map to the duplicated region in one genome and not the other, causing a false positive. Interestingly, one of the positions identified as a duplicated region for this same sequence corresponded to another gene, yhbI, which was cross identified as a false positive in all E. coli conditions and read sets. This gene shows the exact same expression profile, with a highly-expressed region of poorly mapped reads aligning near the end of the gene.

This type of improper alignment is due to how bowtie2 handles reads that map to multiple locations. When multiple sites are identified for possible alignment by bowtie2, reads can be mapped to both positions. For this reason, false positives are identified at points where small duplications have occurred within the genome and minimal divergence has occurred at the duplicated points. One possible solution to this might be to consider only uniquely mapped reads, however this would have the effect of removing all reads that map to multiple locations from all possible mapping positions, which would bias the data for the actual mapping position from the opposite direction, removing the false positive identification for cusC and yhbI, but changing the expression levels of the gene where the reads were truly expressed. This is discussed in further detail later in this study.

2.3.3 SNPs

The other primary cause of false positive identification between native and heterologous genomes was read loss caused by SNPs in highly concentrated windows. A majority of false positives identified for all conditions showed significantly higher proportions of polymorphic sites for false positives on average as compared to the mean level of polymorphic sites between genes for the genomes as a whole. False positives

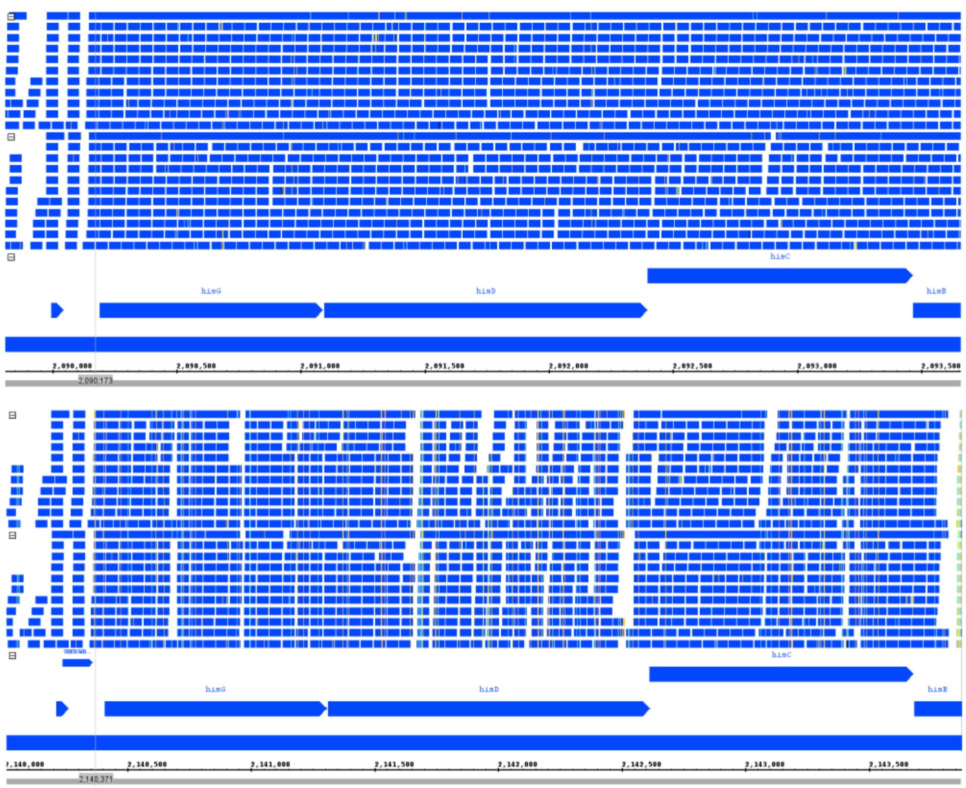

Figure 2.7: hisD reads aligned to native genome (top) and heterologous genome (bottom).

identified due to read alignment loss due to SNPs showed a two to three fold increase in propensity of SNPs with regard to their length, while genes identified as being false positives due to indel/duplication events showed sequence correspondence more similar to average expected levels of difference.

As a representative gene for false positives due to read alignment loss by SNPs, hisD, a gene which codes for histidinol dehydrogenase, was chosen. This gene was selected because it showed very uniform coverage in the native reference genome and because SNPs were distributed widely in various concentrations across the length of the heterologous reference, which makes read loss more visually apparent. Figure 2.7 shows read alignment for hisD from the E.coli batch condition, with the K12 strain being the native reference and the IAI1 strain being the heterologous reference. This gene has 55 SNPs between the native and heterologous genome across a length of 1305 bases. Read loss can be observed particularly in regions of high SNP density, where read alignment becomes increasingly more difficult due to differences in the reference sequences. The two flanking genes, hisG and hisC also show some moderate read loss, but these genes are not identified as false positives because the read loss is less severe and doesn't cause a significant enough log-fold change to trigger differential expression flagging. Other genes identified as false positives show similar read loss when windows of high-density SNPs are present, with some cases having very distinct windows of loss and otherwise similar coverage between genomes, and still others showing staggered read loss across the gene, as was shown here in the case of hisD.

The type of read loss observed between native and heterologous genomes due to SNPs might be reduced by either relaxing read alignment parameters so that reads can be aligned when higher levels of SNPs are present, or otherwise by through the application of sequencing technologies that produce longer reads. One potential problem of approaching this issue by adjusting alignment parameters is that reads may incorrectly map with higher propensity to incorrect regions, further biasing the read set. This problem would be further compounded if only uniquely mapped reads were used, as the proportion of reads that map to multiple locations would necessarily increase as alignment parameters become less restrictive.

2.4 Simulations

2.4.1 Read Length

A majority of the false positives identified were caused by read loss in regions with high levels of SNPs. In order to examine if this effect can be mitigated by read length, several simulations were performed. Using Simulome, a reference genome was simulated based on the E. coli K12 strain (Price, 2017). The simulated reference genome contained 500 genes, with lengths selected in a normal distribution around the mean length of genes for the K12 strain. Each simulated gene was separated from its neighbor by a randomly sized intergenic region. A heterologous version of this reference genome was also simulated, in which each gene

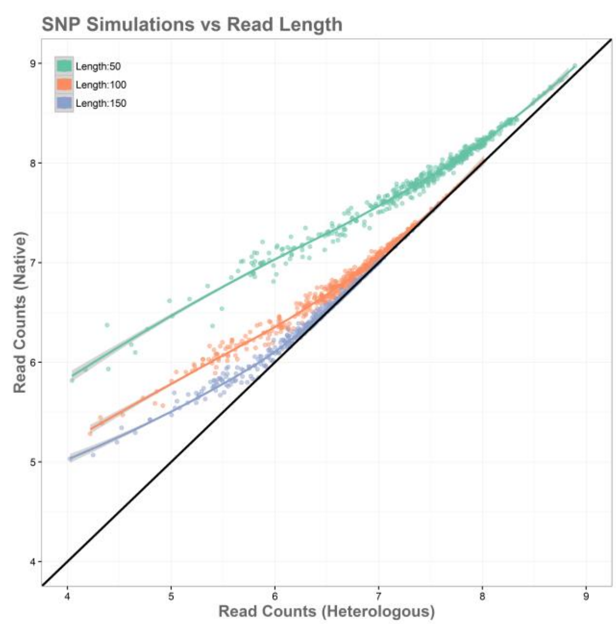

Figure 2.8: Simulation of the relationship between SNPs and read length. Log-fold change in read alignment for native and heterologous genomes.

contained 35 SNPs, approximately the average number of SNPs observed for false positives identified for E. coli that were caused by SNP induced read loss. Read data was then simulated using the ART package (Huang, et al. 2012). Read data was created for the simulated reference genome based on ART's Illumina HiSeq 2500 model, with simulated fold coverage of 150, for read lengths of 50,100, and 150. These parameters were selected to mirror the properties of the actual data for the E. coli data set. The simulated reads were aligned to the simulated reference, which was considered the native reference genome, and also to the mutated reference simulation, in which each gene contained 35 randomly distributed SNPs across the length of each gene. Alignment and read count extraction methods were performed identically to those outlined in the methods section on the actual read data.

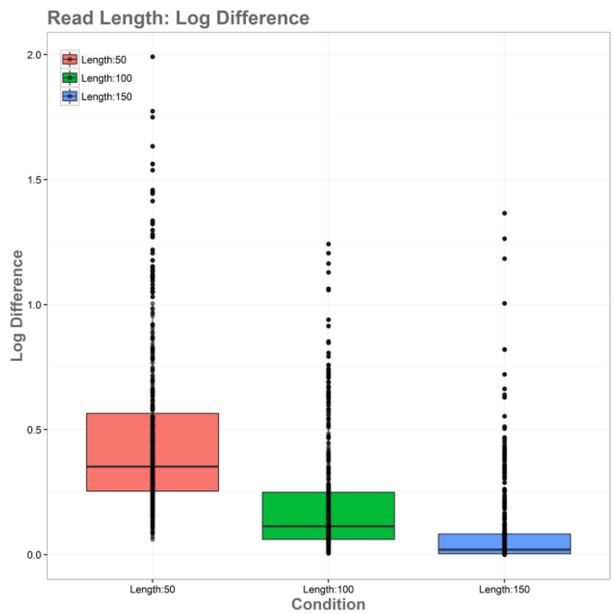

Figure 2.9: Log-fold differences in native vs heterologous alignment for different read lengths.

The simulations showed substantial improvement in accurate alignments between native and heterologous reference genomes as read length increases. This relationship can be seen in figure 2.8. Reads of length 50 performed the most poorly in simulations, with all genes showing read loss when aligned to the heterologous reference genome, and those with lower expression levels showing the greatest log-fold changes. Reads aligned to the heterologous genome with length 50 had 19.77% read loss overall. Reads of length 100 performed significantly better, with log-fold changes in read alignment being substantially closer to expected values and becoming increasingly reliable as expression levels increase. These reads showed a significantly better alignment, with a read loss of 8.33%. Reads of 150 in length showed the best performance among simulations, with higher accuracy for all reads over the entire range of expression levels and complete accuracy being reached at lower expression levels than the 50 and 100 read length simulations. Alignment here was again the best of the simulations, with a read loss of only 3.09%. Figure 2.9 shows an overview of log-fold differences between alignment to native and heterologous genomes for the three simulated read lengths. Overall these simulations show that heterologous reference use is more reliable with longer read lengths, and that the expected number of false positives caused by polymorphisms will be reduced as read length increases.

Figure 2.10: Simulation of the relationship between SNPs and read length shown using bowtie2’s –very-sensitive alignment settings.

2.4.2 Alignment Sensitivity and Read Depth

An additional variation of this simulation was performed using bowtie2's "--very-sensitive" alignment parameter. The use of this argument increased overall alignment in all cases, reducing read loss to 13.65% for the 50 read length simulation, 4.00% for the 100 read length simulation, and to just 1.41% for the 150 read length simulation. This is a modest improvement over the standard alignment parameters and can be seen in figure 2.10 as each curve becoming slightly tighter and approaching accurate read levels across native and heterologous genomes from slightly lower expression levels. The lower range of expression values, however, are not influenced strongly enough for this method to mitigate false positives completely, while it does have value and should be used for native and heterologous alignment issues, the stronger influence appears to come from increases in read length. Next, the effect of read depth was examined. In our sample data, E. coli had an average read coverage of 150x, V. vulnificus had a much greater read depth of around 600x. Simulations of these conditions show that increasing read depth has little influence, simply compressing the range of depth toward the higher end, and maintaining similar ratios of log-fold differences between native and heterologous genomes. The results

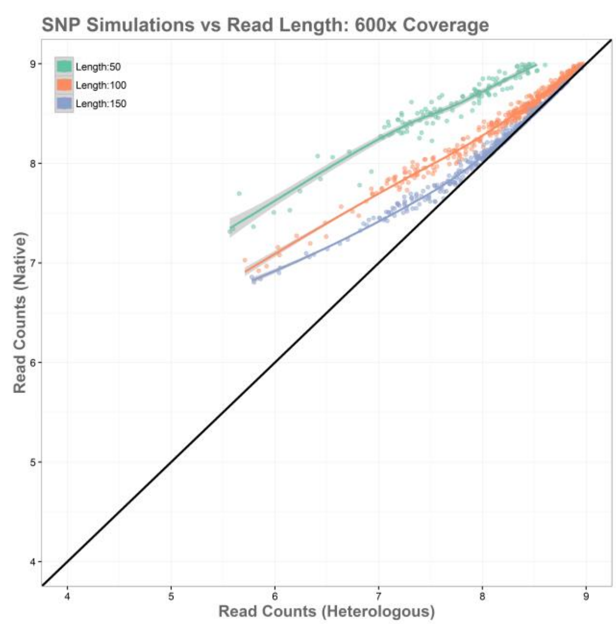

Figure 2.11: Simulation of the relationship between SNPs and read length shown at 600x coverage.

of this simulation can be seen in figure 2.11.

2.4.3 Multiple and Unique Mapping Positions

Some false positives were caused between native and heterologous genomes when insertion events that copied short segments into or adjacent to coding regions were present, causing reads from other regions of the genome to incorrectly map to some genes in the non-native genome. One possible approach to removing these false positives is to consider only reads that uniquely map to a single position in the genome as valid reads. This would mean that reads that map incorrectly would not be included in read counts, but also that those reads would not correctly map to their proper location as well. If the correct mapping location for these reads, however, was orthologous between the native and heterologous genomes, the bias introduced from removing these reads should be roughly the same, with the effect of eliminating false positives while maintaining a true ratio of gene expression for genes containing the correct mapping position.

To simulate the effects of this approach, Simulome was used to create a 500 gene simulated reference genome and a mutated variant with insertion events 100 bases in length, which were copied from other random positions in the original reference. This means that in each gene in the variant genome, an insertion of 100 base pairs exists that also has a correct mapping location elsewhere in the genome. Read data was created using ART for the simulated reference genome based on ART's Illumina HiSeq 2500 model, with simulated fold coverage of 150.

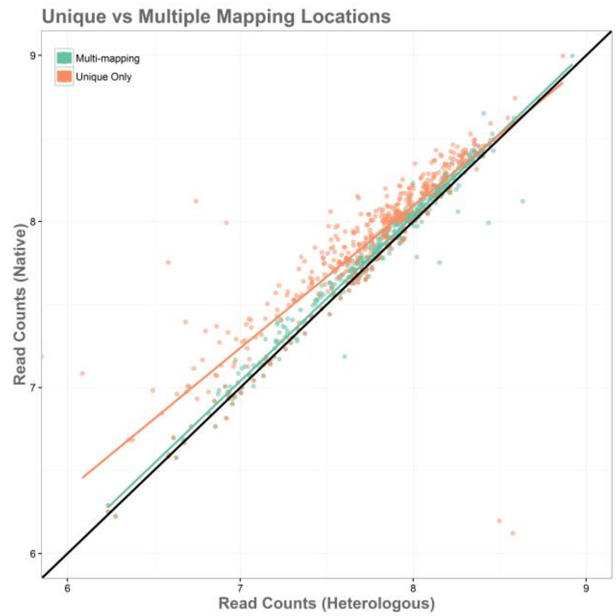

Figure 2.12: Simulation of the reads with ambiguously mapping inserts. Log-fold change in read alignment for native and heterologous genomes.

Figure 2.12 shows the performance of read alignment for the native and heterologous genomes with ambiguously mapping positions included and only uniquely mapped positions. The condition in which multiple mapping locations were included performed much better, with most genes showing appropriate levels of read alignment across the native and heterologous genomes. This scenario did show several genes that would likely be identified as false positives, which can be seen as being more highly expressed in the heterologous genome. These genes were not present as false positives in the case of unique mapping and returned to a more appropriate read alignment ratio between the native and heterologous genomes, but overall the level of read alignment for the heterologous genome is reduced significantly overall, introducing a bias that is far more extreme than the problem it solves.

2.5 Discussion

The use of non-native references genomes relies on the distance between the native and heterologous genomes and the development of high-integrity data that can overcome the naturally occurring differences between those genomes. Several factors should be taken into account by researchers intending to use non-native references for alignment of read sets. The first step that must always be taken is to identify correspondence between orthologous regions. For example, a researcher with a read set with no complete native reference genome available that has a potential heterologous genome available for read alignment should first investigate if the heterologous genome is viable for alignment. To do this, de novo assembly of reads into contigs, followed by ortholog identification using BLAST should be performed. Then, by extracting read counts for orthologous regions, correspondence can be examined as shown in figure 2.3. It is important to mention that the parameters of ortholog identification here are highly relevant. In this study, we performed ortholog identification using very strict parameters (95% identity, length > 200bp, max mismatch = 5) and were able to identify a large subset of orthologous regions with high confidence. By relaxing these ortholog identification criteria, undoubtedly a larger subset of orthologous genes could be identified, however the false positive rate would also correspondingly increase with increased numbers of mismatching regions existing. A researcher intent on using a non-native reference genome for alignment should then properly tune their BLAST ortholog identification parameters to maximize the number of orthologs they can identify between their read set and non-native reference genome, while confirming viability by monitoring the correspondence of a single read set as aligned to both the non-native reference and their de novo assembled contig sets. That is, as long as an identical read set produces strong correspondence when aligned to the native and non-native alignment target, such as that seen in figure 2.3 of this study, comparison between those orthologous regions can be considered viable. If that alignment instead becomes increasingly dispersed, the strictness of BLAST parameters for ortholog mapping should be increased. Once the researcher has determined an appropriate level of ortholog identification, additional investigation can be performed, if desired, to further eliminate false positive outliers.

In this study, we have observed that genomes with short reads are particularly vulnerable to false positives when using a heterologous genome for read alignment, even with very strict correspondence between orthologous regions. Most false positives originate from sites with a high frequency of polymorphic sites, with a few false positives being caused by other mutation events. Our E. coli samples, which used 50 base reads, contained several false positives that we were able to identify and subsequently analyze to gain insight into underlying causes of incorrect information that must be considered when working with non-native reference genomes. By contrast, our V. vulnificus data set showed that by using longer reads with more depth that false positives can be largely avoided, having almost no false positives at all between native and heterologous genomes. With this being the case, researchers using non-native reference genomes should be aware of these issues and take appropriate precautions in their analyses by confirming both proper identification of orthologous regions and the use of longer reads to mitigate incorrect alignments that result in false positives for heterologous alignment.

Additional accuracy can be achieved when necessary by researchers using heterologous reference genomes. By performing BLAST analysis between read sets and the reference genome to be used, potential false positives can be identified by searching for those reads which align with the highest ratio of polymorphic positions. While increasing read length is certainly the best way to avoid false positives and incorrect information when using a heterologous genome for read alignment, it is likely that as the distance between read data and reference genome increases, that the improvement observed through the use of longer reads would degrade. In general, if it is known that a heterologous genome will be used in a study, longer reads should be generated whenever possible. Additional improvements can be made by adjusting read alignment parameters, but these tend to be fairly modest, with false positives still being likely to occur even with strict alignment parameters. Overall the level of false positives in both cases is low, with the false positive discovery rate compounding as more complex comparisons involving more alignments of data are performed. Using a non-native reference genome for research seems to be a safe endeavor in general if genetic distances between native and non-native conditions are not excessively large, however for very sensitive experiments the use of heterologous reference genomes should be approached with caution, as it is possible that some important genes may be subject to bias without the benefit of a native reference genome.

In this study, we have examined the problems that can arise from the use of non-native, heterologous genomes as references for RNA-Seq read alignment. We have described a method for identifying false positives, outlined the underlying causes, and suggested a set of best practices for studies that use non-native reference genomes, that will allow researchers to make informed decisions about how they handle their data analyses. The analysis workflows described in this study can potentially be applied to novel data sets to help investigators estimate whether it is a safe assumption to use a common reference genome -- either for ease of analysis, or because complete reference genomes for all species or strains in the study are not yet available. In the case where partial genomic information is available, reciprocal mapping analysis can be applied to orthologous genes in the unambiguously alignable portions. These regions can be analyzed to determine the level of correspondence of results between alternate mappings, and to identify the fraction of potential false positives in the analyzable subset of the data. While this will not provide a complete reciprocal analysis, it does provide a quantitative basis by which to justify use of a heterologous common reference for multiple strains, or potentially to justify the expense of finishing additional strain genomes to provide a more accurate reference if available genomes are not sufficient.

CHAPTER 3: SIMULOME: A GENOME SEQUENCE AND VARIANT SIMULATOR

3.1 Background

As new data types and methodologies for analysis of biological data are developed, simulation tools are becoming increasingly necessary to the development, testing, and benchmarking of bioinformatics research. Simulations provide a valuable control case in many contexts, such as the identification of read mapping bias (Degner, et al., 2009), correction of read bias in RNA-seq mapping (Satya, et al., 2012), and analysis of the accuracy of gene expression profiling (Hirsch, et al., 2015). It is therefore necessary to develop simulation software that is flexible, accessible, and able to model a wide range of different genomic conditions.

Currently, several tools exist for simulating read data, such as ART (Huang, et al., 2012) and Mason. Simulation tools such as these approach the problem of simulation from the perspective of reads, and are capable of producing impressive read sets that model variation and sequence errors for a variety of sequencing platforms. Simulome, however, approaches the problem of simulation from the perspective of the reference genome, which expands on the set of potential problems that can be addressed using genomic simulation. When Simulome is used in used in combination with read simulation tools like ART and Mason, entirely simulated experimental scenarios become possible.

3.2 Features and Methods

Simulome is a python-based tool that incorporates biopython (Cock, et al. 2009) to generate synthetic prokaryotic reference genomes by sampling and restructuring existing genomic sequence data. Simulome provides to the ability to create pseudo-reference genomes with a specified gene set and controlled intergenic regions, as well as versions of the simulated genome that contain user con-trolled mutation events. This functionality makes it possible to analyze the effect of specific mutation types on a large scale, providing researchers with the ability to investigate the efficacy of analysis methodologies on a large number of genes that contain similar mutation events, while providing a control genome to which comparisons can be made. Simulome's variant simulations can also be applied to whole prokaryotic genomes, allowing researchers to create variants of existing genomes with mutations introduced according to user specifications.

Simulome features four different run modes for simulating mutation events: SNP mode, synonymous/nonsynonymous mutation mode, insertion/deletion mode, and duplication mode. These run modes can be combined to produce variant genomes containing any combination of mutation events. Further customization of both the control and variant genome is possible through additional optional arguments.

3.2.1 Reference Genome Simulation

Creation of a simulated reference genome requires a nucleotide FASTA file containing the sequence of an existing organism, and an associated annotation file in GTF/GFF3 format. Basing the simulation on the properties of a real genome ensures that simulated genomes accurately model natural sequence variation. and allows genomes with different properties to serve as the base for simulation. For example, a researcher studying the effects of GC bias in RNA-seq data may be interested in simulating genomes with low or high GC content, and could do so by providing a low or high GC content genome to Simulome as input. Users can either use the provided genomic data to create a pseudo-genome of arbitrary size and its mutated variant, or simply create a mutated variant of the entire existing genome. If a pseudo-genome is to be created, gene data is extracted from the base genome and a user-specified number of genes are randomly selected to be used in the simulation. These genes are sampled so as to follow a normal distribution of gene length. The mean length and the standard deviation of all genes in the base genome are calculated, and genes are then sampled such that the mean and standard deviation of the simulated reference genome approximates that of the base genome. Randomness is determined by a seed value, and in the event that a user wishes to repeat a simulation, an exact replication can be produced by reusing the same random seed. Once target genes are selected, the simulated genome is created by interspersing intergenic regions and coding regions. Intergenic regions can be simulated in a variety of ways. Users can either specify the use of randomly generated intergenic regions, or can use real intergenic sequence data from the base genome. Random sequences are generated such that each base has a 25% chance of being selected for any given position. When real intergenic regions are used, all intergenic regions for the base genome are extracted and segments are randomly selected when needed. In both cases, users can specify intergenic length or allow randomly sized intergenic regions to be selected. BLAST searches can optionally be performed on each simulated intergenic region to ensure that there are no unintended duplicate regions in the simulation (Altschul, *et al*., 1990). The simulated genome will be output in FASTA and GFF/GTF3 format.

Additional properties, such as operon inclusion and frequency, can be simulated by specifying optional arguments, which are covered in more depth in the Simulome manual, which is provided in Appendix B.

3.2.2 Variant Genome Simulation

Four run modes are available for simulating SNPs, synonymous/nonsynonymous mutations, indels, and/or duplication events. These can be used in any combination and are applied to each gene in the simulated genome. For example, it is possible to simulate a reference genome and/or a variant genome in which each gene contains a specific number of SNPs and deletions. Each run mode can be configured to introduce either an exact number of mutations in each gene, or otherwise to simulate variants in a range based on a Gaussian distribution with user-defined means and standard deviations. Once mutation events are simulated, sequence and annotation files representing the variant simulated genome are written, as well as an additional file containing meta-data for the introduced variants.

In SNP run mode, a specified number of SNP events are introduced at random locations for each gene in the variant genome. By default, polymorphisms can occur at any position in a gene and no base will mutated more than once. However, additional control is provided for SNP mode to allow for the control of SNP density. Users can specify a window size in which SNP events can occur. To simulate the effect of SNPs on read alignment, a user might create simulations with regions of increasingly dense SNPs. By specifying the same number of SNPs in a decreasingly small window size, it would be possible to quantify the effect of locally clustered SNPs on read alignment on a large number of samples.

In Synonymous/Nonsynonymous run mode, mutations are performed such that a specified percentage of mutations will be synonymous mutations. This run mode assumes that the first position of each gene begins the open reading frame and requires that the user provide distribution parameters to determine the number of mutations that will occur in each gene.

Insertion and deletion run mode allows users to simulate indel events for each gene in the simulated reference genome. Users can choose to include insertion events, deletion events, or both, and can specify the number and length of each event. When both insertions and deletion events are specified, deletion events are performed first in order to preserve mutation integrity across all genes.

Duplication run mode allows the user to simulate scenarios where multiple possible mapping locations for a read are present in a genome. In duplication mode, a duplication percentage is specified by the user. A duplicate region of the specified size (e.g. 1000 nt if 10% of a base genome of 10000 nt is specified) is simulated and added to the base genome. Genes are randomly selected for duplication, along with appropriate intergenic regions, until the desired duplication level is reached.

3.3 Performance

To test Simulome's speed, simulations were performed for reference genomes and mutated variants using Escherichia coli strain K12. These tests were performed on the Red Hat Enterprise Linux 7.2 operating system, with an Intel Xeon 2.53 GHz processor. Results of performance testing can be seen in Table 3.1.

Table 3.1: Simulome execution time. Speed for various run mode combinations for various simulations based on the E. coli genome.

Number of Genes	Simulation Mode	Execution time
1000	Reference only	3m5s
1000	Reference + SNP variant	3m12s
1000	Reference + SNP/Indel variant	3m18s
1000	Reference + SNP/Indel/duplication variant	3m41s
2000	Reference + SNP/Indel/duplication variant	10m5s
3000	Reference + SNP/Indel/duplication variant	16m40s
E. coli (full)	Synonymous/Indel variant	0m48s
E. coli (full)	SNP/Indel variant	0m40s
E. coli (full)	SNP/Indel/duplication variant	0m41s

3.4 Conclusion

Simulome provides a powerful and easy to use tool for creating pseudo-genomes and mutated variants for prokaryotes. Simulome makes it possible to create genomes based on any bacterial species, while controlling for a variety of factors or to directly simulate variations in a complete genome based on user specifications. Furthermore, it provides a range of options that can be used in combination to create mutated variants of the simulated genome, which allows for controlled testing of specific genomic conditions. Simulome can be used in combination with reads generated from next-generation sequencing platforms or alternatively with NGS read simulation packages.

CHAPTER 4: CONCLUSIONS

In this dissertation, we aimed for the identification, quantification, and correction of certain types of bias in next generation sequencing technologies and their associated analysis methodologies. Our results presented in Chapters 1~3 indicate that we have largely achieved the goals.

In chapter 1, we generated and analyzed data based on the PARS methodology to experimentally measure RNA secondary structure in three bacterial strains. This novel data set made it possible to identify the cause of GC content bias on the Illumina sequencing platforms as a function of RNA secondary structure formation. We have described the relationships between RNA secondary structure and GC content and between RNA secondary structure and read depth. We have also shown that fragment counts are significantly biased, with a lower frequency of read starts at sites that are folded into RNA secondary structure conformations. Finally, we have shown that the relationship between read depth and GC content causes increasing bias as GC content increases, with less significant biases being caused in lower GC sequences.

In chapter 2 we approached the problem of non-native reference alignment of RNA-seq reads by comparing the effects of read alignment to native and heterologous reference genomes. We described a method to identify false positives caused by improper alignments to the heterologous reference, and examined the underlying causes to provide a set of best practices for research that makes use of non-native reference genomes. The analysis workflows described in chapter 2 can potentially be applied to novel data sets to help investigators estimate whether it is a safe assumption to use a common reference genome -- either for ease of analysis, or because complete reference genomes for all species or strains in the study are not yet available, providing a quantitative basis by which to justify use of a heterologous common reference in research studies.

In chapter 3 we presented Simulome, a reference genome and variant simulator for creating pseudo-genomes and mutated variants for prokaryotic species. Simulome makes it possible to control for a variety of factors, making it a valuable tool in the investigation of systemic bias in software and experimental methodologies. It also provides a range of options that can be used in combination to create mutated variants of the simulated genome, which allows for controlled testing of specific genomic conditions.

To summarize, we have addressed the complexities of bias identification, quantification, and correction in studies that rely on next generation sequencing technologies and associated analysis methodologies. We have identify the cause of GC bias resulting from mRNA secondary structure formation in next generation sequencing platforms and quantify levels of bias in three bacteria spanning low, medium, and high GC content, we have examined the problem of heterologous reference genome usage when comparing closely related bacterial strains, providing multiple analyses of both real and simulated data to provide a set of best practices for the use of non-native reference genome comparison, and have presented Simulome, a simulation tool to generate synthetic reference genomes and simulate mutations, allowing for investigation into biases caused my experimental and computational analysis methodologies.

REFERENCES

Aird, D., Ross, M. G., Chen, W.-S., Danielsson, M., Fennell, T., Russ, C., … Gnirke, A. (2011). Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome Biology*, *12*(2), R18.

Alm, E. J., Huang, K. H., Price, M. N., Koche, R. P., Keller, K., Dubchak, I. L., & Arkin, A. P. (2005). The MicrobesOnline Web site for comparative genomics. *Genome Research*, *15*(7), 1015–1022.

Altenhoff, A. M., & Dessimoz, C. (2009). Phylogenetic and functional assessment of orthologs inference projects and methods. *PLoS Computational Biology*, *5*(1).

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, *215*(3), 403–10.

Aravind, L. (2000). Guilt by association: Contextual information in genome analysis. *Genome Research*, *10*(8), 1074–1077.

Bao, Y. Y., Tang, X. D., Lv, Z. Y., Wang, X. Y., Tian, C. H., Xu, Y. P., & Zhang, C. X. (2009). Gene expression profiling of resistant and susceptible Bombyx mori strains reveals nucleopolyhedrovirus-associated variations in host gene transcript levels. *Genomics*, *94*(2), 138–145.

Bostock, M., Ogievetsky, V., & Heer, J. (2011). D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, *17*(12), 2301–2309.

Brierley, I., Pennell, S., & Gilbert, R. J. C. (2007). Viral RNA pseudoknots: versatile motifs in gene expression and replication. *Nature Reviews. Microbiology*, *5*(August), 598–610.

Buratti, E., Muro, A. F., Giombi, M., Gherbassi, D., Iaconcig, A., & Baralle, F. E. (2004). RNA folding affects the recruitment of SR proteins by mouse and human polypurinic enhancer elements in the fibronectin EDA exon. *Molecular and Cellular Biology*, *24*(3), 1387–1400.

Cain, A. A., Kosara, R., & Gibas, C. J. (2012). GenoSets: Visual Analytic Methods for Comparative Genomics. *PLoS ONE*, *7*(10).

Carver, T., Rutherford, K., Berriman, M., Rajandream, M., & Barrell, B. (2005). ACT: the Artemis Comparison Tool, *2001*.

Chan, C. Y., Carmack, C. S., Long, D. D., Maliyekkel, A., Shao, Y., Roninson, I. B., & Ding, Y. (2009). A structural interpretation of the effect of GC-content on efficiency of RNA interference. *BMC Bioinformatics*, *10 Suppl 1*(1), S33.

Chaudhuri, R. R., Khan, A. M., & Pallen, M. J. (2004). coliBASE: an online database for Escherichia coli, Shigella and Salmonella comparative genomics. *Nucleic Acids Research*, *32*(Database issue), D296–D299.

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., … De Hoon, M. J. L. (2009). Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, *25*(11), 1422–1423.

Cock, P. J. A., Fields, C. J., Goto, N., Heuer, M. L., & Rice, P. M. (2009). The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Research*, *38*(6), 1767–1771.

Codd, E. F., Codd, S. B., & Salley, C. T. (1993). Providing OLAP (on-line Analytical Processing) to User-analysts: An IT Mandate. *Codd and Date*, *32*, 3–5. Retrieved from

Dandekar, T., Snel, B., Huynen, M., & Bork, P. (1998). Conservation of gene order: A fingerprint of proteins that physically interact. *Trends in Biochemical Sciences*, *23*(9), 324–328.

Darling, A. C. E., Mau, B., Blattner, F. R., & Perna, N. T. (2004). Mauve: Multiple alignment of conserved genomic sequence with rearrangements. *Genome Research*, *14*(7), 1394–1403.

Degner, J. F., Marioni, J. C., Pai, A. A., Pickrell, J. K., Nkadori, E., Gilad, Y., & Pritchard, J. K. (2009). Effect of read-mapping biases on detecting allele-specific expression from RNA-sequencing data. *Bioinformatics*, *25*(24), 3207–3212.

Dohm, J. C., Lottaz, C., Borodina, T., & Himmelbauer, H. (2008). Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Research*, *36*(16).

Fitch, W. M. (1970). Distinguishing Homologous from Analogous Proteins. *Systematic Biology*, *19*(2), 99–113.

Fong, C., Rohmer, L., Radey, M., Wasnick, M., & Brittnacher, M. J. (2008). PSAT: a web tool to compare genomic neighborhoods of multiple prokaryotic genomes. *BMC Bioinformatics*, *9*, 170.

Fraser-Liggett, C. M. (2005). Insights on biology and evolution from microbial genome sequencing. *Genome Research*.

Gama-Castro, S., Salgado, H., Santos-Zavaleta, A., Ledezma-Tejeida, D., Mu??iz-Rascado, L., Garc??a-Sotelo, J. S., … Collado-Vides, J. (2016). RegulonDB version 9.0: High-level integration of gene regulation, coexpression, motif clustering and beyond. *Nucleic Acids Research*, *44*(D1), D133–D143. 6

Hillebrand, G. G., & Beattie, K. L. (1985). Influence of template primary and secondary structure on the rate and fidelity of DNA synthesis. *Journal of Biological Chemistry*, *260*(5), 3116–3125.

Hirsch, C. D., Springer, N. M., & Hirsch, C. N. (2015). Genomic Limitations to RNAseq Expression Profiling. *The Plant Journal*, *84*(3), n/a-n/a.

Huang, W., Li, L., Myers, J. R., & Marth, G. T. (2012). ART: A next-generation sequencing read simulator. *Bioinformatics*, *28*(4), 593–594.

Jacob, F., & Monod, J. (1961). Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, *3*(3), 318–356.

Jordan, I. K., Makarova, K. S., Spouge, J. L., Wolf, Y. I., & Koonin, E. V. (2001). Lineage-specific gene expansions in bacterial and archaeal genomes. *Genome Research*.

Kertesz, M., Wan, Y., Mazor, E., Rinn, J. L., Nutter, R. C., Chang, H. Y., & Segal, E. (2010). Genome-wide measurement of RNA secondary structure in yeast. *Nature*, *467*(7311), 103–107.

Kulikova, T., Akhtar, R., Aldebert, P., Althorpe, N., Andersson, M., Baldwin, A., … Apweiler, R. (2007). EMBL Nucleotide Sequence Database in 2006. *Nucleic Acids Research*, *35*(SUPPL. 1), 16–20.

Kurniawan, A. (2015). *Node.js*. *Succintly* (Vol. 1).

Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nat Methods*, *9*(4), 357–359.

Lee, C. Y., & Iandolo, J. J. (1986). Integration of staphylococcal phage L54a occurs by site-specific recombination: structural analysis of the attachment sites. *Proceedings of the National Academy of Sciences of the United States of America*, *83*(15), 5474–8. Retrieved from

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., … Durbin, R. (2009). The Sequence Alignment / Map (SAM) Format and SAMtools 1000 Genome Project Data Processing Subgroup. *Bioinformatics (Oxford, England)*, *25*(16), 1–2.

Li, L., Stoeckert, C. J., & Roos, D. S. (2003). OrthoMCL: Identification of ortholog groups for eukaryotic genomes. *Genome Research*, *13*(9), 2178–2189.

Liao, Y., Smyth, G. K., & Shi, W. (2013). The Subread aligner: Fast, accurate and scalable read mapping by seed-and-vote. *Nucleic Acids Research*, *41*(10).

Long, J. S. (1997). Regression models for categorical and limited dependent variables. *American Journal of Sociology*.

Lorenz, R., Bernhart, S. S. H., Höner zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P. F. P., … Matthews, B. (2011). ViennaRNA Package 2.0. *Algorithms for Molecular Biology : AMB*, *6*(1), 26.

Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, *15*(12), 550.

Lyubetsky, V. A., Rubanov, L. I., Seliverstov, A. V, & Pirogov, S. A. (2006). Model of Gene Expression Regulation in Bacteria via Formation of RNA Secondary Structures. *Mathematical and System Biology*, *40*(3), 440–453.

Mamanova, L., Andrews, R. M., James, K. D., Sheridan, E. M., Ellis, P. D., Langford, C. F., … Turner, D. J. (2010). FRT-seq: amplification-free, strand-specific transcriptome sequencing. *Nature Methods*, *7*(2), 130–2.

Morrison, S. S., Williams, T., Cain, A., Froelich, B., Taylor, C., Baker-Austin, C., … Gibas, C. J. (2012). Pyrosequencing-based comparative genome analysis of Vibrio vulnificus environmental isolates. *PLoS ONE*, *7*(5).

Munson, G. P., Lam, D. L., Outten, F. W., & O'Halloran, T. V. (2000). Identification of a copper-responsive two-component system on the chromosome of Escherichia coli K-12. *Journal of Bacteriology*, *182*(20), 5864–5871.

Novikova, I. V., Hennelly, S. P., & Sanbonmatsu, K. Y. (2012). Structural architecture of the human long non-coding RNA, steroid receptor RNA activator. *Nucleic Acids Research*, *40*(11), 5034–5051.

Ochman, H., Lawrence, J. G., & Groisman, E. a. (2000). Lateral gene transfer and the nature of bacterial innovation. *Nature*, *405*(6784), 299–304.

Perna, N. T., Plunkett 3rd, G., Burland, V., Mau, B., Glasner, J. D., Rose, D. J., … Blattner, F. R. (2001). Genome sequence of enterohaemorrhagic Escherichia coli O157:H7. *Nature*, *409*(6819), 529–533.

Quail, M., Smith, M. E., Coupland, P., Otto, T. D., Harris, S. R., Connor, T. R., … Salzberg, S. (2012). A tale of three next generation sequencing platforms:

comparison of Ion torrent, pacific biosciences and illumina MiSeq sequencers. *BMC Genomics*, *13*(1), 341.

Ralph Kimball. (2013). *The Data Warehouse Toolkit*. *Journal of Chemical Information and Modeling* (Vol. 53).

Rogozin, I. B., Makarova, K. S., Wolf, Y. I., & Koonin, E. V. (2004). Computational approaches for the analysis of gene neighbourhoods in prokaryotic genomes. *Briefings in Bioinformatics*, *5*(2), 131–49.

Schwartz, J. J., & Quake, S. R. (2009). Single molecule measurement of the “speed limit” of DNA polymerase. *Proceedings of the National Academy of Sciences*, *106*(48), 20294–20299.

Sendler, E., Johnson, G. D., & Krawetz, S. A. (2011). Local and global factors affecting RNA sequencing analysis. *Analytical Biochemistry*, *419*(2), 317–322.

Sonnhammer, E. L. L., & Koonin, E. V. (2002). Orthology, paralogy and proposed classification for paralog subtypes. *Trends in Genetics*.

Suo, Z., & Johnson, K. A. (1997). Effect of RNA secondary structure on RNA cleavage catalyzed by HIV-1 reverse transcriptase. *Biochemistry*, *36*(41), 12468–12476.

Tian, G., Yin, X., Luo, H., Xu, X., Bolund, L., Zhang, X., … Li, N. (2010). Sequencing bias: comparison of different protocols of microRNA library construction. *BMC Biotechnology*, *10*, 64.

Uchiyama, I., Higuchi, T., & Kobayashi, I. (2006). CGAT: a comparative genome analysis tool for visualizing alignments in the analysis of complex evolutionary changes between closely related genomes. *BMC Bioinformatics*, *7*, 472.

Vijaya Satya, R., Zavaljevski, N., & Reifman, J. (2012). A new strategy to reduce allelic bias in RNA-Seq readmapping. *Nucleic Acids Research*, *40*(16).

Vital, M., Chai, B., Østman, B., Cole, J., Konstantinidis, K. T., & Tiedje, J. M. (2015). Gene expression analysis of E. coli strains provides insights into the role of gene regulation in diversification. *The ISME Journal*, *9*(5), 1130–1140.

Wan, Y., Qu, K., Ouyang, Z. Q., & Chang, H. Y. (2013). Genome-wide mapping of RNA structure using nuclease digestion and high-throughput sequencing. *Nat Protoc*, *8*(5), 849–869.

Wan, Y., Qu, K., Ouyang, Z., & Chang, H. Y. (2016). Genome-wide probing of RNA structures in vitro using nucleases and deep sequencing. In *Methods in Molecular Biology* (Vol. 1361, pp. 141–160).

Williams, T. C., Blackman, E. R., Morrison, S. S., Gibas, C. J., & Oliver, J. D. (2014). Transcriptome sequencing reveals the virulence and environmental genetic programs of vibrio vulnificus exposed to host and estuarine conditions. *PLoS ONE*, *9*(12).

Wolf, Y. I., Rogozin, I. B., Kondrashov, A. S., & Koonin, E. V. (2001). Genome alignment, evolution of prokaryotic genome organization, and prediction of gene function using genomic context. *Genome Research*, *11*(3), 356–372.

Zuker, M. (2003). Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, *31*(13), 3406–3415.

APPENDIX A: Gene-RiViT: A visualization tool for comparative analysis of gene neighborhoods in prokaryotes

ABSTRACT:

The genomes of prokaryotes are dynamic and shuffling of gene order occurs frequently, along with horizontal transfer of genes from external sources. Local conservation of gene order tends to reflect functional constraints on the genome or on a biochemical subsystem. Comparison of the local gene neighborhoods surrounding a gene of interest gives insight into evolutionary history and functional potential of the gene. The Genomic Ring Visualization Tool (Gene-RiViT) is a high speed, intuitive visualization tool for investigating sequence environments of conserved genes among related genomes. Gene-RiViT allows the user to interact with interconnected global and local visualizations of gene neighborhoods and gene order, through a web-based interface that is easily accessible in any browser. The primary visualization is a wheel of nested rotating circles, each of which represents a single genome. This visualization is similar to common circular genome alignment views, except that the rings can be realigned with each other dynamically based on user selections within the ring view or one of the coordinated views. By allowing the user to dynamically realign genomes and focus on a locally conserved region of interest, and using orthology connections to highlight corresponding structures among genomes, this view provides insight into gene context and preservation of neighbor relationships as genomes evolve. Visualizations are linked into a coordinated multiple view interface to provide multiple selection methods and

entry points into the data. These approaches make Gene-RiViT a flexible, unique tool for examining gene neighborhoods that improves on existing methods.

INTRODUCTION

It has long been known that multiple independent genes can be coordinately expressed and behave as a single unit in prokaryotic organisms. These units, called operons, encode functionally linked proteins, with a conserved gene order. It has been shown that gene order within operons is often conserved in both closely related and highly divergent organisms, and can therefore be used for making inferences about genes and their functions (Dandekar, et al. 1998; Jacob et al. 1961). For example, if the function of a single gene in a region of conserved gene order for multiple organisms is known, it is possible to infer information about neighboring genes and transfer functional annotation, even when function is known for only a single organism in the comparison (Aravind, et al. 2000). Conversely, if a gene is found in one genome outside of its previously observed functional context, this may be an indication that there is no longer selection pressure acting to keep components of an operon in their functional order. Several tools currently exist that allow their users to examine prokaryotic genes in the context of gene neighborhoods (Alm, et al. 2005; Carver, et al. 2005; Chaudhuri, et al. 2004; Fong, et al. 2008; Uchiyama, et al. 2006). These existing tools are limited in a number of ways. Most lack the capacity for users to use their own data, and are confined to pre-loaded data sets. This may be helpful for studying broad concepts, but can be limiting when specific organisms are being examined and data for those organisms has not been pre-computed. It is also limiting if the user wishes to analyze unpublished data. The existing gene neighborhood analysis tools generally make use of visualizations based on linear alignment to a fixed reference genome. The user queries the database using text-based or pulldown based queries, and then a visualization of a local region is generated. It is not possible to seamlessly move back and forth between the whole genome context and the local neighborhood, or to initiate a query in a comparative data set based on observed relationships in the summary visualization rather than by keyword access to a known region of interest. The Genomic Ring Visualization Tool (Gene-RiViT) is a web based visual analytic tool that uses emerging web technologies to provide a coordinated multiple view interface to a comparative genomic database. The current version of Gene-RiViT combines a familiar, global 2D dotplot view with an adaptive local neighborhood view, demonstrating the potential of a visual analytic approach for real time exploration of conserved gene neighborhoods and their genomic context.

2 GENE ORDER

Comparison of gene order in closely related or even highly diverged genomes can suggest the biochemical context of a gene in a system, though conserved gene order does not necessarily provide complete biochemical information (Wolf, et al. 2001). When multiple prokaryotic genomes are being compared in order to understand gene content differences among strains that may lead to differences in pathogenicity, in host preference, or in survival in the environment, the first line of inquiry is often simply to examine genomic similarities and differences (Morrison, et al. 2012). For instance, if a component of an operon frequently associated with pathogenicity, such as the Type IV secretion system used by many bacteria to transport proteins and toxins out of the cell, is found as a differentiating feature between two bacterial strains with different levels of pathogenicity, this is potentially of interest. The next question, after we identify those potentially interesting differences, is whether the complete Type IV secretion system is present or whether that gene is isolated out of its functional context, perhaps due to a horizontal transfer event or a reshuffling of the genome. The number of prokaryotic genome sequences available is growing rapidly, and comparative studies focusing on identifying core genome features common to a set of genomes, or dispensable features that distinguish them, are common. Gene-RiViT builds on an existing genomic comparative analysis platform (Cain, et al, 2012), adding the capability to dynamically examine the genomic context of these gene discoveries, as well as the gene-level effects of observed insertions, deletions, and inversions on a set of genomes.

Typical sequence alignment visualizations assume that sequences are collinear, and don’t adequately display permutations in gene order, especially when multiple genomes are being compared. At the genome level, however, gene order is commonly rearranged and analysis of these permutations cannot be disregarded in a comparison procedure (Rogozin, et al. 2004). Analysis of gene order conservation using gapped local alignments of 25 prokaryote genomes has shown that 5-25% of the genes in bacterial and archaeal genomes belong to gene strings that are shared by at least two of the examined genomes, once closely related species were excluded (Wolf, et al. 2001). Gene-RiViT addresses the pervasive permutation problem associated with analysis of gene order, by providing visualizations that make rearrangements in gene neighborhoods obvious and comparable between multiple genomes.

3 ORTHOLOGY

In order to compare gene order and identify commonalities among different genomes, it is first necessary to determine orthology relationships between genes.

Orthologs are defined to be homologous genes that diverged from an ancestral gene in the most recent common ancestor of the species under comparison, while paralogs are genes that are related by a gene duplication event in an ancestral gene (Firch, 1970). Co-orthology refers to paralogs produced by the duplications of orthologs subsequent to a given speciation event, a phenomenon which is commonly observed between distantly related species (Jordon, et al. 2001). Inparalogs are paralogs in a given lineage that evolved by gene duplications occurring after a given speciation event (Sonnhammer, et al. 2002). Gene-RiViT uses OrthoMCL to cluster genes by orthology, coorthology, and inparalology. OrthoMCL identifies orthologous groups from the results of all-against-all BLAST comparisons, identifying reciprocal best hits (Ochman, et al. 2000). This method of ortholog clustering is based on the principle that orthologous genes are the most similar among all compared pairs of genes (Li, et al. 2003). OrthoMCL has been shown to outperform other clustering methods in terms of efficiency and accuracy (Altenhoff, et al. 2009), however the modular design of the database that supports Gene-RiViT allows for straightforward substitution of other methods for identifying orthologs as new methods evolve. The use of orthoMCL to prepare genomic data for analysis in Gene-RiViT allows any set of genomes to be compared to one another, regardless of whether or not they have been made publicly available or incorporated into existing orthology databases such as Clusters of Orthologous Groups of proteins(COG).

4 GENE-RIVIT

Gene-RiViT uses three main modules to process data and provide it to the user: an OLAP database, a custom-built web server, and the client-side visualization. Each of these modules is designed to be scalable and to allow for fast interaction with genome-scale data. Gene-RiViT incorporates multiple coordinated views that use state of the art web technologies to create a dynamic, visually appealing and intuitive interface that provides researchers with the ability to contrast the relationships between genes. The publically available interface requires no setup and is capable of visualizing any combination of prokaryotic genomes. Users can also set up GeneRiViT to run on local systems with relatively little setup. We currently provide support for importing any genome available in the EMBL database, however, the GenoSets back-end which supports the Gene-RiViT system also supports import and annotation of unpublished genome data.

4.1 Web Server

Gene-RiViT relies on custom-built middleware that functions as a web server and data processing hub between the database and the visualization. This module of Gene-RiViT was developed using Node.js, an efficient and scalable platform for data-intensive, real-time applications. As the amount of genomic data processed can be quite substantial, it was important to address the issue of network latency when designing Gene-RiViT. Node.js achieves high performance when processing large amounts of data, by using the performance optimized javascript V8 engine along with a non-blocking, asynchronous model for data processing and communication. This allows Gene-RiViT to efficiently handle on-the-fly queries on genome-scale data sets and send results back to users over

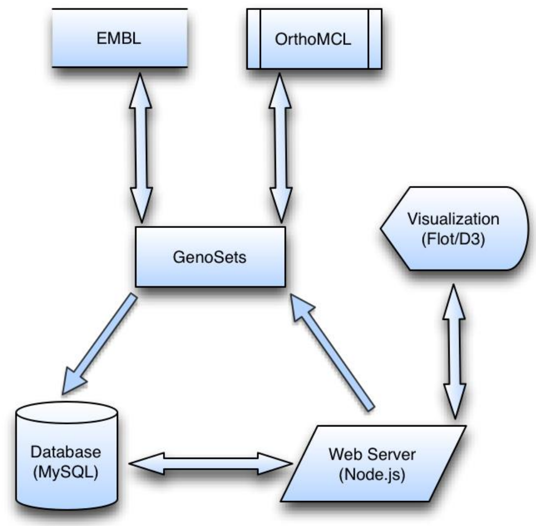

Figure A0.1: A central web server processes data between the database and the client-side visualizations. New data is generated via GenoSets, which retrieves and clusters data before adding it to the database.

the net at speeds sufficient to allow for seamless interaction. Figure A0.1 shows the overall architecture of Gene-RiViT. Of particular interest is the central role of the web server in handling data processing and communication. When a user interacts with a gene of interest in the visualization, the web server will query the database for information about that gene, homologous genes in other organisms that are being examined, and their neighboring genes. The results are then processed into a format that can be recognized by the visualization and are returned over the network as they are processed. The asynchronous processing of node.js allows data to be effectively streamed back to the user, rather than returned as a single, large block. This allows users to interact with large amounts of complex data in real-time.

4.1.1 GenoSets

GenoSets is a comparative genomic analysis platform that supports annotation parsing, manages ortholog clustering via orthoMCL, assigns Gene Ontology (GO) terms to genes, and structures data in the database with consistent gene definitions for a set of genomes [5]. GenoSets provides functionality for Gene-RiViT that lets researchers specify any dataset in the EMBL-Bank public repositories (Kulikova, et al 2007). Through the Gene-RiViT interface, researchers can select any combination of genomes from a list of completed microbial projects or EMBL accession id. User requests from the visualization layer are passed to the web server, which uses GenoSets to download the specified genome data, cluster the data using orthoMCL, and load the processed data into the database. When the process is complete, users are notified by e-mail that their data is ready for viewing. The amount of time required for processing is dependent on the size and number of genomes being loaded. A trial process that was run to cluster and load data of six different E. coli strains took approximately two hours to complete on a desktop computer, however, this is not an innate restriction on the system. Once data is loaded, users can switch between data sets through the Gene-RiViT interface and access any previously loaded data. The entire process of downloading, ortholog clustering, and loading data into the database, however, is a completely invisible process, which contributes to the ease of use of Gene-RiViT.

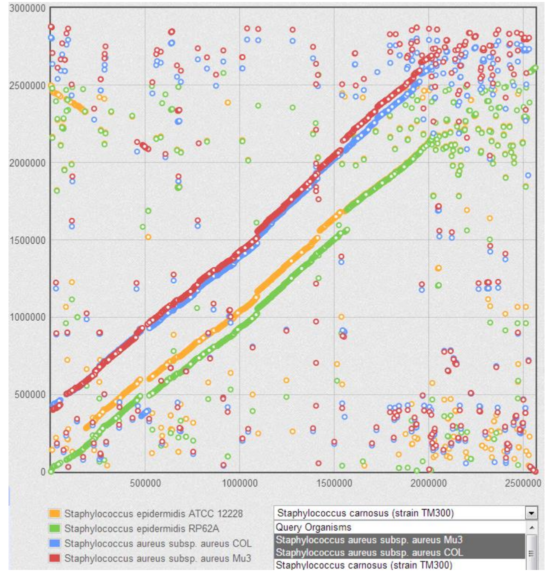

Figure A0.2: Homologous genes in four strains of staphylococcus with respect to a reference strain. The dot plot view provides a global representation of homologous gene positions of multiple organisms as compared to any selected reference genome. The x axis represents the position of the gene in a selected reference and the y axis shows the position in selected query genomes.

4.2 Database

Gene-RiViT uses a multi-dimensional architecture to support Online Analytical Processing (OLAP): a model typically used in business intelligence software to support real-time, ad hoc querying of data at different levels of granularity (Codd, et al 1993).

The database employs a star schema, in which source data is partitioned into facts that represent different dimensions of information. This database design facilitates the fast querying capacity that is necessary for interactivity with such a large amount of data. For example, information about a gene's location in the genome is stored separately from information about the ortholog clusters that gene may belong to. Each table in the database stores a different type of information about the gene and all of these dimensions of information are relationally linked through a central table (Kimball, et al. 2002). By keeping the data separated and stored at this level of detail, it is possible for queries to be made for only the information that is necessary on tables that contain only the necessary information. This reduces the overhead that would otherwise slow down the querying process if the database were required to search for and parse out scattered chunks of information from several genomes worth of data.

4.3 Visualization

Gene-RiViT incorporates multiple views that are implemented using javascript-based visualization libraries. The two main libraries used are Flot and D3 (Bostock, et al. 2011). Flot is a javascript-based plotting library that uses the jQuery framework and HTML5 canvas to produce graphical plots on the fly on the client side. The Data Driven Documents library, or D3, is a fast and efficient javascript library for developing interactive web-based visualizations. Both of these readily available technologies ensure compatibility on almost any system with no setup or configuration for the user, in addition to providing fast interfaces that allow users to view and interact with genome-scale data.

4.3.1 Plot View

Gene-RiViT provides researchers with multiple coordinated views of prokaryotic genome data. A global view is provided as a dot plot, in which positions of genes are plotted on a graph with respect to their positions in a reference organism. The reference organism can be selected from a list of any of the genomes that have been loaded into the database. Any number and combination of genomes can be selected for viewing with respect to the selected reference organism, and the reference genome can be changed dynamically. Each organism is represented as a different color in the plot and standard visualization features, such as panning and zooming, are incorporated. Figure A0.2 shows an example of four strains of Staphylococcus plotted with respect to a fifth strain. The strong main diagonal in this plot indicates that there is high gene-order similarity between these organisms, which is not unexpected as they are very closely related. However, the dot plot makes insertions, deletions and rearrangements easily visible as well as giving access to off diagonal similarities that would simply show up as gaps in a standard linear reference-based genome alignment. Considering the large amount of data represented in the dot plot, it is necessary to incorporate methods for locating areas that might be of interest. A number of methods for accessing the data presented in the dot plot are implemented. Annotation data stored in the database is provided to the user as a list of Gene Ontology terms. The interface provides a method for selecting GOterms, such as cell surface binding, from a menu, which results in all genes that are either annotated or homologous to genes with the selected function being highlighted in the plot, while all genes without the specified function will be colored grey to make the selected genes more obvious. This functionality can help researchers to identify genes in potentially interesting functional categories as starting points for more detailed exploration, or to improve on the level of detail provided in annotation information.

4.3.2 RiViT View

While the dot plot view provides an overall picture of the organization of genes for selected organisms on a global scale, the RiViT view provides a local context in which to examine relative ordering and reshuffling of genes. The RiViT view consists of a series of nested, rotating semi-circles, each of which represents the ninety genes around a central gene that is aligned with homologous genes in other circles. A gap is incorporated at the nine o'clock position of the view to show that there is a discontinuity between the genes shown in the view and that the circles represent only a local view of a larger volume of information. When the circles rotate clockwise, genes from previously viewed regions of the genome fade into the circle from the gap, while genes entering the gapped region fade out. The opposite is true when rotation is counter clockwise. For cases where an alignment exists, but homologous genes are not found for every organism selected, the genes on an alignment for organisms with no matches are shaded grey to indicate that there is no alignment for that organism. In the opposite case, where multiple possible alignments are found in a single organism, navigation buttons are provided that allow users to rotate the circle for that organism through the set of existing matches. The number of rings displayed in the RiViT view is dependent on the number of organisms selected for investigation. The previous example of four staphylococcus strains aligned to a fifth reference strain would result in five color-coded rings representing the selected organisms. When a user interacts with genes in the dot plot view, these rings rotate to align homologous genes for all organisms at the three o'clock angle. A user specified

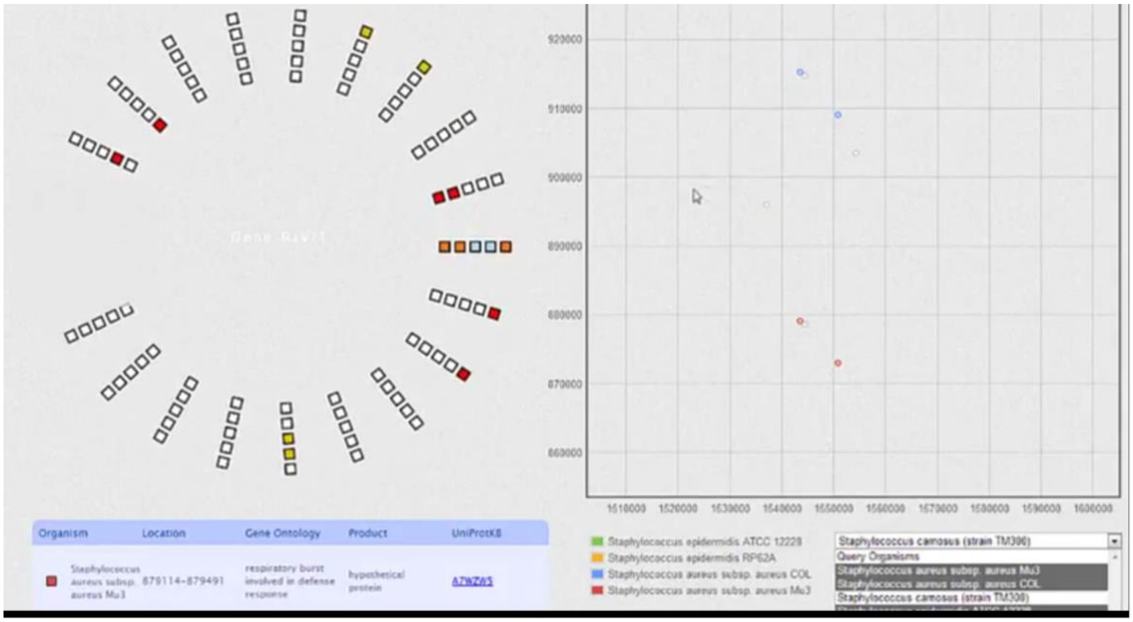

Figure A0.3: A conserved operon examined using Gene-RiViT. Left(a): The RiViT view provides local context information about gene neighborhoods. Homologous genes are aligned in green. Genes shown in grey show that no homologs were found for an organism. Right(b): A zoomed view of the dot plot view shows a conserved operon for two query species. Five homologous genes are highlighted around a central selected gene. All other points are shaded grey to reduce visibility.

number of neighboring genes around the selected gene are also checked against neighboring genes in the query organisms for homology and color-coded accordingly. The RiViT view, then, provides researchers with information about relative restructuring or conservation that has occurred at a local level between organisms under study without regard to long-range structural changes. This allows genes and gene neighborhoods to be examined and compared in a local context that can provide information about how specific genes might be functionally linked. Figure A0.3a shows an example of the RiViT view aligning a set of genes from a conserved operon in staphylococcus. Users are also able to select any gene in the RiViT view to make that gene the new point of reference. In this way, gene neighborhoods can be smoothly explored by allowing users to switch between different organisms that have differing homology relationships, or move in steps along the genome to explore wider ranges of neighborhood information. For example, organisms a, b, and c may have an orthologous match that can be aligned by selecting any of the genes in a specific ortholog cluster. Organisms c and d, may have related genes in the same neighborhood that can be examined by switching focus to organism c. This sort of exploratory analysis could be particularly useful in comparing genomes of species at varying evolutionary distance.

4.3.3 Details View

In addition to the interactive views discussed above, a detailed list of information about genes is provided. This list provides details about each of the genes in the alignment, as well as a list of neighboring genes that can be selected to view details. By default, size, starting and ending positions in the genome, and annotation information are displayed, though users can select additional annotation information to view by selecting options from a menu. Selecting a new gene for investigation will automatically update the list to show information about genes and their neighbors in the new alignment.

5 DISCUSSION

Gene-RiViT is designed to be useful in a variety of research situations. In more recently divergent organisms, Gene-RiViT can be used to examine the effects of gene rearrangement by comparing multiple organisms. In more evolutionarily distant organisms, it can also be used to identify and examine the details of conserved sets of genes and for functional inference by association. A simple example of the utility of Gene-RiViT was performed to illustrate proof of concept. Figure 3 shows Gene-RiViT in use. In this instance, four strains of Staphylococcus are being examined: aureus Mu3, aureus COL, carnosis TM300, and epidermitis ATCC 12228. In this case, strain Mu3 was selected as the reference strain. A region was selected based on a local alignment of genes, visible as a short diagonal, that was observed in the global dot plot view shown in figure A0.2. The area was zoomed in and a gene from the center of the aligned region in strain aureus COL(red) was selected. Points in the plot that were not homologous to one another in a range of one hundred genes around the selected gene were shaded grey in the plot, showing homologous genes in the region between the reference genome, strain Mu3, and strains TM300(blue) and COL(red). The RiViT view rotated to align the gene selected in the plot with the homologous genes in other organisms. Detail information about the aligned genes was then provided in the detail view. In the case of strain ATCC 12228, no homologous genes were found and the visible genes from this species were shaded grey in the alignment to show that there were no gene neighborhood matches. The alignment, highlighted in green, shows that two genes prior to the selected gene are also homologous to one another, as is the following gene and another gene six gene positions away.

Table A1: Gene neighborhood alignment annotation information

Distance	Mu3	COL	TM300	ATCC 12228
-2	hypothetical protein	prophage L54a, terminase, large subunit,putative	putative phage terminase, large subunit	No match
-1	hypothetical protein	prophage L54a, Clp protease, putative	putative Clp protease, phage associated	No match
0	hypothetical protein	hypothetical protein	hypothetical protein	No match
1	hypothetical protein	conserved hypothetical protein	conserved hypothetical protein	No match
6	phi PVL ORF 15 and 16 homologue	prophage L54a, tail tape meausure protein, TP901family	truncated phiSLT orf2067-like protein	No match

Table A1 shows annotation information for the aligned genes, with the distance column indicating the distance in steps away a gene is from the selected gene, and zero indicating the gene that was selected. In this case, the gene selected was annotated as a hypothetical protein in each of the organisms under comparison. Examination of the surrounding homologous genes, however, shows very similar annotation information between the COL and TM300 strains. Based on the homology information showing known orthology in a conserved range in multiple genomes with a lack of rearrangements, it is possible to infer with reasonable confidence that the function of the hypothetical proteins at the center of the alignment is related to prophage L54a, though this could be experimentally verified to obtain a higher level of confidence and precision. Because the information about the known genes shows that these genes are phage related and have a conserved order spanning multiple genes, most likely these genes exist as a result of a horizontal transfer event (Ochman, et al. 2000). Investigating prophage L54a revealed that the integration of prophage L54a results in a loss of lipase activity in Staphylococcus aureus PS54 due to insertion at the 3’ end of the lipase structural gene (Lee, et al, 2003). A researcher using Gene-RiViT could verify this result by navigating through the local context provided by the RiViT view to see if neighboring genes were in fact involved in lipase activity.

Researchers can perform a variety of studies using Gene-RiViT. The previous example illustrates that Gene-RiViT can be used to identify local, functionally significant similarities among genomes even when genomes are not completely collinear. Gene-RiViT is not restricted to only this use. Researchers could use Gene-RiViT, for example, to identify genes associated with pathogenicity in several related species and make observations about their functions and implications in other species who either share the same genes or are specifically lacking them. Gene-RiViT could also be used to make more intelligent decisions about evolutionary distance between closely related species in cases where precision is a factor by allowing researchers to examine the scope of rearrangements that have occurred within a collection of genomes.

6 CONCLUSION AND FUTURE WORK

We presented Gene-RiViT, a visual analytic tool for the on-the-fly analysis of gene neighborhoods in bacterial genomes. Gene-RiViT provides a coordinated set of visualizations for examining genomic data at multiple levels of granularity, with particular focus on gene order in local gene neighborhoods. Gene-RiViT uses state of the art web technology to present data as a dynamic and adjustable alignment, rather than the more common presentation of a fixed alignment pegged to a reference genome. The ring visualization in GeneRiViT allows the user to pick any gene in any of the genomes in the set as the query, upon which the entire genomic alignment rapidly rearranges to bring orthologs in the other genomes into alignment with the query. Highlights on the genome then show orthology relationships in the neighborhood surrounding the query gene, giving insight into the conservation of local genome context and the preservation of functional operons. A second visualization of the genomes being compared as a familiar 2D dot plot allows the user to pinpoint regions of interest based on observed features in the 2D alignment. Selections in the dot plot visualization can be used to control and highlight the dynamic genome ring visualization, and vice versa. Keyword searches and Gene Ontology based searches are also available as entry points to the data. Gene-RiViT has a wide variety of potential uses in comparative genomics studies and will be freely available and easily accessible to the public. The visualization tools in Gene-RiViT are designed to function as part of a coordinated multiple view interface that includes multiple methods for target gene selection. For instance, we have previously implemented a java Parallel Sets visualization, used in conjunction with a visualization of Gene Ontology categories in a tree map view, to rapidly identify common and differentiating genes in multiple genome data sets and to further subdivide those gene lists by functional category. Gene-RiViT provides an intermediate level of detail between these high level abstractions of the genome data set and the literal linear views to which biologists are accustomed. Integration of Gene-RiViT with Parallel Sets and Gene Ontology hierarchy views is planned, along with incorporation of other feature markup such as operon predictions.

APPENDIX B: SIMULOME MANUAL

Simulome

October 21, 2016

Version: 1.2

Title: Simulome: Prokaryote genome and variant simulator.

Author: Adam Price

Maintainer: Adam Price <price0416@gmail.com>

Description: Simulome provides a powerful and easy to use tool for creating pseudo-genomes and mutated variants for prokaryotes. Simulome makes it possible to create genomes based on any bacterial species, while controlling for a variety of factors. Furthermore, it provides a range of options that can be used in combination to create mutated variants of the simulated genome, which allows for controlled testing of specific genomic conditions. Simulome can be used in combination with reads generated from next-generation sequencing platforms or alternatively with NGS read simulation packages.

URL:

Dependencies

Simulome was developed in a linux/unix environment and requires the following libraries for proper functionality.

- Python 2.7.2
- Biopython 1.6.1+
- BLAST 2.3.0+

Description

Simulome takes an existing genome and the corresponding annotation information for that genome and samples a subset of the genes to use as a simulated genome. Sampling is performed based gene length and genes are selected to approximate a normal distribution of read lengths. That is, the mean length of all genes in the provided reference genome and the standard deviation are calculated, and genes are then sampled such that the mean and standard deviation of the simulated reference genome approximates that of the originally provided genome. An initial simulation is created by using these sampled genes in conjunction with non-duplicating intergenic regions, or by randomly sampling from the intergenic regions of the provided reference genomes. Once the initial genome is simulated, a variant genome can be simulated to meet desired specifications. Alternatively, users can specify not to simulate a pseudo-genome and can directly apply Simulome's variant tools to create a mutated genome based directly on the provided reference genome. Four run modes are available and can be used in any combination to produce variants containing SNPs, Synonymous/nonsynonymous mutations, indels, and/or duplicate regions. Additional optional arguments are available to allow direct control over selection criteria and genomic structure. The resulting simulations will each be provided as a FASTA nucleotide file, a GTF/GFF3 annotation file, and a variant metadata file.

Usage

python simulome.py --genome <genome.fasta> --anno <genome.gff> --output <destination> <RUN MODE> <OPTIONAL ARGUMENTS>

Required Arguments

--genome	File representing genome. FASTA nucleotide format.
--anno	File containing genome annotation information in GTF/GFF3 format. This file should correspond to the FASTA file representing the selected genome.
--output	Output destination. This option will create a folder named with the supplied argument containing output files. Providing a –o option of 'ecoli' will create the directory, ./ecoli/ and populate it with files such as: ./ecoli/ecoli_simulated.fasta

SNP Run Mode Arguments

--snp	Boolean. Set this option to TRUE to enable SNP mutations in the variant genome.
--num_snp	The number of SNPs to simulate per gene. This argument is required for SNP run mode.
--snp_window	Window size in which to simulate SNPs. This option allows control over the density of SNP mutations. If a window size is specified, the number of SNPs specified by the –s option will occur within a randomly determined range of this specified window size.

I.E. <-s 5 –w 10> will create 5 SNPs within a 10 base pair window. If this option is not specified, SNPs will be distributed randomly over the length of each gene.

--snp_distrib

Boolean. Create different numbers of SNPs in each gene based on a Gaussian distribution. If this option is true, --num_snp will be used as the mean of the distribution.

--snp_std_dev

This option is required if –-snp_distrib=true. Standard deviation for the distribution of SNP counts for each gene. A larger standard deviation will result in a wider range of SNP counts per gene, and a smaller deviation will result in a more condensed range.

Synonymous/Non-synonymous Run Mode

--syn

Boolean. Set this option to TRUE to enable Synonymous/Non-synonymous run mode. This run mode allows you to specify a percentage of synonymous mutations to occur in each gene. It assumes the start position of the gene to be the open reading frame. Requires “mutation_log.dat” file as provided, in $PATH or local directory.

--syn_percent

The percentage of mutations per gene that will be synonymous.

--syn_mean

The mean number of total mutations desired per gene.

--syn_std_dev

Standard deviation for the distribution of mutations counts per gene. A larger standard deviation will result in a wider range of total number of mutations, and a smaller deviation will result in a more condensed range.

Insertion/Deletion Run Mode

--indel	This option specifies insertion/deletion for mutations in the variant genome. Possible values are: 1 = Insertions only. 2 = Deletions only. 3 = Both insertions and deletions.
--ins_len	Length of insertion events. Required for insertion mode.
--num_ins	Number of inserts to simulate in each gene. Default = 1.
--is_copy_event	Boolean. If this option is true, insertions sequences will be randomly copied from existing regions of the genome.
--ins_distrib	Boolean. Create different length insertion sequences in each gene based on a Gaussian distribution. If this option is true, --num_ins will be used as the mean of the distribution.
--ins_std_dev	This option is required if –-ins_distrib=true. Standard deviation for the distribution of insertion lengths. A larger standard deviation will result in a wider range of insertion lengths, and a smaller deviation will result in a more condensed range.
--del_len	Length of deletion events. Required for deletion mode. Deletions cannot be longer than the target genes, in which event, genes shorter than desired deletion

length will be omitted from mutation and warnings will be displayed.

--num_del	Number of deletes to simulate in each gene. Default = 1.
--del_distrib	Boolean. Create different length deletion events in each gene based on a Gaussian distribution. If this option is true, --num_del will be used as the mean of the distribution.
--del_std_dev	This option is required if –-del_distrib=true. Standard deviation for the distribution of deletion event lengths. A larger standard deviation will result in a wider range of deletion lengths, and a smaller deviation will result in a more condensed range.

Duplication Run Mode

--duplicate	Boolean. Set this option to TRUE to create duplications in the variant genome. Allows control for reads that map to multiple locations. Uses the initial genome simulation and appends duplicate regions until the desired level of duplication is reached.
--percent_dup	Percent of duplicate regions to include in the genome. Required for duplication mode.

Optional Arguments

--whole_genome	Boolean. If this is true, the provided genome will be used instead of a simulated pseudo-genome and variants will be performed directly on the provided reference. Cannot be used with –num_genes.

--num_genes	Number of genes to simulate. Default = 100.
--sort_log	How to sort the variant log file. Acceptable options are 'genome' and 'mutation'. 'Genome' will sort the output log by the order mutations occur in the genome, while 'mutation' will sort the output log in the order mutations were created. (Default=Genome)
--intergenic_len	Length of intergenic regions. For random length intergenic regions, specify 0 for this option. Random intergenic length range is 0-2000. Default = 500.
--random_intergenic	Boolean. If this is true, intergenic regions will be randomly synthesized between genes. If false, intergenic regions from the provided genome will be randomly sampled. (Default=False)
--operon_level	Simulate operons. Input should be approximate percentage of desired operon content. Default = 0.
--seed	Specifies a seed for the random number generator. By default a random seed will be selected for each run. By specifying a seed, the same gene selection and mutations can be repeated identically across multiple runs.
--type	Feature type to simulate from annotation file. I.E: gene, exon, CDS. Case sensitive. Note that this must match the desired feature type in the annotation file provided. Default = gene.

--strict_dup	Boolean. Allow duplicate sequence regions to exist in the initial genome simulation. Selecting FALSE for this option will BLAST each gene and simulated intergenic region against the growing simulation and prevent duplicate regions from being included in the genome. Depending on the level of natural duplication in the genome provided, this may result in fewer genes existing in the genome than specified. Can be memory intensive in some cases. Default = False.
-v, --verbose	Verbose level. Default = 1. [0 = Quiet, 1 = Verbose, 2 = Very Verbose]

Examples

- Simulate a genome based on e.coli containing 100 genes, output files to a folder called ecoli_simulation/.

python simulome.py --genome=ecoli_genome.fasta --anno=ecoli_anno.gtf --output=ecoli_simulation --num_genes=100

- Simulate a genome based on e.coli containing 500 genes, and a variant of the simulated genome in which each gene contains 10 SNPs, output to a folder called ecoli_simulation/.

python simulome.py --genome=ecoli_genome.fasta --anno=ecoli_anno.gtf --output=ecoli_simulation --num_genes=500 --snp=TRUE --num_snp=10

- Simulate a genome based on e.coli containing 500 genes, and a variant of the simulated genome in which each gene contains a variable number of SNPs based on a Gaussian distribution with a mean of 10 and a standard deviation of 3, output to a folder called ecoli_simulation/.

python simulome.py –genome=ecoli_genome.fasta –anno=ecoli_anno.gtf –output=ecoli_simulation --num_genes=500 --snp=TRUE --num_snp=10 --snp_distrib=true --snp_std_dev=3

- Simulate a genome based on e.coli containing 500 genes, and a variant of the simulated genome in which each gene contains a number of Synonymous/nonsynonymous mutations based on Gaussian distribution with a mean of 10 and a standard deviation of 3. In each case, approximate 70% of mutations to be synonymous. Output to a folder called ecoli_simulation/.

```
python simulome.py –genome=ecoli_genome.fasta –anno=ecoli_anno.gtf –output=ecoli_simulation --
num_genes=500 --syn=TRUE --syn_percent=70 --syn_mean=10 --syn_std_dev=3
```

- Simulate a genome based on e.coli containing 500 genes, and a variant of the simulated genome in which each gene contains 10 SNPs that are concentrated in 50 base pair windows, output to a folder called ecoli_simulation/.

```
python simulome.py –genome=ecoli_genome.fasta –anno=ecoli_anno.gtf –output=ecoli_simulation –
num_genes=500 --snp=TRUE –-num_snp=10  --snp_window=50
```

- Simulate a genome based on e.coli containing 100 genes, and a variant of the simulated genome in which each gene contains an insertion event of length 100, output files to a folder called ecoli_simulation/.

```
python simulome.py --genome=ecoli_genome.fasta --anno=ecoli_anno.gtf –output=ecoli_simulation --
num_genes=100 --indel=1 --ins_len=100
```

- Simulate a genome based on e.coli containing 100 genes, and a variant of the simulated genome in which each gene contains an insertion event of length 100, and two deletion events of length 25, output files to a folder called ecoli_simulation/.

```
python simulome.py --genome=ecoli_genome.fasta --anno=ecoli_anno.gtf --output=ecoli_simulation --
num_genes=100 --indel 3 --ins_len=100 --del_len 25 --num_del=2
```

- Simulate a genome based on e.coli containing 100 genes, and a variant in which 10% of the genome is duplicated, output files to a folder called ecoli_simulation/.

```
python simulome.py --genome=ecoli_genome.fasta --anno=ecoli_anno.gtf --output=ecoli_simulation --
num_genes=100 --duplicate=TRUE  --percent_dup=10
```

- Simulate a genome based on e.coli containing 100 genes, with a variant genome in which each gene contains 5 SNPs, an insertion of length 500, a deletion of length 100, 10% genome duplication, and random intergenic region lengths. Output files to a folder called ecoli_simulation/.

```
python simulome.py --genome=ecoli_genome.fasta --anno=ecoli_anno.gtf --output=ecoli_simulation --
num_genes=100 --snp=TRUE --num_snp=5 --indel 3 --ins_len=500 --del_len=100 --duplicate=TRUE --
percent_dup=10
```

- Using the whole reference genome, simulate a variant genome in which each gene contains insertions with lengths based on a distribution with a mean of 100 and a standard deviation of 20, and a number of codon mutations with a total mean number of mutations of 15 and a standard deviation of 7, of which approximately 60 percent will be synonymous, output files to a folder called ecoli_simulation/.

```
python simulome.py --genome=ecoli_genome.fasta --anno=ecoli_anno.gtf --output=ecoli_simulation --
indel=1 --ins_len=100 --ins_distrib=TRUE --ins_std_dev=20 --syn=TRUE --syn_percent=60 --syn_mean=15
--syn_std_dev=7
```

- Using the whole reference genome, simulate a variant genome in which each gene contains deletions with lengths based on a distribution with a mean of 50 and a standard deviation of 25, and a number of codon mutations with a total mean number of mutations of 10 and a standard deviation of 3, of which approximately 30 percent will be synonymous, additionally creating 10% genome duplication. Use full verbose mode. Output files to a folder called ecoli_simulation/.

```
python simulome.py --genome=ecoli_genome.fasta --anno=ecoli_anno.gtf --output=ecoli_simulation --
indel=2 --del_len=100 --del_distrib=TRUE --del_std_dev=50 --syn=TRUE --syn_percent=30 --
syn_mean=10 --syn_std_dev=3 --duplicate=TRUE --percent_dup=10 --verbose=2
```