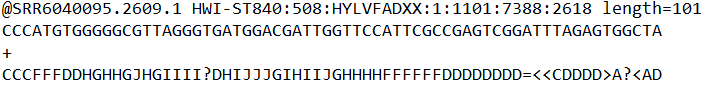


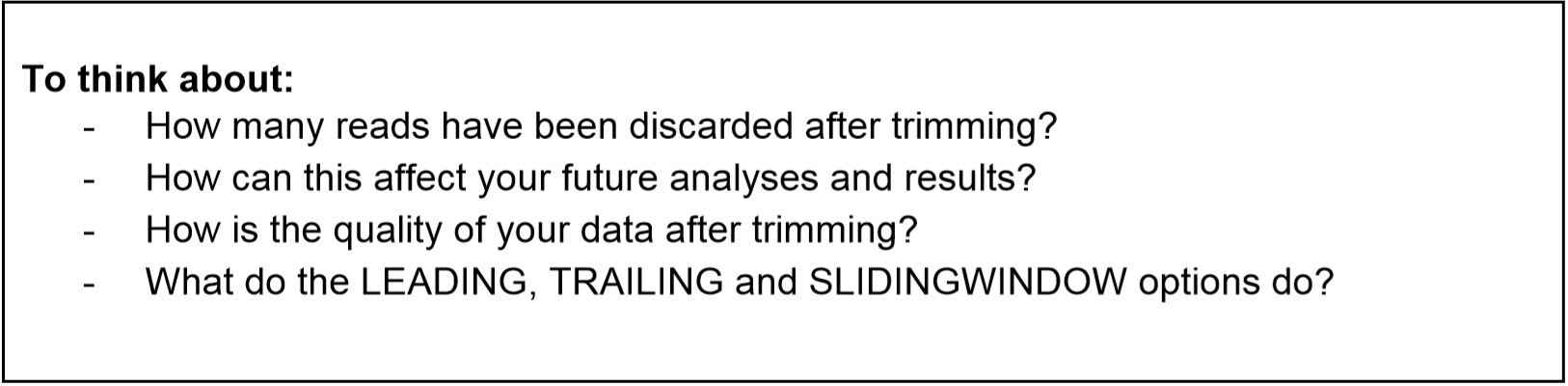
Take an example of my trimmed RNA-seq Illumina:



There are 4 lines. Line 1 begins with '@' followed by a sequence identifier and optional descriptions. Line 2 is the raw sequence letters. Line 3 begins with '+', can be empty or followed by the same sequence identifier and description again. Line 4 shows the quality, each symbol represents the quality of corresponding base, in my case the data is in medium quality.

The data before trimming is from NCBI SRA, after trimming it’s still in the same format. But it lost information about paired-end due to fastq-dump. In other cases, /1 (paired) or /2 (not paired), or just 1:Y, 2:N, 2:Y.. in line 1 shows the information.

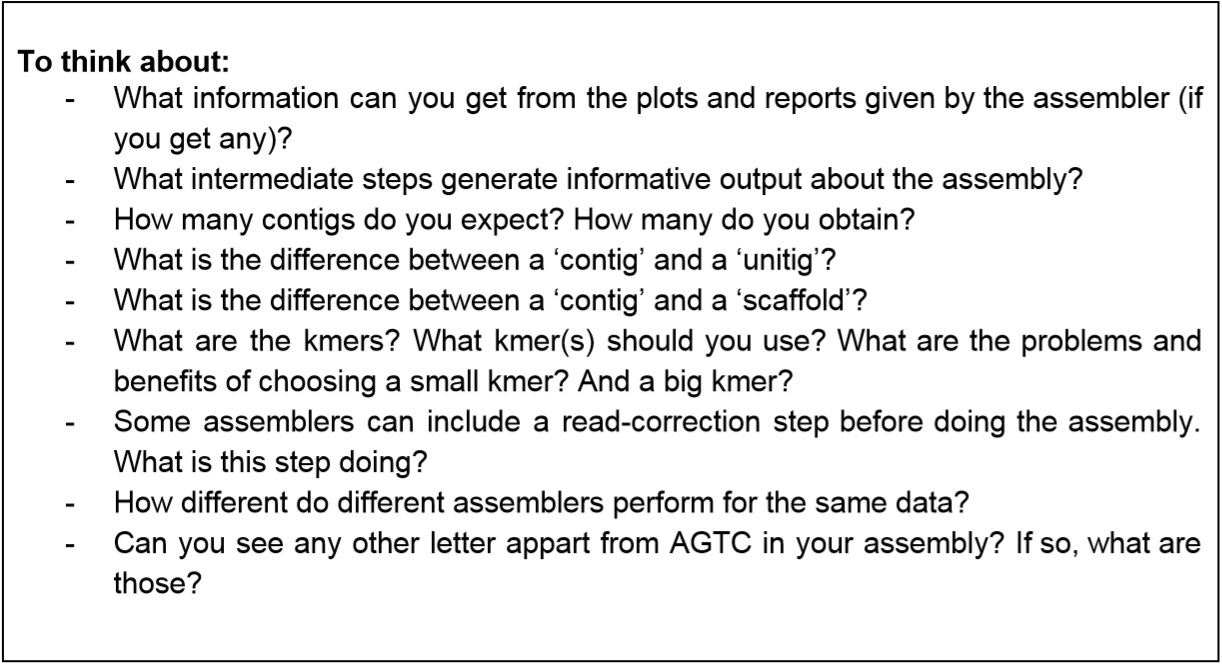
According to the FastQC report, there could be some issues. My data is trimmed, so Per base sequence quality, Per Sequence Quality Scores and Per base N content are nice. But content distribution (Per Base Sequence Content, Per Base GC Content) turns ‘Red’, which could be caused by overrepresented sequence, indicates the possible bias in PCR replicate. If so, this could influence the following differential analyses. But it could also due to the real differential expression.



I implement trimmamotic on SRR6040095\_scaffold\_06.1 and 06.2, using the paired end mode (PE), remove bases with quality below 3 started from 5’ end (LEADING:3), remove bases with quality below 3 started from 3’ end (TRAILING:3), set the size of sliding window to 4, and cutoff when average quality below 20 (SLIDINGWINDOW:4:20), if reads after cutoff is shorter than 50bp, discard this read (MINLEN:50).

The result shows like: Input Read Pairs: 2768706 Both Surviving: 2634801 (95.16%) Forward Only Surviving: 36149 (1.31%) Reverse Only Surviving: 83525 (3.02%) Dropped: 14231 (0.51%)

The quality of my data after trimmed improved in N base rate and sequence quality, but it can’t deal with issues caused by replication.



According to CANU’s documentation, it will report the histogram of read lengths, the histogram or k-mer in the raw and corrected reads, the summary of corrected data, summary of overlaps, and the summary of contig lengths.

There are 3 phases: correction, trimming and assembly. In correction and trimming phase, reads quality improved, can see report or histogram of reads and mers of each step. The assembly phase will order the reads into contigs, generate consensus sequences and create graphs of alternate paths, also provide reports and histograms.

In my result, I get 659 contigs. According to the formula (with N:232420, c: 66.86, T: 1, average L: 7767) and NCBI information, I should have 1 contig.

A contig seems to be reads linked via overlaps. A unitig can be part of a contig, or be ‘an over compressed assembly of several high-fidelity copies of a repeat’.

Scaffold contains gaps and contigs. The order of contigs in scaffold is indicated via paired-end or mate-end library.

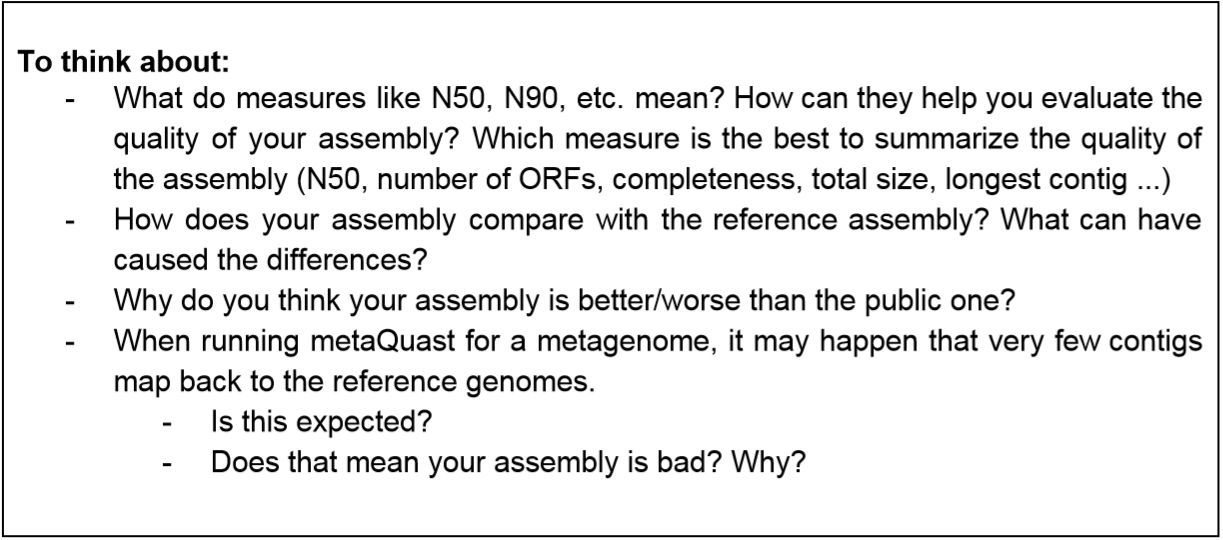
K-mer is defined by the user, the length of the substrings of the reads. For example, AATTCC can be divided into AATT ATTC TTCC when setting k-mer=4. I’m analyzing plant, so usually I should use k-mer=31. Choosing smaller k-mer increase the chance for all the k-mer to overlap, but could result a higher level of path ambiguities. Larger k-mer can detect small repeat regions, but the risk of not overlapping with another k-mer by k-1 increased. By the way, smaller k-mer results in more calculation effort, larger k-mer results in more memory to store DNA sequence.

In CANU’s case, read-correction step includes correction (improve the accuracy of bases) and trimming (such as removing suspicious adapter).

It aims to increase the read quality, which could reduce the calculation burden of the following assembly step.

When comparing my genome assembly (using CANU) with the assembly in paper (using FALCON), there’s many ‘noises’ which indicates some small difference between the two assembly. This could partly due to a smaller subset of data, also could be caused by different tools of assembly. But at large, the majority of assembly keeps the same. Also, via different working process and different parameter settings, different tools could be used to assemble: short-reads/long-reads, genomic/transcriptome, single-cell/metagenomics, etc.

I didn’t see any in my CANU output fasta, but there could be: U (for RNA), R (for G A), Y (for T C), K (for G T) , M (for A C), S (for G C), W (for A T) , B (for G T C), D (for G A T), H (for A C T), V (for G C A), N (for A G C T). This allows a certain ambiguous degree.

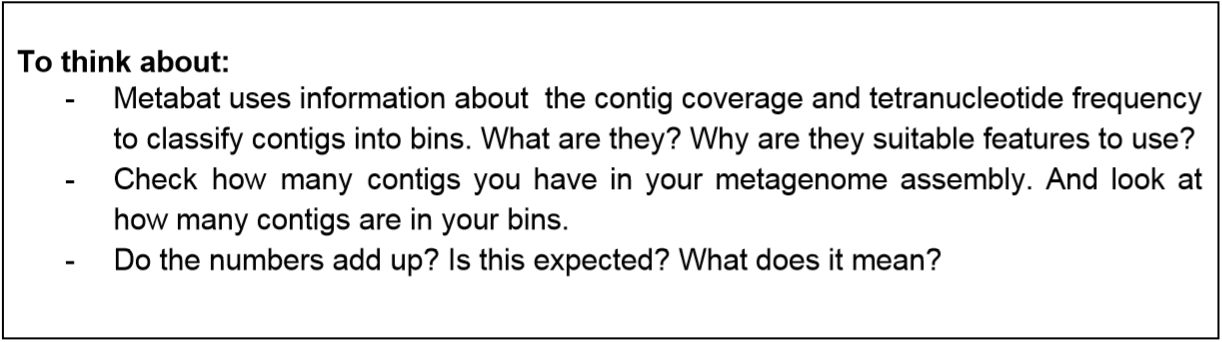


N50 means the sequence length of the shortest contig at 50% of the total genome length. N90 is 90%. Longer N50 may indicate a better assembly quality, but it’s not absolute. When summarizing quality, a fine way might be to compare with the ‘correct assembly’, but it usually doesn’t exist. So, it is better to combine the result of every measurement together.

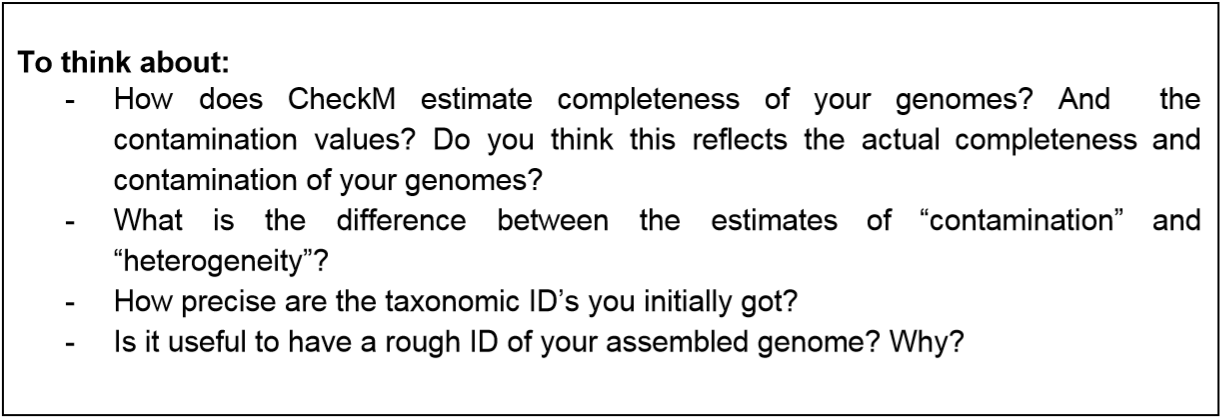
My assembly is similar to the reference assembly, but have some ‘noise’ around. It could due to the different assembly tools, also could due to different coverage.

I think my assembly is worse than the public one, because in paper the researchers tried CANU (I’m using that) but found a poor quality, than they turn to another tool. Also, in my trial, the polish tool Pilon doesn’t improve the assembly. And my coverage is far more less than that in paper.

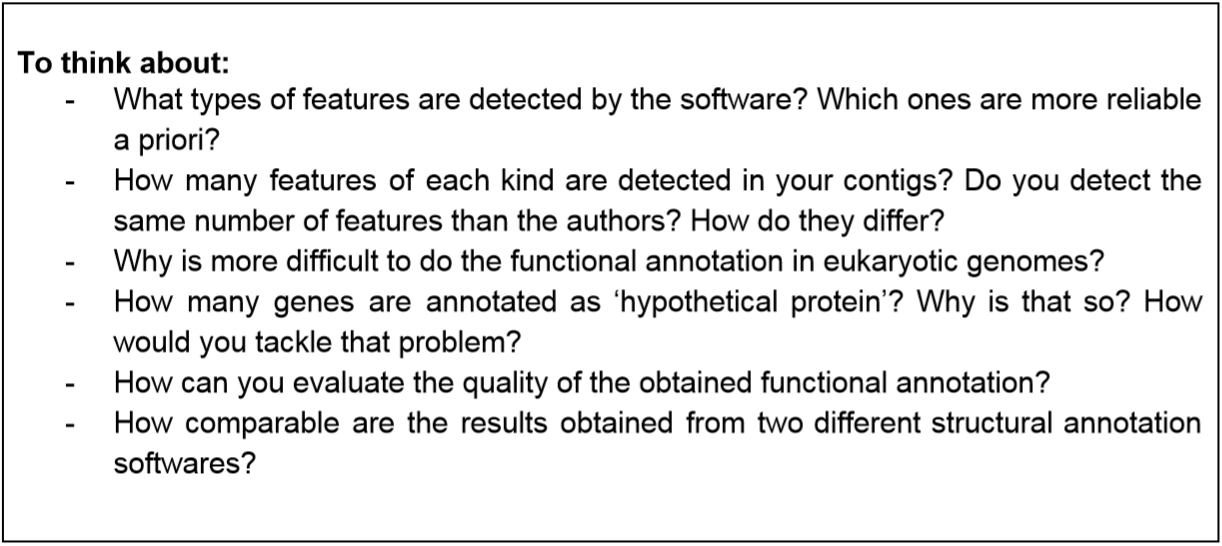
Didn’t do metaQuast but here’s the guess: There might be local misassembles or interspecies shifts. And some poor-quality reads are filtered out before assembly.



Didn’t have this task in Paper 5.



Didn’t have this task in Paper 5.



Maker detect structural features (CDS, repeat, mRNA, exon, 5’ UTR, etc.), in my case CDS are more reliable priori, since there might be intron. EggNOGmapper predicts functional annotation of predicted sequences.

For CDS I detected 1439. I didn’t detect the same number, due to: My assembly quality is worse than paper. I didn’t use protein as input in the first round of MAKER. I abandon MAKER result with trained snaphmm and AUGUSTUS and use the untrained one, which is not correct. Though I didn’t see the gene number of contig 6 in paper, I think I get far more CDS than the paper, because some of them should be masked in the following MAKER rounds.

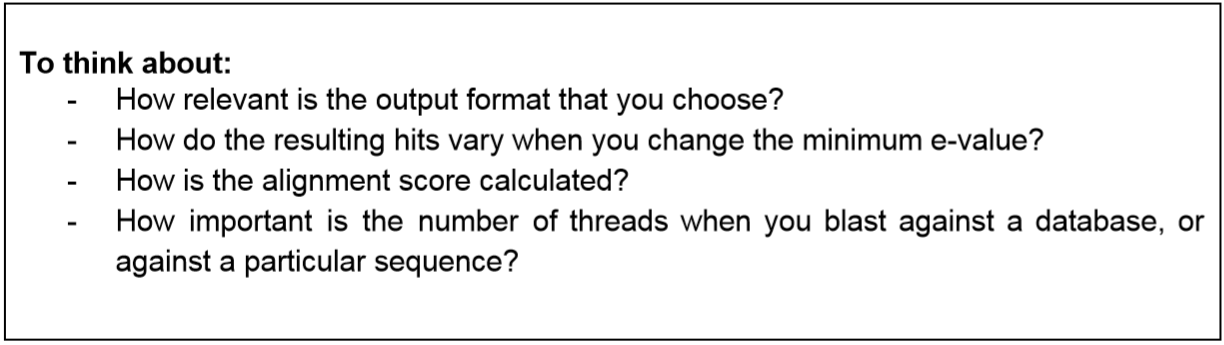
Functional annotation is more difficult in eukaryotic, partly due to alternative splicing (and sometimes undetected introns), which could not be detected via blast.

I grep 1444 CDS from my gff file, 1061 sequence but only 1012 can be annotated

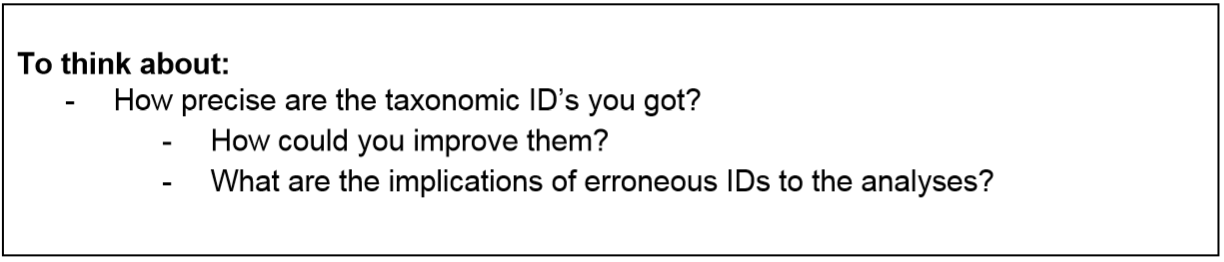
In my ‘export.aa’ file there are 1061 sequence, converted from ‘uni.ann’ and ‘uni.dna’ , since there’s almost none overlap gene in eukaryotes, on each sequence there’s single gene. But only 1012 can be annotated to a function, the rest might be hypothetical protein.

There are many methods to evaluate structural annotation, such as ‘sn’, ‘sp’ of exon annotation. But I’m not sure about the functional annotation. I think we could check the result of blast (such as e-value) to deduct the quality of each gene, and count the percentage of each quality to see if this annotation is good. (Not manually, can write a script.) Also, can compare the result of different annotation software.

I think different structural annotation software may use different pipeline, for different usages. The pipeline of annotating prokaryote and eukaryote can be different due to exon/intron, different promoter, etc. Also, different software may use different training set or model. If choose the wrong pipeline the result could be significantly different. But if using the suitable ones, the overall result should be the same,



Didn’t have this task in Paper 5.



Didn’t have this task in Paper 5.

