**We’ll Assemble It for You Wholesale: A Report on the *De Novo* Genome Assembly of *Candidatus* Sulcia muelleriEndosymbiont of Plant Pest *Kolla Palula***

Copeland HC, Lynch BG, Pollio AR, Spencer NJ

***1. Response to Reviewers***

*1.1 Background and Significance*

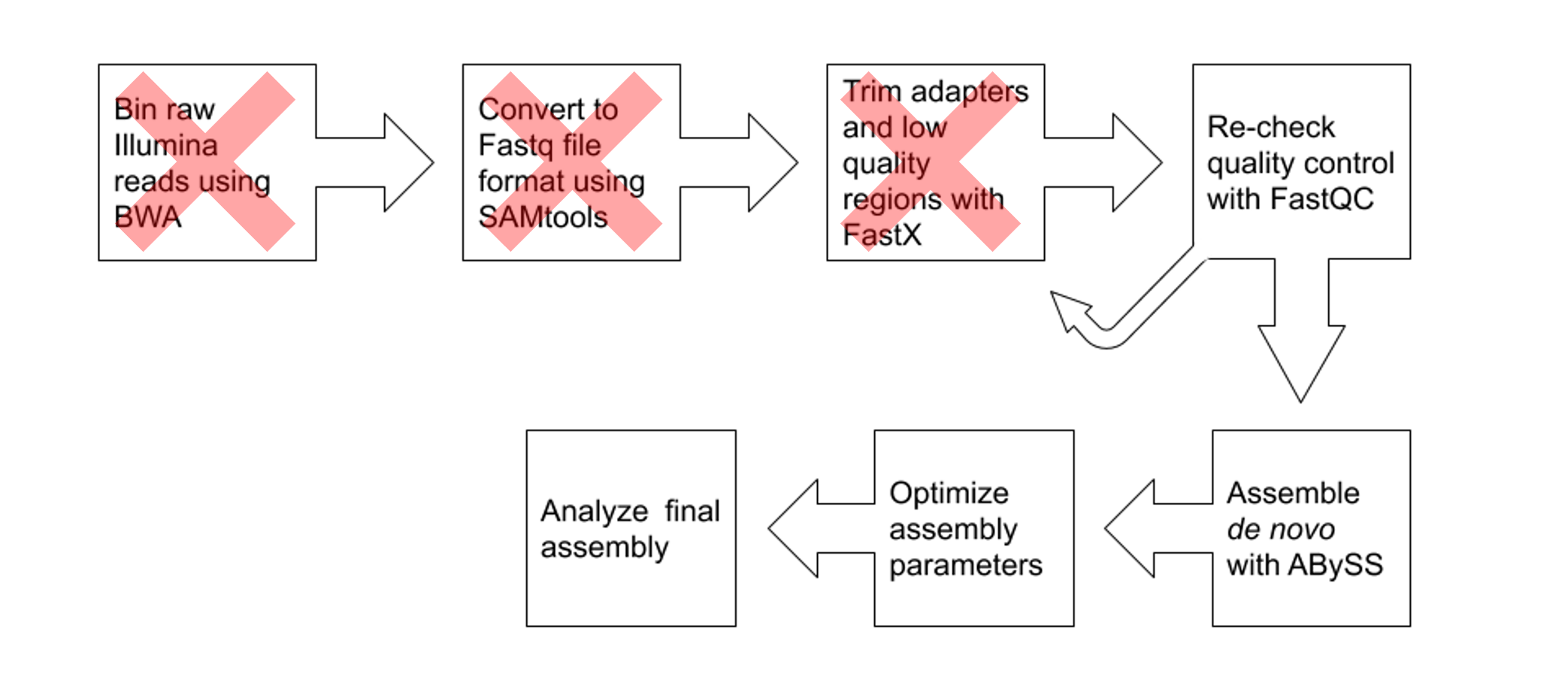
Based on the reviewer questions and comments, we would like to clarify that *Candidatus* Sulcia muelleri (hereafter *Sulcia*) is a symbiotic bacterium that provides key nutritional supplementation to sap-feeding insect hosts like the sharpshooter leafhopper *Kolla palula* (1,2). Their long coevolutionary history has led to close physiological integration, with the symbiont serving multiple key functional roles and the host displaying specific adaptations to facilitate this relationship (such as the development of specialized tissues for housing symbiont populations).

*K. palula*, as a sap-feeding organism, interfaces directly with plants and can serve as a vector for plant pathogens. In particular, *K. palula* is an effective vector of the plant pathogen *Xyella fastidiosa*, the causative agent of Pierce’s disease in grapevines (3). This disease and its transmission by these leafhoppers has caused serious damage to vineyards in Taiwan.

In response to comments regarding the clarity of our objective, we would like to emphasize that, because of the close relationship that this leafhopper has with its symbiotic *Sulcia* and the numerous impacts of this microbe on its host, the genome of this symbiont could reveal metabolic or immunological differences between this leafhopper-*Sulcia* partnership and those associated with less effective plant pathogen vectors. In essence, we are using the endogenous microbiota of this leafhopper to assess how it may influence the leafhopper’s interactions with more transient microbes that it passes on to its grapevine food source.

*1.2 Pipeline*

After reviewing the reviewers’ suggestions, we determined that FastQC quality control analyses of the raw reads prior to binning by organism with BWA-mem would be appropriate and updated our pipeline accordingly. However, our findings during the assembly process determined that binning the reads was unnecessary altogether, and trimming was also deemed unnecessary. Both of these changes and the reasoning behind them are outlined in detail in our results. Figure 1 shows an updated pipeline that reflects how we arrived at our final assembly.



**Figure 1. Updated pipeline.** Pipeline steps marked with a red “X” were omitted from the final assembly process.

***2. Results***

*2.1 Data Retrieval and File Format Conversion with SRA Toolkit*

Raw sequencing data for *Sulcia muelleri* from *Kolla palula* used for the assembly reported by Shih et al. was successfully obtained from the Sequence Read Archive (SRA) (4). The Illumina MiSeq data consisted of 54,400,000 spots, each consisting of two single-end reads for a total of 15.3Mb of sequence. The reads were obtained in an SRA-specific file format, and the spots had not yet been separated into individual reads. To generate a usable raw sequence file, the acquired data file was converted to a fastq file format with spots separated into reads using SRA Toolkit.

2.2 *FastQC Quality Control Analysis*

Raw read quality was verified using FastQC. The resulting output showed overall high quality (average Phred scores for all nucleotide positions > 32) with no reads below our trimming threshold (Phred 28). No overrepresented sequences or other quality issues were flagged by the program. Even though 15.3Mb of sequence data would provide more than sufficient coverage for our expected genome size, we chose not to trim the raw reads based on the consistent high quality across each read.

*2.3 Read Sorting with BWA-mem*

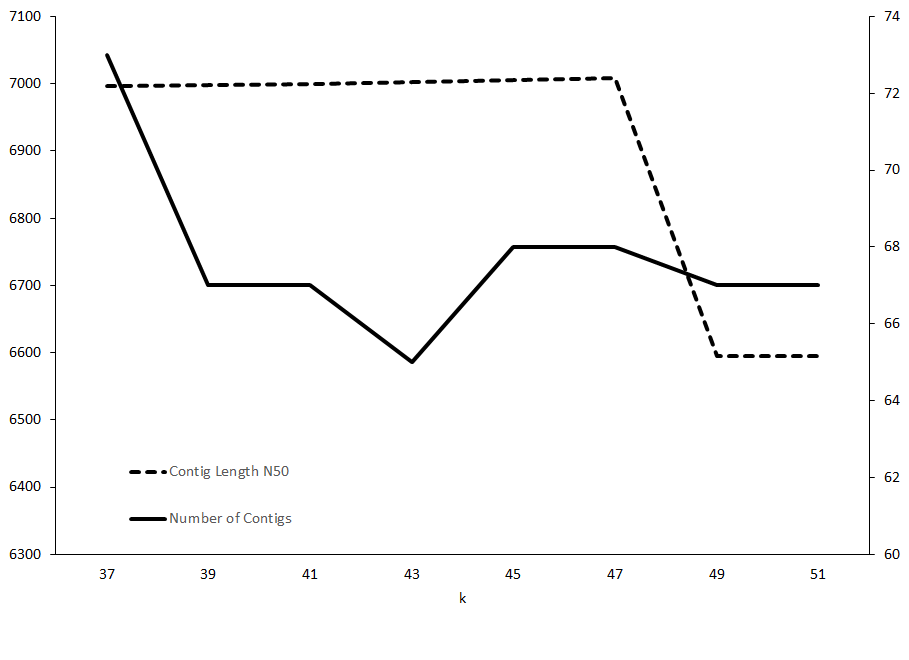
Reads were binned using BWA-mem based on their mapping to a reference *Sulcia* genome (5), and the mapped and unmapped reads from the resulting SAM file were converted to fastq files using SAMTools. Because 79601 of the 101043 sequences mapped to *Sulcia* and because FastQC analysis showed little variation in the %GC of the reads (despite *Sulcia*’s higher AT-richness relative to its host), flagged and unflagged reads were both assembled and the resulting contigs BLASTed against the nr\_refseq database to verify the mapping. The resulting mapped and unmapped reads all aligned with *Sulcia* suggesting that the data had already been enriched for *Sulcia* or that host reads had been filtered out before distribution on the SRA, despite the sample being listed as metagenomic in nature. As a result, the original reads were used for the final genome assembly.

*2.4 Final Assembly with ABySS*

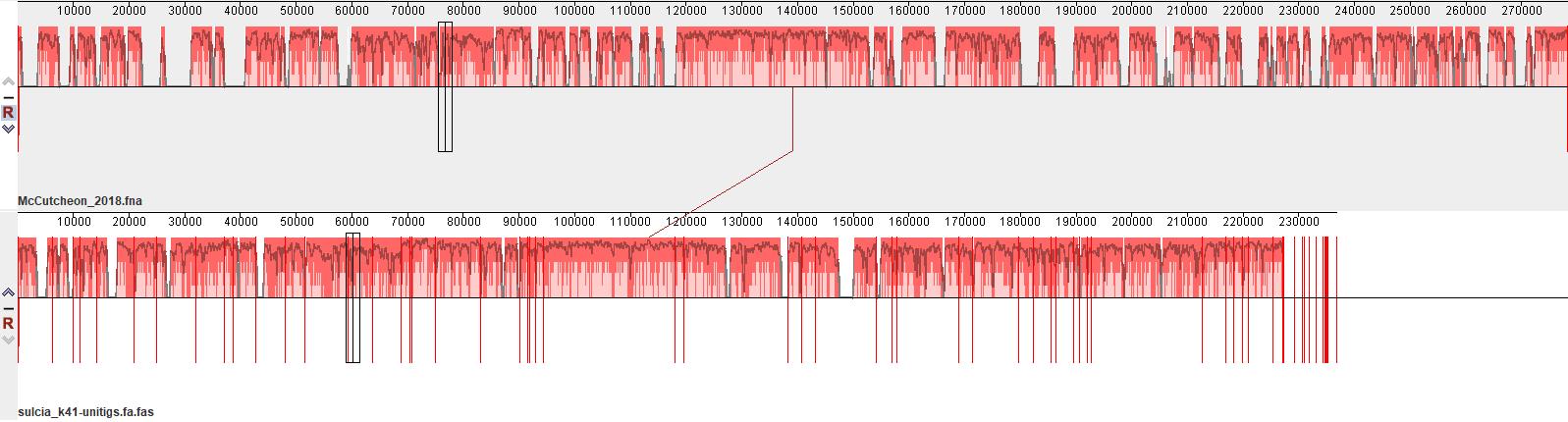
Once it was determined that the original sequence data was made up of *Sulcia* sequence, it was used for the final assembly in ABySS. The small genome size and single-end reads led to consistently rapid assemblies, so we were able to run several assemblies sequentially on the Spruce cluster with different values of k to optimize the assembly process. Initially, odd k values at intervals as large as ten were used to narrow down an optimal range based on contig length N50 (the minimum length met by at least 50% of the assembled contigs) and the total number of contigs generated. Ultimately, k values in the range of 37 to 51 produced the optimal values for these statistics, as well as for total length of contigs and length of longest contig. Figure 2 shows that these values begin to stray from the optimal values as k values increase or decrease past this range.

Overall, the assemblies led to a relatively complete picture of the target genome with the maximum sum of contig lengths being 234.4 kb, 92.3% of the total genome size reported from the original assembly with these reads (4). There were, however, obvious gaps in our assembly. Our smallest number of contigs from an assembly was 65, indicating at least 64 regions where no overlapping sequence connected the contigs together.

To visually assess these gaps and the quality of our assembly, we used Mauve to map the assembled contigs to the reference *Sulcia* genome initially used for BWA-mem (5). This alignment shows that our assembly resemble previous *de novo* assemblies of *Sulcia* endosymbionts, and information from reference genomes could help us determine sequences like repeat regions that could be responsible for the gaps observed in our assembly (Figure 3).



**Figure 2. Assembly statistics by k value.** Contig length N50 (left y-axis) and number of contigs (right y-axis) are shown for selected k values. Assemblies with k values at larger intervals identified this range as optimal based on these two assembly statistics. Notably, k=43 produces the minimum number contigs at 65 while k=47 produced the maximum contig length N50 of 7008 bp.

**

**Figure 3. Mauve alignment of contigs to *Sulcia muelleri* reference genome.** Our assembled *Sulcia* contigs (bottom) are shown aligned against an assembled *Suclia muelleri* reference genome from a publication by Matsuura et al. (2018).

***3. Discussion***

As outlined in our proposal of this assembly, we did not anticipate obtaining closed genome because of the nature of *de novo* assembly of short reads and the characteristics of most genomes, as well as time constraints on the project. As such, our assembly of roughly 92% of the target genome into about 65 contigs aligns with our predicted results. While our end result was a satisfactory attempt at the assembly, there were several obstacles to obtaining our final results. For instance, during the initial data retrieval, the spots were not originally separated into individual reads, leading to confusing results during quality control when each read appeared to be twice the expected length. Thankfully, this was quickly identified and remedied by quality control, illustrating the importance of quality control checking early in the process.

Later issues would also hinge on quality control assessments. During the initial mapping of the reads, we neglected to separate out mapped and unmapped reads. Once again, a quality control check revealed the error once it became clear that mapping was having no impact on the assembly, as the total number of sequences had not decreased following mapping.

Once the reads were finally sorted based on mapping with BWA-mem, even further issues became evident. While quality control data did indicate that the mapped reads were fewer in number than in the raw data set, assemblies with the resulting reads showed worse results than assemblies conducted with the unmapped reads. This could be attributed to chimeric contigs containing host sequence in the previous assembly, but we were persuaded to check our mapping results again by the GC content distribution indicated in the original FastQC report, which was more uniform than would be expected of a metagenomic data set. Assembly and BLAST searches of both mapped and unmapped reads revealed that both were made up of *Sulcia* sequence, meaning that the original data did not contain a mixture of sequences, and any reads that didn’t map were likely the result of variation between the target genome and the reference used for mapping.

Ultimately, these challenges have conferred one major lesson, which is to check the data frequently and critically. Some of our issues were solved by simply performing quality control analysis, but our final problem was not indicated by any warning messages or out-of-range values. Going forward, we will emphasize a holistic look at quality control reports and other metadata and compare them with our approach and assumptions about the data rather than simply looking for things that are obviously “wrong.” Part of the process of bioinformatics, as we have come to realize, is being able to understand your data and the context for it enough to accurately apply computational tools, which possess only enough information to do their jobs, to your research goals.

***4. Pipeline Code***

*4.1 Data File Format Conversion with SRA Toolkit*

wget "ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sdk/current/sratoolkit.curr

ent-centos\_linux64.tar.gz"

tar -xzf sratoolkit.current-centos\_linux64.tar.gz

~/sratoolkit.2.9.6-1-centos\_linux64/bin/fastq-dump --split-spot SRR8245541.1 sulcia\_reads.fastq

*4.2 Quality Control Checking with FastQC*

Script submitted to spruce using qsub fastqc.sh:

#PBS -q debug

#PBS -lwalltime=00:05:00

#PBS -nodes=1:ppn=12

#PBS -N fastqc

#PBS -m arp -M njs0012@mix.wvu.edu

module load genomics/bioconda

source /shared/software/miniconda3/etc/profile.d/conda.sh

conda activate tpd0001

cd $SCRATCH

fastqc sulcia\_reads.fastq

conda deactivate

*4.3 Read Binning with BWA-mem and File Conversion with SAMtools*

Script submitted to spruce using qsub bwa.sh:

#PBS -q debug

#PBS -lwalltime=00:05:00

#PBS -nodes=1:ppn=12

#PBS -N bwa

#PBS -m arp -M njs0012@mix.wvu.edu

module load genomics/bioconda

source /shared/software/miniconda3/etc/profile.d/conda.sh

conda activate tpd0001

cd $SCRATCH

bwa index reference.fa

bwa mem reference.fa sulcia\_reads.fastq > sorted.sam

samtools view -S -b -F4 sorted.sam > mapped.bam

samtools bam2fq mapped.bam > Sulcia\_reads\_mapped.fastq

samtools view -S -b -f4 sorted.sam > unmapped.bam

samtools bam2fq unmapped.bam > Sulcia\_reads\_unmapped.fastq

conda deactivate

*4.5 De novo Assembly with ABySS*

Script submitted to spruce using qsub abyss.sh:

#PBS -q comm\_mmem\_day

#PBS -lwalltime=24:00:00

#PBS -nodes=1:ppn=12

#PBS -N abyss

#PBS -m arp -M njs0012@mix.wvu.edu

module load genomics/bioconda

source /shared/software/miniconda3/etc/profile.d/conda.sh

conda activate tpd0001

cd $SCRATCH

abyss-pe k=19 name=sulcia\_k19 se="sulcia\_reads.fastq"

abyss-pe k=21 name=sulcia\_k21 se="sulcia\_reads.fastq"

abyss-pe k=27 name=sulcia\_k27 se="sulcia\_reads.fastq"

abyss-pe k=31 name=sulcia\_k31 se="sulcia\_reads.fastq"

abyss-pe k=33 name=sulcia\_k33 se="sulcia\_reads.fastq"

abyss-pe k=35 name=sulcia\_k35 se="sulcia\_reads.fastq"

abyss-pe k=37 name=sulcia\_k37 se="sulcia\_reads.fastq"

abyss-pe k=39 name=sulcia\_k39 se="sulcia\_reads.fastq"

abyss-pe k=41 name=sulcia\_k41 se="sulcia\_reads.fastq"

abyss-pe k=43 name=sulcia\_k43 se="sulcia\_reads.fastq"

abyss-pe k=45 name=sulcia\_k45 se="sulcia\_reads.fastq"

abyss-pe k=47 name=sulcia\_k47 se="sulcia\_reads.fastq"

abyss-pe k=49 name=sulcia\_k49 se="sulcia\_reads.fastq"

abyss-pe k=51 name=sulcia\_k51 se="sulcia\_reads.fastq"

abyss-pe k=61 name=sulcia\_k61 se="sulcia\_reads.fastq"

abyss-pe k=71 name=sulcia\_k71 se="sulcia\_reads.fastq"

abyss-pe k=81 name=sulcia\_k81 se="sulcia\_reads.fastq"

abyss-pe k=91 name=sulcia\_k91 se="sulcia\_reads.fastq"

abyss-pe k=101 name=sulcia\_k101 se="sulcia\_reads.fastq"

conda deactivate

***References***

1. J. P. McCutcheon, B. R. McDonald, N. A. Moran, Convergent evolution of metabolic roles in bacterial co-symbionts of insects. PNAS. 106, 15394–15399 (2009).

2. G. M. Bennett, N. A. Moran, Small, Smaller, Smallest: The Origins and Evolution of Ancient Dual Symbioses in a Phloem-Feeding Insect. Genome Biol Evol. 5, 1675–1688 (2013).

3. S.-J. Tuan, F.-T. Hu, H.-Y. Chang, P.-W. Chang, Y.-H. Chen, T.-P. Huang, Xylella fastidiosa Transmission and Life History of Two Cicadellinae Sharpshooters, Kolla paulula and Bothrogonia ferruginea (Hemiptera: Cicadellidae), in Taiwan. J Econ Entomol. 109, 1034–1040 (2016).

4. H.-T. Shih, C.-C. Su, C.-J. Chang, S. Vargas, Z. Dai, J. Chen, Draft Genome Sequence of “Candidatus Sulcia muelleri” Strain KPTW1 from Kolla paulula, a Vector of Xylella fastidiosa Causing Pierce’s Disease of Grapevine in Taiwan. Microbiol Resour Announc. 8, e01347-18 (2019).

5. Y. Matsuura, M. Moriyama, P. Łukasik, D. Vanderpool, M. Tanahashi, X.-Y. Meng, J. P. McCutcheon, T. Fukatsu, Recurrent symbiont recruitment from fungal parasites in cicadas. Proc Natl Acad Sci U S A. 115, E5970–E5979 (2018).