**File types in NGS**

FASTA files (.fasta or .fa)

>MCHU - Calmodulin - Human, rabbit, bovine, rat, and chicken  
ADQLTEEQIAEFKEAFSLFDKDGDGTITTKELGTVMRSLGQNPTEAELQDMINEVDADGNGTID  
FPEFLTMMARKMKDTDSEEEIREAFRVFDKDGNGYISAAELRHVMTNLGEKLTDEEVDEMIREA  
DIDGDGQVNYEEFVQMMTAK\*

“>” denotes a quote in NGS – usually for scaffold assembly/cDNA/upstream promoter sequences found in databases

FASTQ files (.fastq or .fq)

@A00627:41:HMGW3DSXX:2:1101:6180:1110 1:N:0:ATCACGAT+TCTTTCCC  
CNTCACACTAAATACGTACCGATTGCGGTACTGGATAAATTTCCAAATGAACAATCCAGCCATTATCCCAATCCACAGTGCAAGTACCCAGAGGCGTTTCCAATTGTCCTCCAGATAATAGCTTAATTTGCTGCTCATTTTTCGGATCGG

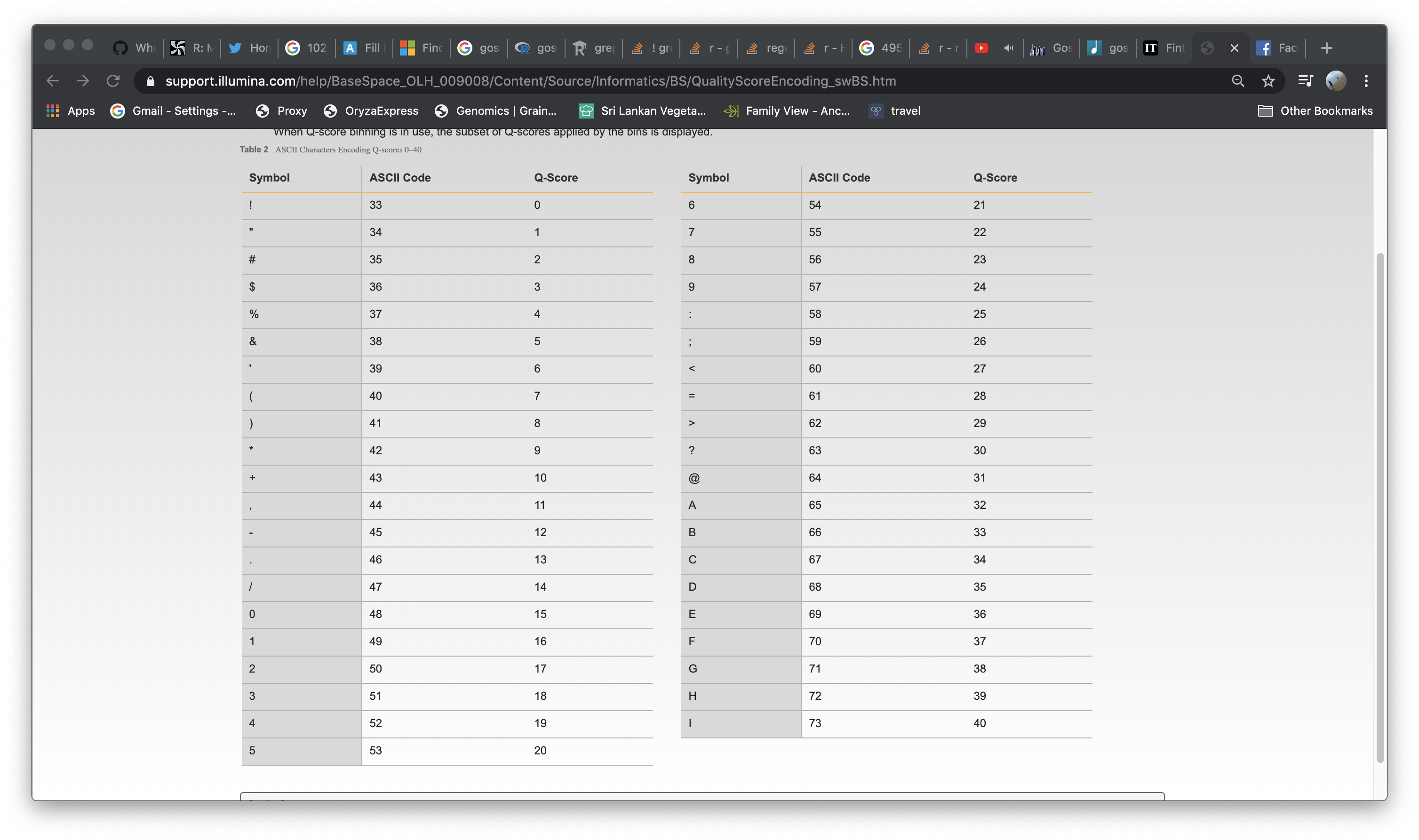
+

F#FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF,FFFFFFFFFF:FFFFFFFFFFFF:FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF:FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Line 1: “@” denotes the location flowcell and other details of run/machine etc. Can see the red coloured nucleotides are the adapter sequences.

Line 2: Base call with N signifying an inconclusive call

Line 3/4: FASTQ call – see table below



SAM/BAM file format

Sam file (.sam or .sorted.sam)

HWI-ST188:3:1103:7821:157455#0 355 Chr01 3683 3 101M = 7126 3498 GGAGAACTCAAGTTCAAATTTGTGGACAAATGGTAGAGATGCATCTACCTCCACACACAACCGCGCATAGTCTAGGCGCTTACATCCGAGTGTCAACTCAT bbbeeeeegfgggihhhhiihhfgigihiidfhfdgfhfghh\_gfhihghhhiiifhhhhhffhedcecd\_bcddbRKWZa\

Sam files are human-readable files that have 11 mandatory columns that tell information of how the fastq read mapped to the indexed reference.

Col 1: Query template (depends on aligner)

Col 2: bitwise flag

Col 3: Scaffold (reference name)

Col 4: Position

Col 5: Mapping quality

Col 6: CIGAR string

Col 7: Ref name of next read

Col 8: Position of the next read

Col 9: Observed template length

Col 10: segment sequence

Col 11: ASCII of Phred-scaled base Quality

Bam files (.bam or .sorted.bam)

Bam files are written in binary, they’re faster to process, take up less memory but are not human readable.

GTF/GFF files

Different flavours of the same file type. These files contain all the location and feature information for a genome assembly. Programs that count features use these files as their guide.

The BAM/SAM file stores an alignment to a particular location of the reference assembly and your feature counting program will say something akin to “Oh this read maps into the same location as gene X in the reference, this must be a read from gene X. I’ll change gene X from *n* to *n+1*”.

I’ve copied and pasted some info from the ensembl website here to explain what all the columns mean. A new format they’ve started using is called gff3 which allows an extra 10th column for whatever attribute you want. This isn’t a hugely popular development as some old programs that aren’t being updated can’t use these inputs, they just get confused about the extra column.



Fields

Fields must be tab-separated. Also, all but the final field in each feature line must contain a value; "empty" columns should be denoted with a '.'

1. seqname - name of the chromosome or scaffold; chromosome names can be given with or without the 'chr' prefix. Important note: the seqname must be one used within Ensembl, i.e. a standard chromosome name or an Ensembl identifier such as a scaffold ID, without any additional content such as species or assembly. See the example GFF output below.
2. source - name of the program that generated this feature, or the data source (database or project name)
3. feature - feature type name, e.g. Gene, Variation, Similarity
4. start - Start position of the feature, with sequence numbering starting at 1.
5. end - End position of the feature, with sequence numbering starting at 1.
6. score - A floating point value.
7. strand - defined as + (forward) or - (reverse).
8. frame - One of '0', '1' or '2'. '0' indicates that the first base of the feature is the first base of a codon, '1' that the second base is the first base of a codon, and so on..
9. attribute - A semicolon-separated list of tag-value pairs, providing additional information about each feature.