# ATB: Supplementary Materials

This document is largely copy-pasted from the same document written for the *Streptococcus pyogenes* strain 5005 experiment[1]. I have so far just removed details specific to that work and filled in details relevant to group B Streptococcus and relevant to changes which have occurred in my workflow since then.

## Sequence Filtering and Alignments

### Scripts used

Italicized directory names (ending in /) refer to directories within the git repository for this project. Many of the post-processing is handled by the hpgltools[2] R package, functions used are italicized and suffixed with ().

### Annotation Collection

Three *Streptococcus agalactiae* strains were used as the reference for this: a909, cjb111, and 2603vr. The genomes and gff annotations are in the *reference/* directory and are accessions: NC\_007432, AAJQ01000001, and NC\_004116 respectively. As of the writing of this document, most analyses were performed using a909 as the reference.

### Genome Alignments and Essentiality

The approximately 2.5 million reads of each raw library were queried for quality with Fastqc[3] before removing the *mariner* ITR leading sequences with cutadapt[4]. These libraries were aligned against the genomes with bowtie[5,6] using options to allow one mismatch (-v 1) and randomly assign multi-matched reads to one of the possible matching positions (-M 1). The resulting alignments were converted to sorted/compressed binary alignments[7] and counted[8] against the reference genome CDS and intergenic regions.

The essentiality software package[9] provides an opportunity to query statistically significant stretches of TAs which have no observed insertions to further inform its metric of essentiality. The insertion data was therefore converted into its expected format and passed to the version 1.21 of the implementation python script. The resulting table provided a count of the number of insertions observed in each ORF, the number of observed TAs, the maximum length of non-observed sequence, the nucleotide span of this region, a call on whether each ORF is essential, and the posterior probability for each call. The default options were used except multiple runs were performed with the minimum hit parameter set to: 1,2, 4, 8, 16, 32. These operations were performed via CYOA[10]. In a separate invocation, the three replicates for the control, low concentration, and high concentration samples were concatenated into a single sample and essentiality was run on the combined samples.

## Library Metrics

The libraries were quantified with respect to relative coverage, similarity, and saturation with respect to available TA insertion points. These tasks were performed using the hpgltools and the input text/wig files for the essentiality package. Thus, the essentiality input files were read into the R function *plot\_saturation()* and used to visualize the saturation of each library. This was done by taking the log2(hits + 1) for each position and plotting them as a set of histograms.

### Normalization and visualization strategies

The most expedient method of comparing the libraries was to treat them as if they were components of an RNA sequencing experiment and assuming similar normalization strategies[11] apply. This strategy is very similar to that taken by the essentials software package[12], but uses a combination of voom/limma, EdgeR, DESeq2, EBSeq, and a statistically uninformed basic analysis instead of EdgeR. Pairwise Euclidean distances, Spearman correlation coefficients, and principle component analyses were then used to visualize the similarities/differences between normalized libraries.

# References

1. Le Breton Y, Belew AT, Valdes KM, Islam E, Curry P, Tettelin H, et al. Essential genes in the core genome of the human pathogen streptococcus pyogenes. Sci Rep. 2015;5. doi:10.1038/srep09838

2. hpgltools [Internet]. Available: https://github.com/elsayed-lab/hpgltools

3. Andrews S. FASTQC. A quality control tool for high throughput sequence data. In: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/. 2010.

4. Martin M. Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet J. 2011; 10–12.

5. Langmead B. Aligning short sequencing reads with Bowtie. Curr Protoc Bioinforma. 2010; 1–24. doi:10.1002/0471250953.bi1107s32.Aligning

6. Langmead B, Trapnell C, Pop M, Salzberg SL. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol. 2009;10: R25. doi:10.1186/gb-2009-10-3-r25

7. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009;25: 2078–9. doi:10.1093/bioinformatics/btp352

8. Anders S. HTSeq: Analysing high-throughput sequencing data with Python. In: … . embl. de/users/anders/HTSeq/doc/overview. html. 2010 pp. 1–28.

9. DeJesus M a, Zhang YJ, Sassetti CM, Rubin EJ, Sacchettini JC, Ioerger TR. Bayesian analysis of gene essentiality based on sequencing of transposon insertion libraries. Bioinformatics. 2013;29: 695–703. doi:10.1093/bioinformatics/btt043

10. github.com/abelew/cyoa [Internet].

11. Dillies M-A, Rau A, Aubert J, Hennequet-Antier C, Jeanmougin M, Servant N, et al. A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. Brief Bioinform. 2013;14: 671–83. doi:10.1093/bib/bbs046

12. Zomer A, Burghout P, Bootsma HJ, Hermans PWM, van Hijum S a FT. ESSENTIALS: software for rapid analysis of high throughput transposon insertion sequencing data. PLoS One. 2012;7: e43012. doi:10.1371/journal.pone.0043012