YOUR TITLE GOES HERE

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Scott Clark

August 2012

YOUR TITLE GOES HERE

Scott Clark, Ph.D.

Cornell University 2012

Your abstract goes here. Make sure it sits inside the brackets. If not, your biosketch page may not be roman numeral iii, as required by the graduate school.

BIOGRAPHICAL SKETCH

Scott Clark grew up in Tigard, Oregon and graduated from Central Catholic High School in Portland, Oregon in 2004. He recieved Bachelor of Science degrees in Mathematics, Computational Physics and Physics (Magna Cum Laude) from Oregon State University in 2008.

In 2008 Scott was awarded a Department of Energy Compuational Science Graduate Fellowship (CSGF) for his doctoral work at Cornell University Center for Applied Mathematics (CAM).

This document is dedicated to ...

ACKNOWLEDGEMENTS

Your acknowledgements go here. Make sure it sits inside the brackets.

My research was supported by a Department of Energy Computational Science Graduate Fellowship, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-FG02-97ER25308. I was also supported by a Startup and Production Allocation Award from the National Energy Research Scientific Computing Center (NERSC), which is supportedby the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Part I

ALE: Assembly Likelihood Evaluation

CHAPTER 1

ALE INTRODUCTION

CHAPTER 2

ALE METHODS

2.1 The ALE Score and the likelihood of an assembly

The ALE framework is founded upon a statistical model that describes how reads are generated from an assembly. Given a proposed assembly (a set of scaffolds or contigs), S, and a set of reads R, this probabilistic model gives the likelihood of observing this set of reads if the proposed assembly were correct. We write this likelihood, $P(R|S)$, and its calculation includes information about read quality, agreement between the mapped reads and the proposed assembly, mate pair orientation, insert length (for paired end reads), and sequencing depth. This statistical model also provides a Bayesian prior probability distribution $P(S)$ describing how likely an assembly S would be, if we were unable to observe any read information. This prior probability is computed using the k-mer distribution of the assembly.

The ALE score is computed from these two values, and it is proportional to the probability that the assembly S is correct. We write this probability as $P(S|R)$. Bayes rule tells us that this probability is

$$P(S|R) = \frac{P(S|R)\,P(S)}{Z} \tag{2.1}$$

where Z is a proportionality constant that ensures that P(S—R) is a probability distribution. We see P(S—R) as a statistical measure of the overall quality of an assembly S. As is typical in large-scale applications of Bayesian statistics, it is computationally intractable to compute the constant Z exactly. The ALE score is computed by replacing the constant Z with an approximation described in the

Methods section.

Although the ALE score can be reported as a standalone value, and understood as an approximation to P(R—S), it is most useful for comparing two different assemblies of the same genome, using the same set of reads to evaluate them. As shown in the Methods section, the assembly with the higher ALE score is then also the one with the larger probability of being correct. Moreover, we show that the difference between two assemblies ALE scores describes their relative probabilities of correctness. If one assemblys ALE score is larger than the others, with a difference of x between their ALE scores, then the assembly with the larger ALE score is more likely to be correct by a multiplicative factor of ex. Below, we refer to the ALE score more precisely as the total ALE score, to differentiate it from the sub-scores (described below) used to construct it.

Figure 1 shows the pipeline used to compute the total ALE score. Given a set of reads and a proposed assembly, ALE first takes as input the alignments of the reads onto the assembly in the form of a SAM or BAM file (Li and Durbin 2009), which can be produced by a third-party alignment algorithm such as bowtie (Langmead et al. 2009) or bwa (Li et al. 2009). ALE then determines the probabilistic placement of each read and a corresponding placement sub-score for each mapped base, which describes how well the read agrees with the assembly. In the case of paired end reads, ALE also calculates an insert sub-score for all mapped bases of the assembly from the read pair, which describes how well the distance between the mapped reads matches the distribution of lengths that we would expect from the library. ALE also calculates a depth sub-score, which describes the quality of the sequencing depth accounting for the GC bias prevalent in some NGS techniques. The placement, insert and depth scores together

determine P(R—S). Independently, with only the assembly and not the reads, ALE calculates the k-mer sub-score and P(S). Each sub-score is calculated for each scaffold or contig within an assembly independently, allowing for variations commonly found in metagenomes. The four sub-scores are then combined to form the total ALE score. The constituent calculations in this pipeline are described in the Methods section.

In addition, these four sub-scores are reported by ALE as a function of position within the assembly, and can be visualized with the included plotting package or imported in table form to another package such as the Integrative Genomics Viewer (IGV) (Nicol et al. 2009), or the UCSC genome browser (Kent et al. 2002). When used in this way, these sub-scores can be used to locate specific errors in an assembly.

2.2 Probabilistic ingredients of the total ALE score

2.2.1 Placement sub-score

2.2.2 Insert sub-score

2.2.3 Depth sub-score

2.2.4 k-mer sub-score

2.3 Approximating Z

2.3.1 Aprroximating $Z_{placement}$

2.3.2 Aprroximating Z_{insert}

2.3.3 Aprroximating Z_{depth}

2.3.4 Aprroximating Z_{kmer}

2.4 Relationship of the difference of total ALE scores to probability of correctness

2.5 Thresholding the total ALE score

scores. We do this by averaging scores within windows, allowing for the discovery of large errors in the assembly while smoothing out the noise. By the Central Limit Theorem, when we average many independent and identically distributed random variables the result is approximately normally distributed. This allows us to create a threshold for which to delineate good scores from bad and pinpoint problematic regions. This is represented by a solid black line in the figures labeled 5σ. This line is calculated by assuming that the individual scores at each position in the assembly are drawn from a mixture of two normal distributions: one for high accuracy and another for low accuracy. We use maximum likelihood to determine the mean and variance of the two underlying distributions. The threshold is set as five standard deviations from the mean of the high accuracy distribution. This allows us to readily find areas of inaccuracy that are unlikely to be drawn from an accurate region. Five standard deviations corresponds to 1 false positive in 2 million positions if the joint normal distribution assumptions hold. The number of standard deviations at which the black line is drawn can be set from the command line by the user.

A black bar is drawn on the plot if the likelihood falls below the threshold at a significant fraction of the positions in any contiguous region with a given length (this fraction and length are user defined, and are initially set to 0.01% and 1000bp respectively) see figure XXX. These red bars correspond to regions of potential inaccuracy in the assembly that should be examined further. The plotter outputs these regions in a tab delineated text file for easy input into genome viewing software programs like IGV.

2.6 Influence of alignment input

ALE takes as input a proposed assembly and a SAM/BAM (Li 2009) file of alignments of reads onto this proposed assembly. This allows ALE to calculate the probability of observing the assembly given the reads. ALE assumes that this mapping will include, if not all possible mappings, at least the ”best” mapping for each read in the library (if such a mapping exists). For assemblies with many repeat regions (¿100) or libraries with large insert sizes, this can be difficult to obtain due to the bias introduced using default parameters of standard aligners. While an extensive review of alignment packages and their optimization is beyond the scope of this paper a review can be found in (Li and Homer 2010). If an assembly has many repeats and the aligner bias causes the reporting of reads only mapping to a fraction of possible regions, then ALE will see the unmapped regions as having 0 depth (no supporting reads) which will result in artificially low depth sub-scores. The robustness of ALE will still allow for comparison between assemblies with similar biases, but should be taken into account if the input to ALE is biased for only certain assemblies. To avoid this bias some mappers must be explicitly forced to search for all possible placements (-a in bowtie).

In summary, ALE determines the likelihood of an assembly given the reads and an accurate, unbiased alignment of those reads onto the assembly, without which the model assumptions are violated. These preconditions are usually met except for certain pathological genomes, and even in these cases can be readily corrected for by changing the parameters of the aligner used to make ALEs input.

CHAPTER 3

ALE RESULTS

CHAPTER 4

ALE IMPLEMENTATION

Part II

EPI: Expected Parallel Improvement

CHAPTER 5

EPI INTRODUCTION

CHAPTER 6

EPI METHODS

CHAPTER 7

EPI RESULTS

CHAPTER 8

EPI IMPLEMENTATION

Part III

Velvetrope

CHAPTER 9

VELVETROPE INTRODUCTION

CHAPTER 10

VELVETROPE METHODS

CHAPTER 11

VELVETROPE RESULTS

CHAPTER 12

VELVETROPE IMPLEMENTATION

APPENDIX A

CHAPTER 1 OF APPENDIX

Appendix chapter 1 text goes here