Scott Clark, CS6780 Final Report, *ALE: an Assembly Likelihood Evaluation Framework to Estimate the Quality of Metagenome Assemblies*

1 Introduction

DNA sequencing technology works by taking an organism (or many organisms in the metagenomic case) and extracting the entirety of the DNA into a test tube (on the order of 10^{10} base pairs (bps)). The DNA is then cut randomly into about equal pieces (we assume a normal distribution with known mean and variance for each experiment, average sizes are 200bps to 5000bps). These pieces of DNA are then put into a sequencer that can "read" the ends of these pieces (about 100bps). Thus leaving us with some information about the very ends of these pieces of DNA with some unknown insert length between them (drawn from a known distribution). The end result of this lab-work is many millions or billions of short, paired reads that can then be used to (attempt to) reassemble the entire genome of the organism (or organisms). This can be thought of as trying to reassemble many mixed jigsaw puzzles with missing, overlapping and duplicated pieces of variable sizes. There are many programs that can create these assemblies (Velvet, Abyss). However, there is no good metric for determining the quality of these assemblies, the current techniques either just use the overall size of the pieces outputted (N50: accuracy irrelevant) or try to map the suggested assembly onto a known reference, which is unknown in almost all cases. Another difficulty these programs have is "finishing' assemblies, going from large pieces of contiguous proposed assembly (contigs) and scaffolding them together and filling in the gaps to create a complete, finished assembly.

We want to be able to calculate the probability of seeing an assembly (set of seeds, contigs) $\mathcal{S}$, given a set of reads $\mathcal{R}$ from a set of libraries of reads $\mathcal{L}$, which have possibly different parameters for insert length and orientation.

$$P(\mathcal{S}^{(t)}|\mathcal{R}).$$

Basically, we want to be able to look at a set of contigs or an assembly and give it a "score" based off of all of the reads. This allows us to compare different assemblies. Once we can quickly compare two assemblies we theoretically have an implicit assembler by just maximizing this objective function in the space of all possible assemblies (very large). Start with some assembly $\mathcal{S}^{(0)}$, score it, then look at closely related assemblies and score all of them, choose the best one to be $\mathcal{S}^{(1)}$, repeat until some threshold is met. On it's own, and what we will present is a way of determining the quality of a given assembly using a rigorous mathematical framework, something that the field lacks, especially in the global and metagenomic setting.

There has been some previous work in this field, most notably by [Phillippy *et al.*, 2008] who describes a single genome assembly validation method that uses a basic statistical framework including the requirement of overlapping reads to agree, insert length needed to be within a threshold of the mean, and most reads must map onto the assembly. Additionally they require that when reads are mapped onto the assembly their coverage, or depth, must behave like a Poisson distribution, which is consistent with a random shearing process, shown by [Lander and Waterman, 2008]. Unfortunately they require all reads to be oriented inward, a constraint not followed by the data and they assume a single genome (to prevent reads being able to map to other similar genomes and resolving repeats). Their statistical rigor is also lacking as they primarily

use thresholds and never build an overall likelihood of a given assembly. They use this method to build scaffolds as opposed to score already formed assemblies which is another deviation from our current work. This is similar to the work of [Olsen, 2009] who combines the methods of [Choi, 2008] who uses machine learning to find sequence errors based on some non-physical features and a questionable dataset to compare to references and [Zimin, 2008] who combines different assemblies to create a single, "better", one and introduces a metric (CE statistic) for finding repeat regions that are underrepresented in the assembly. [Dayarian, 2010] developed a method for building scaffolds via some statistical optimization but their feature space is limited, their code is closed source, their exact method is not reported and they make a single genome assumption. [Kelly, 2010] and [Meader, 2010] independently developed methods that compare multiple references to a potential assembly and then creates a hybrid assembly. Overall, there is a wide breadth of work in the area, but most methods lack mathematical rigor and depend on looking at small regions of a single genome, usually requiring a reference. The field lacks a mathematically rigorous way of comparing different assemblies and for validating metagenomic assemblies in general.

2 Background

Each read has the following properties,

Read R_k Properties

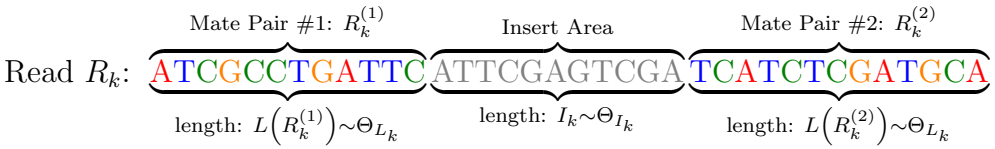


Figure 1: Every read $R_k \in \mathcal{R}$ is composed of two mate pairs $R_k^{(1)}$ and $R_k^{(2)}$ with corresponding lengths $L\left(R_k^{(1)}\right)$ and $L\left(R_k^{(2)}\right)$ drawn from a distribution Θ_{L_k}. These mate pairs are separated by a distance I_k drawn from a distribution Θ_{I_k}.

Read Orientation ω_k

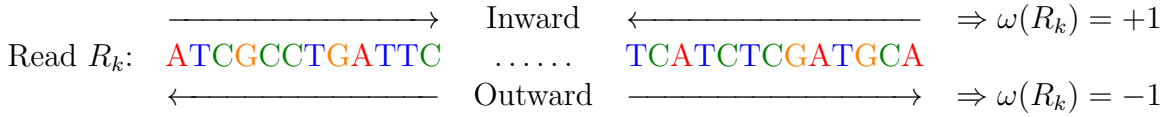
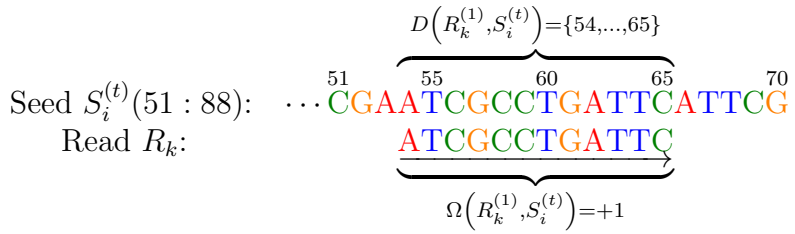


Figure 2: Every read has a unique orientation based on how the enzymes in the sequencer bind to it. A read can be sequenced inward, towards the middle, which would imply $\omega(R_k) = +1$ in our notation. Alternately a read can be sequenced outward, from the middle to the ends, which would imply $\omega(R_k) = -1$.

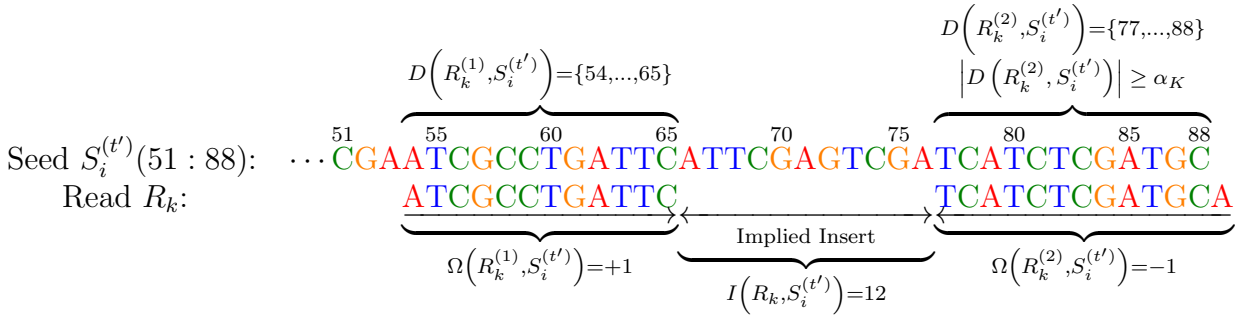
Each read can map onto the seed (assembly) in the following ways,

Read R_k *Anchored* to Seed $S_i^{(t)}$



$$I\left(R_k, S_i^{(t)}\right) > 5 \Big| \omega_k = +1, \quad I\left(R_k, S_i^{(t)}\right) > 53 \Big| \omega_k = -1$$

Read R_k *Doubly Anchored* to Seed $S_i^{(t')}$ at some later time t'



$$O\left(R_k, S_i^{(t')}\right) = \{1, 2\}, \quad \omega\left(R_k, S_i^{(t')}\right) = +1$$

Figure 3: A read R_k can be either *anchored* or *doubly anchored* to a seed $S_i^{(t)}$ if one or both of it's corresponding mate pairs match some subset of $S_i^{(t)}$ of size at least α_k exactly. If a read is singly anchored we can obtain the implied orientation with respect to the seed and bound the minimum insert size I_k conditioned on the reads overall orientation ω_k. If the read is doubly anchored we can directly calculate the implied total orientation ω_k, insert length I_k and ordering $O\left(R_k, S_i^{(t')}\right)$.

3 Likelihood Framework

The Likelihood needs to take at least the following into account:

1. Read **sequences** need to agree with assembly.
    - If a read is drawn from a contig then the individual bases must agree, at least with a probability greater than some threshold.
2. **Insert lengths** must be consistent with library.
    - Once we determine probabilistically where on an assembly the reads were drawn we can compare their insert lengths with the known distribution.
3. **Orientation** of mated reads needs to be consistent with library.
    - Just like with insert lengths, once the reads are mapped we can compare orientations to the Bernoulli distribution from which they were drawn,
4. **Coverage** Poisson distributed over entire assembly.
    - If we assume the libraries are generated from randomly sliced DNA then when all of the reads are mapped back onto the assembly the depth at which every base pair is represented (number of reads) should be Poisson distributed.
5. k**-mer frequency** of the contig.
    - Every genome has a unique vector corresponding to the frequency of the possible 4^k k-mers (k contiguous base pairs) that make up the entire genome. Every k-mer within an assembly should be consistent with its respective genomes frequency.

3.1 Sequences

For each mate pair part p of each read r_k the likelihood that it was drawn from an assembly seed S_i at offset Ω is related to the "quality score" at that position in the read. The quality score is the probability that the sequencer reported the correct base at that position as is an output of the experiment. $Q_{r_k}^{(p)}(j)$ is the probability that $r_k^{(p)}(j)$ was reported correctly, ie if it says it is an 'A', then it is an 'A' with this probability. The likelihood then is related to

$$P_\Omega(j) = P\left(\text{observing } r_k^{(p)}(j) | r_k^{(p)} \text{ maps to } S_i \text{ with offset } \Omega\right) = \begin{cases} Q_{r_k}^{(p)}(j) & \text{if } r_k^{(p)}(j) = S_i(j+\Omega) \\ \frac{1-Q_{r_k}^{(p)}(j)}{3} & \text{otherwise} \end{cases}$$

So the overall likelihood that this part of this read maps onto this seed with this offset is $\prod_{j=1}^{L_{r_k^{(p)}}} p(j)$. This can be done for both parts of the reads (with all combinations of offsets) to determine the probability that those parts map onto the seed in those positions taking only the sequence into account.

3.2 Insert lengths and orientation

Once we know where the read parts are potentially mapping the insert length and orientation are determined and we can find their likelihoods by explicitly finding how likely they are from their given distributions for each library.

3.3 Coverage

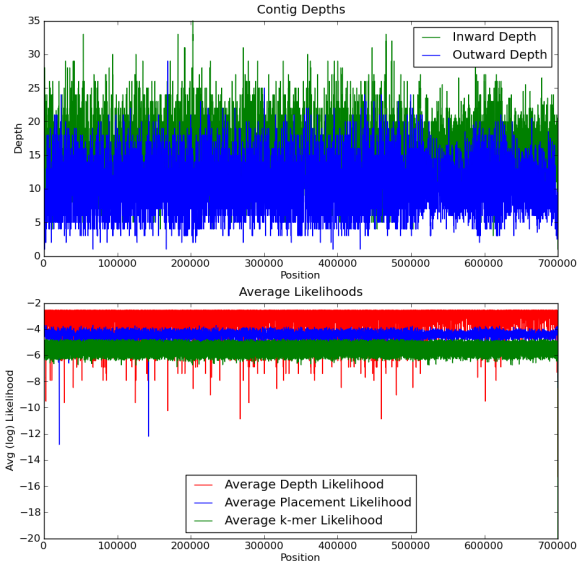
Once we tentatively place all of the reads onto all of the assembly pieces (with each placement given an explicit likelihood) we can find the expected coverage (depth, how many reads correspond to each position of the sequence) and the percentage of reads that map to some part of the assembly with a likelihood above a certain threshold. We know the coverage should be Poisson distributed with a common mean and most of the reads should map somewhere (for a good assembly).

3.4 k-mer frequency

Every genome has a unique k-mer frequency. This is the frequency of any set of k base pairs appearing in order in the genome. This 4^k dimensional vector is conserved across a genome and can be used to help determine if two different genomes have been mistakenly added together. We calculate the probability based on assuming a uniform Dirichlet prior on the vector space and explicitly comparing the actual frequency with what k-mers are observed at each position in a contig/assembly.

3.5 Depths and individual metrics

Below is an example of all of the metrics for a subset of a known genome (the first 700,000 base pairs of E.Coli) 100,000 randomly generated reads from it (using lab inspired distributions for the library).



3.6 Putting it all together

When we take all of these likelihoods together we come up with the likelihood that we described in the introduction,

$$P(\mathcal{S}^{(t)}|\mathcal{R}) \propto \frac{1}{|S|}\sum_{i=1}^{|S|}\Bigg[\underbrace{\log\left(\frac{1}{\sum_{j=1}^{|R_{S(i)}|}P(R_{S(i)})}\sum_{j=1}^{|R_{S(i)}|}P(R_{S(i)})P_{\text{placement}}(R_{S(i)}(j)\rightarrow S(i))\right)}_{\text{Placement score}}+$$

$$\underbrace{\log\left(P_{\text{depth}}\left(\left(\sum_{j=1}^{|R_{S(i)}|}P(R_{S(i)})\right)\sim\text{Poisson}\left(E_i\left[\sum_{j=1}^{|R_{S(i)}|}P(R_{S(i)})\right]\right)\right)\right)}_{\text{Depth score}}+$$

$$\underbrace{\log\left(\frac{1}{|kmer(S(i))|}\sum_{j=1}^{|kmer(S(i))|}P_{\text{kmer}}(kmer_j(S(i))\sim kmer(S))\right)}_{\text{k-mer score}}\Bigg]$$

4 Results

4.1 Score deterioration with permutation of known assembly

If we start with a known genome (the first 700,000 base pairs of E.Coli) and randomly generate 100,000 reads from it and then see how the metrics perform we expect everything to behave well, with occasional dips and rises in the likelihood.

If we then replace a string of 70 base pairs from the known genome near the middle (265350-265420) of the genome with either all A's or all N's (unknowns) then the metrics drop significantly in the local and global scores, as well as visually.

	Global Score	Local Score (265350-265420)
Perfect	-12.8364	-12.8178
A's	-12.8407	-13.5256
N's	-12.8424	-13.8076

By permuting the known genome the local score and global score both drop and we can visually see all metrics drop in the permuted region.

4.2 Likelihood as a function of errors

We would expect this likelihood function to increase with assembly quality. To test this we synthesis 4 million reads with known parameter distributions from a known genome (E. Coli). We then see what the likelihood of observing the true assembly is compared to other permuted

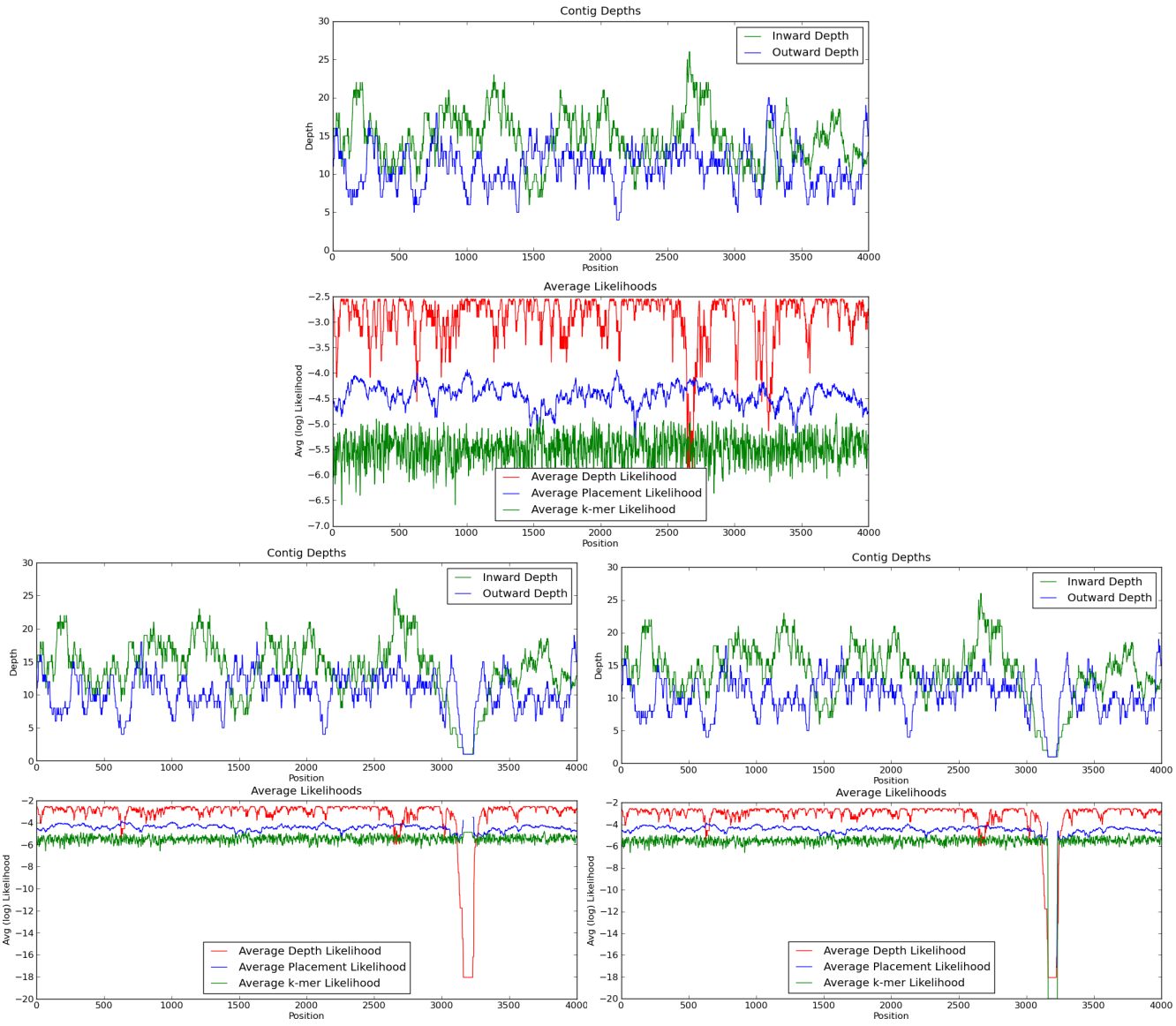
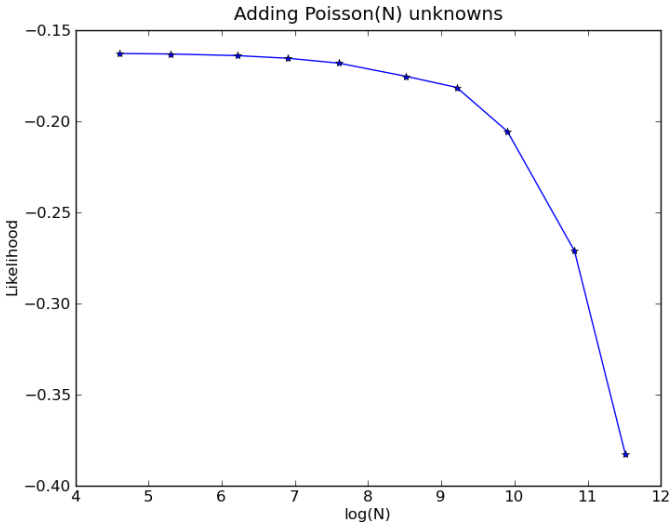


Figure 4: Here we look at the various statistics (depth inward and outward) as well as the placement, kmer and depth likelihoods for a region (bp 265350-265420) of E.Coli where we take the true sequence as the assembly, replace a section with all errors (As), and replace a section with unknowns (counter clockwise from top). As we can see the score greatly deteriorates within the perturbed region. We will show that the degradation in score is a function of the extent of the permutation, with no permutation being the best.

versions of that assembly (first replacing large parts with N unknowns, then by breaking up the assembly with N overlap). We note that the (normalized, relative to perfect) likelihood decreases with the permutation.



5 Conclusion

We were able to come up with a probabilistically motivated likelihood function for genome assemblies. This allows for assemblies to be independently and accurately scored for the first time, opening up a myriad of applications from genome finishing to assembly as global optimization.

I plan to write this up as a paper in Genome Research with my advisors and release the 4000+ lines of C/python as open source in hopes that this will catch on in the community.

References

[Phillippy *et al.*, 2008] Adam Phillippy, Michael Schatz, Mihai Pop (2008) Genome assembly forensics: finding the elusive mis-assembly. *Genome Biology* **9**(3), R55.

[Lander and Waterman, 2008] E.S. Lander, M.S. Waterman (1988) Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics*, **2**(3), pp 231-239.

[Choi, 2008] J.H. Choi *et al.* (2008) A Machine Learning Approach to Combined Evidence Validation of Genome Assemblies. *BMC Bioinformatics*, **24**(6), pp 744-750.

[Zimin, 2008] A.V. Zimin *et al.* (2008) Assembly Reconciliation. *BMC Bioinformatics*, **24**(1), pp 42-45.

[Olsen, 2009] M.R. Olsen (2009) New Methods for Assembly and Validation of Large Genomes. *Master's Thesis, Notre Dame.*

[Dayarian, 2010] A. Dayarian *et al.* (2010) SOPRA: Scaffolding algorithm for paired reads via statistical optimization. *BMC Bioinformatics*, **11**(1), 345.

[Kelly, 2010] K.A. Kelly (2010) Genome Segmentation From High-Throughput Sequencing Data. Sumbitted 2010.

[Meader, 2010] S. Meader (2010) Genome assembly quality: Assessment and improvement using the neutral indel model. *Genome Research* **20**(5), pp 675-684