**W3D4 Identification of RNAseq reads**

**<1> Map the (single end) reads in X.fastq to the Arabidopsis genome useing Bowtie2, let the output be SAM format and as index chose Bowtie2Index/genome. Use default settings. Record the number of mapped and unmapped reads.**

bowtie2 -x Bowtie2Index/genome -U X.fasta -S X.sam

247 reads; of these:

247 (100.00%) were unpaired; of these:

55 (22.27%) aligned 0 times

185 (74.90%) aligned exactly 1 time

7 (2.83%) aligned >1 times

77.73% overall alignment rate

**<2> Create a sorted BAM file from the SAM file using samtools.**

samtools view -bS X.sam > x.bam

samtools sort x.bam x\_sorted

**<3> Create an index for the sorted BAM file using samtools.**

samtools index x\_sorted.bam

created the index file x\_sorted.bam.bai

**<4> Start the IGV genome browser and load the A.thaliana TAIR10 genome (“Load Genome From Server”)**

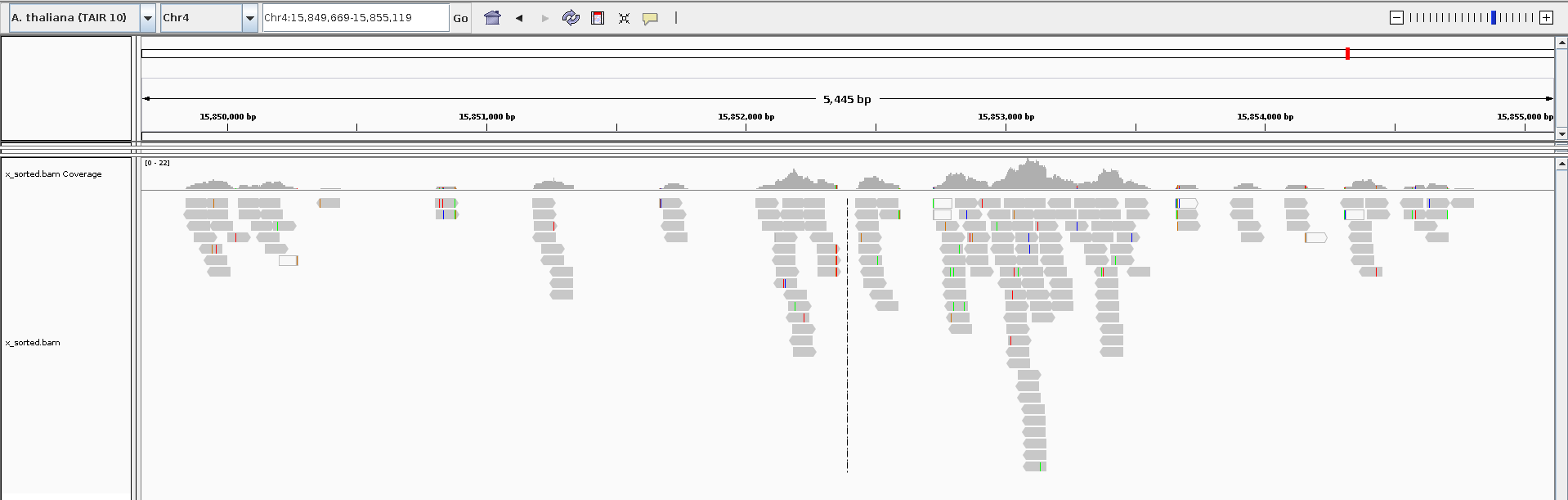
**<5> Load the BAM file in IGV (with the index) and check which isoforms are detected (find the right genome coordinates in the SAM file). To see the isoforms, you might have to right click on the gene track and change Collapsed to Expanded.**

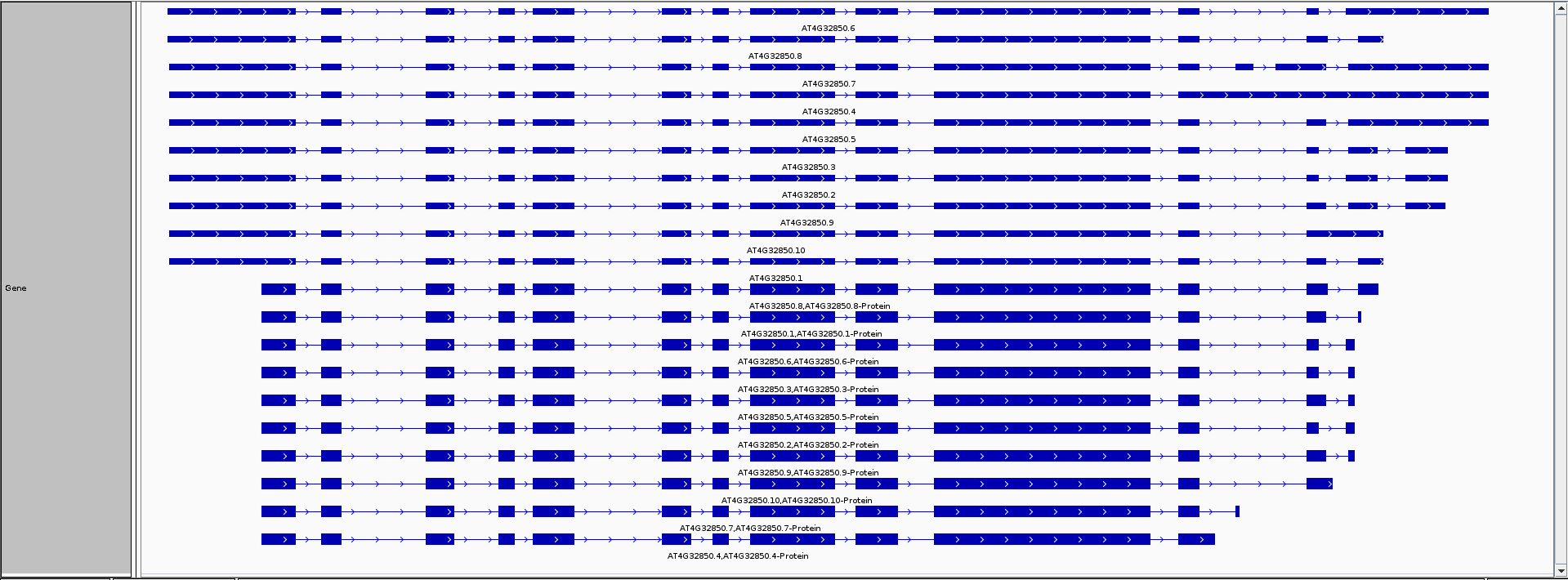
nano X.sam

find the positions for isoform is on chr4:around 15850000

igv& (open igv)

go to the positions in the IGU





**<6> repeat steps 1-5 using HISAT2 as the ligner, as index use TAIR (this will use the TAIR.\*.ht2 files). In IGV check the ‘cigar’ values for some of the new informative reads.**

*Map the X.fastq to the Arabidopsis genome*

hisat2 -x TAIR -U X.fastq -S x\_hisat.sam

247 reads; of these:

247 (100.00%) were unpaired; of these:

0 (0.00%) aligned 0 times

240 (97.17%) aligned exactly 1 time

7 (2.83%) aligned >1 times

100.00% overall alignment rate

*Create a sorted BAM file from the SAM file*

samtools view -bS x\_hisat.sam > x\_hisat.bam

samtools sort x\_hisat.bam x\_hisat\_sorted

*Create an index for the sorted BAM file*

samtools index x\_hisat\_sorted.bam

created the index file x\_hisat\_sorted.bam.bai

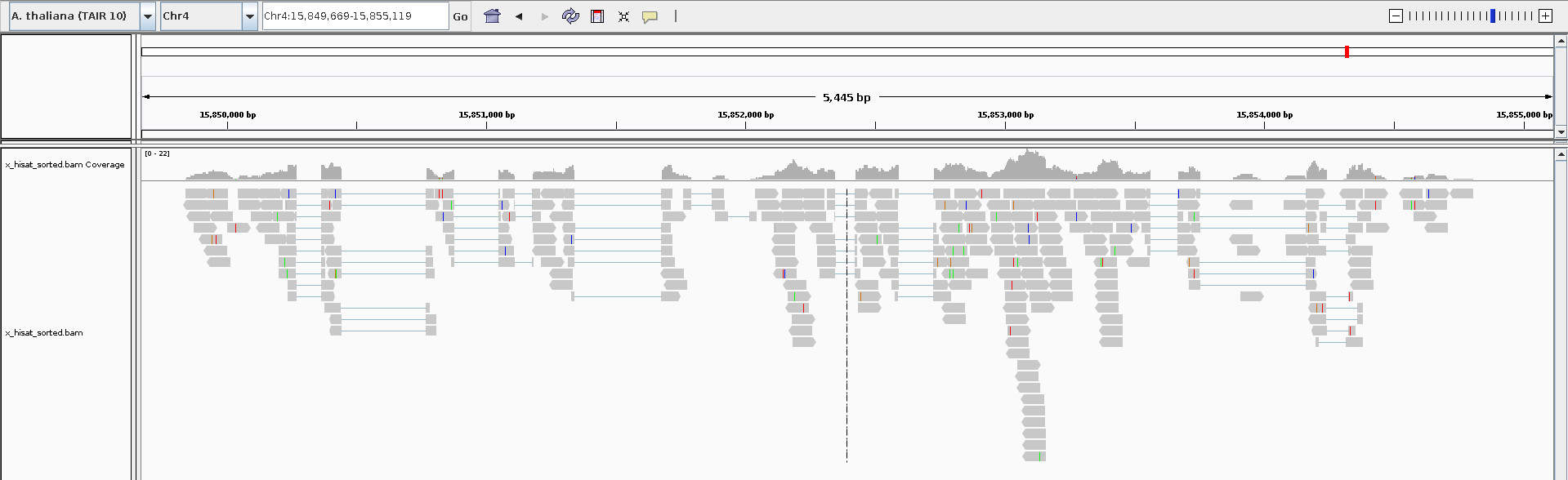
*Start the IGV genome browser and check for the isoforms*

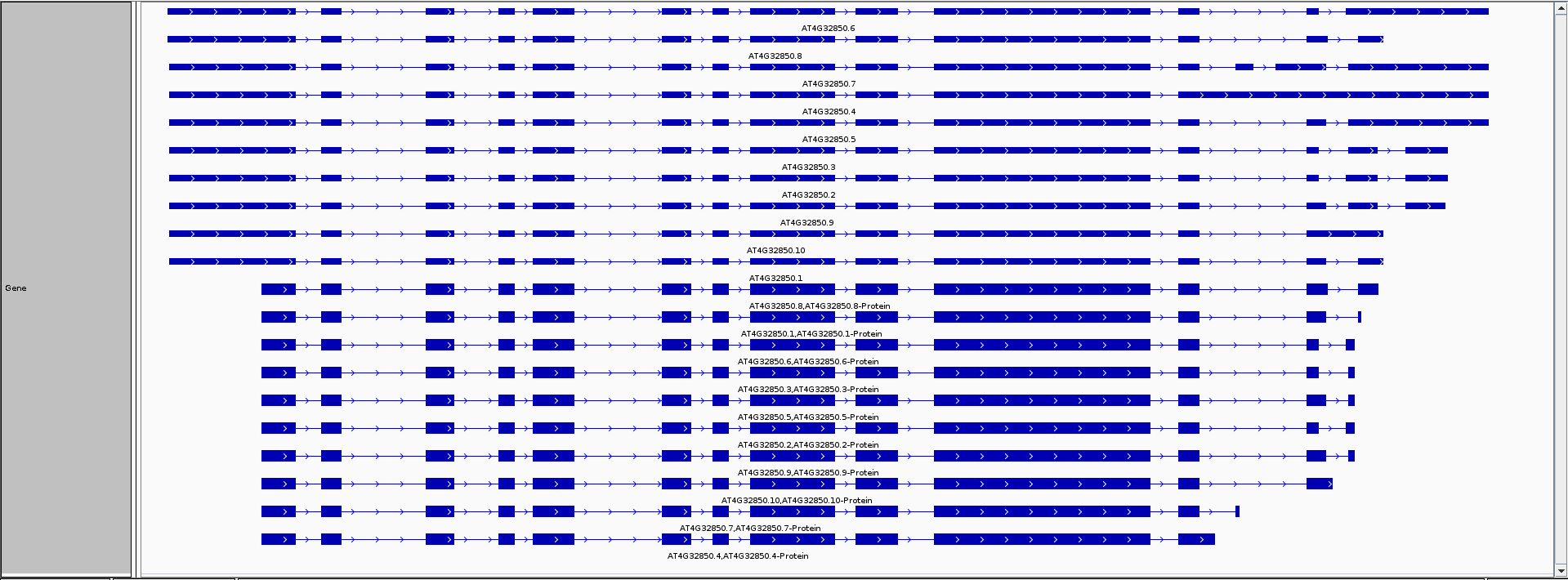
nano x\_hisat.sam

find the positions for isoform is on chr4:around 15852000

igv& (open igv)

go to the positions in the IGU





1. **Which tool performs the best?**

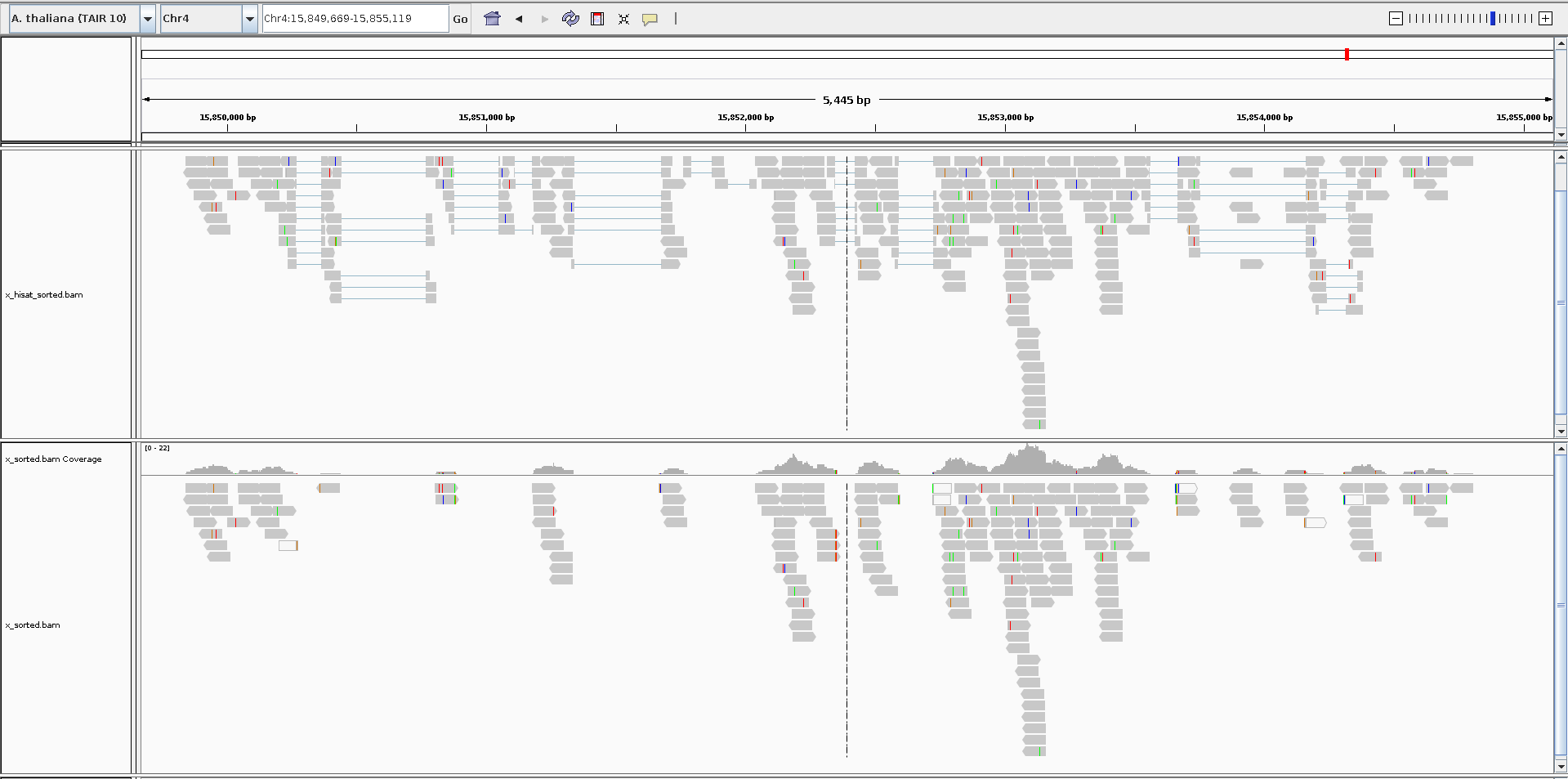
‘Cigar’ values tells how good the mapping is:

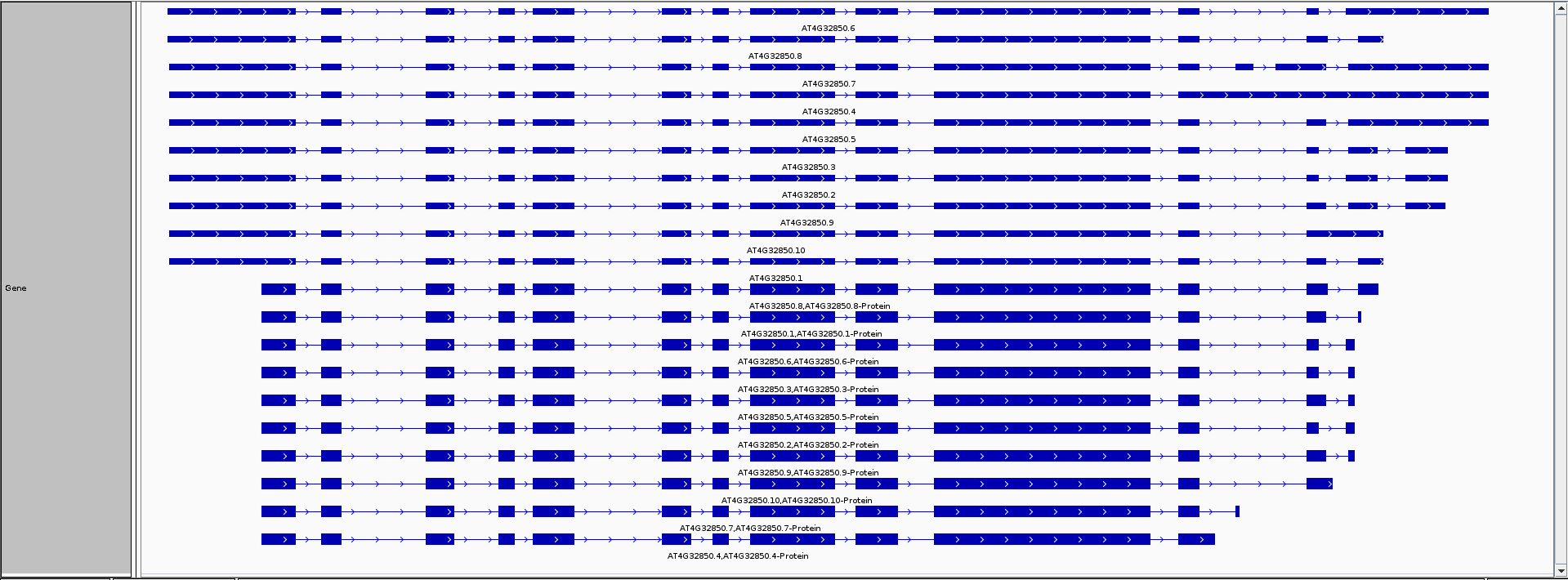
* for HISAT2, it is 78M
* in Bowtie, there are other values, for example:50M81N28M

HISAT2 performs better than Bowtie2:

* it cut the edges more clearly and accurately (can find the intron)
* It also shows the overlap

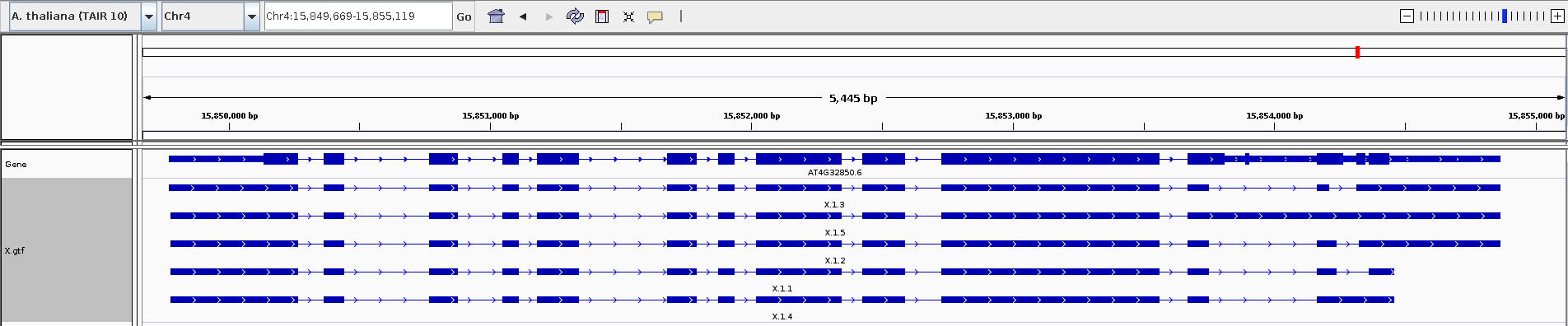
1. **Were these reads strand specific?**





**<7> Use StringTie to predict the transcripts from the HISAT2 BAM output, use the genes.gtf annotation file. Check the resulting GTF file in IGV, do the predicted transcript correspond with the mapped reads?**

stringtie -G genes.gtf -o X.gtf -l X x\_hisat\_sorted.bam



5 transcripts has been predicted

**<8> Use Trinity to do a *De Novo* assembly of the reads, using default settings. The results of the assembly will be in: trinity\_out\_dir/Trinity.fasta**

Trinity -seqType fq --max\_memory 10G -single X.fastq

**<9> Make a Blast database called TAIR10 from the Arabidopsis TAIR 10 genome.**

makeblastdb -in genome.fa -dbtype nucl -out TAIR10

**<10> Use Blast to search for the Trinity.fasta sequences in the TAIR10 database. Use an E-value cut-off of 1E-10, choose output format 7 (tabular).**

blastn -query trinity\_out\_dir/Trinity.fasta -db TAIR10 -evalue 1e-10 -outfmt 7 -out trinity.blast

**<11> Write a python script to convert the output to a valid GFF file, like below:**

chr1 blast exon 1234567 1234569 . + . gene=Unknown

**<12> Load the GFF file into IGV and look at the resulting transcriptome (zoom in to the right coordinates). Did Trinity do a good job? Why?**

2 transcripts have been predicted. Less than the result from StringTie (5). Because StringTie used the genome information.

**<13> Use kallisto to “pseudo align” the reads to the Arabidopsis transcriptome. The TAIR.idx index for that is already available. Set “length mean” to 200 and sd to 20, and specify an output directory.**

kallisto quant -i TAIR.idx -o X --single -l 200 -s 20 -t 20 X.fastq

**<14> the abundance.tsv file contains the results of the kallisto run, which transcripts did kallisto identify?**

cat X/abundance.tsv | awk '$4 > 0'

target\_id length eff\_length est\_counts tpm

AT4G32850.2 2217 2018 26.5809 130751

AT4G32850.9 2238 2039 14.1023 68654.5

AT4G32850.1 2226 2027 13.7518 67344.5

AT4G32850.5 2238 2039 14.1023 68654.5

AT4G32850.8 2298 2099 112.86 533737

AT4G32850.4 2193 1994 0.00304219 15.1446

AT4G32850.10 2238 2039 0.00471269 22.9429

AT4G32850.7 2151 1952 0.00921966 46.8849

AT4G32850.3 2208 2009 0.00444594 21.9675

AT4G32850.6 2217 2018 26.5809 130751

(5 of them should matches the results from SpringTie)

**<15> What is your conclusion after using all these tools to identify the transcripts that produced the reads in X.fastq?**