Modifications to the project:

- .fa and .gff files are in Dr H's email. I have put them in HPC
- Don't trim anymore. It causes data loss and the Brar paper has no mention of it.
- Go directly to FastQC (quality checker)
- Use HISAT or STAR to align for this RNA-seq instead of BWA.
- After that I presume is where I download all the data and work on it via RStudio. I can probably download them to the exSD.nal SSD.

Note to self: Workflow for FASTQ alignments (from HW2)

- load modules, get all the .fastq files (from SRR on NCBI GEO. get list if needed)
- FastQC
- STAR to align (uses .fa file directly and has higher rate than HISAT2)
- Download data and continue on RStudio

Materials:

- SRR Acc List (Only for those in mRNA-seq traditional time course). There are 12. (Will obtain fastqs from this)
- FASTA file
- GFF file

In [75]:
```
cat ./SRR_Acc_List_REAL.txt
# 12 fastq files from this.
```

```
SRR387839
SRR387840
SRR387846
SRR387845
SRR387844
SRR387843
SRR387842
SRR387841
SRR387848
SRR387849
SRR387850
SRR387865
```

In [76]:
```
wc -l ./SRR_Acc_List_REAL.txt
```

```
12 ./SRR_Acc_List_REAL.txt
```

In [54]:
```
# .fa and .gff files from Dr. Andreas Hochwagen
ls | grep *.fa
ls | grep *.gff
```

```
SK1.genome.fa
SK1.all_feature.gff
```

Obtaining fastq files via fasterq-dump

In [6]:
```
pwd
```

```
/scratch/df2463/AppGen2024/Brar Final Project
```

In [7]:
```
module avail sra-tools
```

```
--------------------------- /share/apps/modulefiles ----------------------------
   sra-tools/2.10.9    sra-tools/3.1.0
```

In [24]:
```
module load sra-tools/3.1.0
```

In [25]:
```
module list
```

```
Currently Loaded Modules:
  1) anaconda3/2024.02   2) sra-tools/3.1.0
```

In [5]:
```
#fasterq-dump
    # argument: the SRR # of desired file
    # paired end reads: include --split3 argument
```

Required: Create bourne shell loop that finds all the SRR numbers in Brar et al, and run them in fastq-dump (done) SRR number list text file was grabbed from NCBI GEO page. (done)

In [7]:
```
touch SRR-grabber-dumper.sbatch
```

In [66]:
```
cat SRR-grabber-dumper.sbatch
```

```
#!/bin/bash
#SBATCH --nodes=4
#SBATCH --ntasks-per-node=4
#SBATCH --cpus-per-task=4
#SBATCH --time=2:00:00
#SBATCH --mem=16GB

#SBATCH --job-name=srr-grab-dump

module purge
module load sra-tools/3.1.0


#cat SRR_Acc_List_REAL.txt | while read code
#do
#fasterq-dump $code
#done

cat SRR_Acc_List_REAL.txt | while read -r code
do
    echo "Processing $code"
    if fasterq-dump "$code"; then
        echo "Successfully processed $code"
    else
        echo "Failed to process $code"
    fi
done < SRR_Acc_List_REAL.txt

# WARNING: This code requires the input .txt file to have terminating linefeed, i.e.
an extra linebreak or two at the end. Otherwise lines might be omitted.
# Description: pipe output of reading the list into while read.
    # read command will read every word and save it into a variable code
    # run fasterq-dump on every variable code
```

In [1]:
```
#cat SRR_Acc_List.txt | while read code
#do
#fasterq-dump $code
#done

# pipe output of reading the list into while func.
```

In [67]:
```
sbatch SRR-grabber-dumper.sbatch
```

```
Submitted batch job 45700053
```

In [79]:
```
squeue -u df2463
```

```
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          45700378        cs srr-grab   df2463  R       7:21      4 cs[065-068]
          45697119     short ood-jupy   df2463  R    1:44:55      1 cm001
```

In [86]:
```
ls -l | grep '\.fastq' | wc -l
# one fastq for each SRR code.
# 12 fastq files
```

```
12
```

Quality checking via FastQC

In [87]:
```
module avail fastqc
```

```
--------------------------- /share/apps/modulefiles ----------------------------
   fastqc/0.11.9
```

In [88]:
```
module load fastqc/0.11.9
```

In [16]:
```
fastqc -h
```

```
                FastQC - A high throughput sequence QC analysis tool

SYNOPSIS

        fastqc seqfile1 seqfile2 .. seqfileN

    fastqc [-o output dir] [--(no)extract] [-f fastq|bam|sam]
           [-c contaminant file] seqfile1 .. seqfileN

DESCRIPTION

    FastQC reads a set of sequence files and produces from each one a quality
    control report consisting of a number of different modules, each one of
    which will help to identify a different potential type of problem in your
    data.

    If no files to process are specified on the command line then the program
    will start as an interactive graphical application.  If files are provided
    on the command line then the program will run with no user interaction
    required.  In this mode it is suitable for inclusion into a standardised
    analysis pipeline.

    The options for the program as as follows:

    -h --help       Print this help file and exit

    -v --version    Print the version of the program and exit

    -o --outdir     Create all output files in the specified output directory.
                    Please note that this directory must exist as the program
                    will not create it.  If this option is not set then the
                    output file for each sequence file is created in the same
                    directory as the sequence file which was processed.

    --casava        Files come from raw casava output. Files in the same sample
                    group (differing only by the group number) will be analysed
                    as a set rather than individually. Sequences with the filter
                    flag set in the header will be excluded from the analysis.
                    Files must have the same names given to them by casava
                    (including being gzipped and ending with .gz) otherwise they
                    won't be grouped together correctly.

    --nano          Files come from nanopore sequences and are in fast5 format. In
                    this mode you can pass in directories to process and the program
                    will take in all fast5 files within those directories and produc
e
                    a single output file from the sequences found in all files.

    --nofilter      If running with --casava then don't remove read flagged by
                    casava as poor quality when performing the QC analysis.

    --extract       If set then the zipped output file will be uncompressed in
                    the same directory after it has been created.  By default
                    this option will be set if fastqc is run in non-interactive
                    mode.
```

```
-j --java       Provides the full path to the java binary you want to use to
                launch fastqc. If not supplied then java is assumed to be in
                your path.

--noextract     Do not uncompress the output file after creating it.  You
                should set this option if you do not wish to uncompress
                the output when running in non-interactive mode.

--nogroup       Disable grouping of bases for reads >50bp. All reports will
                show data for every base in the read.  WARNING: Using this
                option will cause fastqc to crash and burn if you use it on
                really long reads, and your plots may end up a ridiculous size.
                You have been warned!

--min_length    Sets an artificial lower limit on the length of the sequence
                to be shown in the report.  As long as you set this to a value
                greater or equal to your longest read length then this will be
                the sequence length used to create your read groups.  This can
                be useful for making directly comaparable statistics from
                datasets with somewhat variable read lengths.

-f --format     Bypasses the normal sequence file format detection and
                forces the program to use the specified format.  Valid
                formats are bam,sam,bam_mapped,sam_mapped and fastq

-t --threads    Specifies the number of files which can be processed
                simultaneously.  Each thread will be allocated 250MB of
                memory so you shouldn't run more threads than your
                available memory will cope with, and not more than
                6 threads on a 32 bit machine

-c              Specifies a non-default file which contains the list of
--contaminants  contaminants to screen overrepresented sequences against.
                The file must contain sets of named contaminants in the
                form name[tab]sequence.  Lines prefixed with a hash will
                be ignored.

-a              Specifies a non-default file which contains the list of
--adapters      adapter sequences which will be explicity searched against
                the library. The file must contain sets of named adapters
                in the form name[tab]sequence.  Lines prefixed with a hash
                will be ignored.

-l              Specifies a non-default file which contains a set of criteria
--limits        which will be used to determine the warn/error limits for the
                various modules.  This file can also be used to selectively
                remove some modules from the output all together.  The format
                needs to mirror the default limits.txt file found in the
                Configuration folder.

-k --kmers      Specifies the length of Kmer to look for in the Kmer content
                module. Specified Kmer length must be between 2 and 10. Default
                length is 7 if not specified.

-q --quiet      Supress all progress messages on stdout and only report errors.
```

```
   -d --dir        Selects a directory to be used for temporary files written when
                   generating report images. Defaults to system temp directory if
                   not specified.

BUGS

    Any bugs in fastqc should be reported either to simon.andrews@babraham.ac.uk
    or in www.bioinformatics.babraham.ac.uk/bugzilla/
```

In [89]:
```
mkdir fastqc
```

In [25]:
```
# test run
#fastqc -o ./fastqc/ SRR387838.fastq
```

```
Started analysis of SRR387838.fastq
Approx 5% complete for SRR387838.fastq
Approx 10% complete for SRR387838.fastq
Approx 15% complete for SRR387838.fastq
Approx 20% complete for SRR387838.fastq
Approx 25% complete for SRR387838.fastq
Approx 30% complete for SRR387838.fastq
```

In [11]:
```
#cp /scratch/df2463/AppGen2024/agweek2/fastqc.sbatch ./
```

In [12]:
```
#mv ./fastqc.sbatch fastqcHeavy.sbatch
```

In [91]:
```
cat ./fastqc.sbatch
# This runs fastqc on each of the 12 files
```

```
#!/usr/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=4
#SBATCH --cpus-per-task=4
#SBATCH --time=1:00:00
#SBATCH --mem=16GB

#SBATCH --job-name=fastQC


module purge
module load fastqc/0.11.9


fastqc -t 6 -o ./fastqc/ *.fastq

# output = fastqc directory
# input = all .fastq files
# -t    Specifies the number of files which can be processed
                   simultaneously.  (6 here)
```

In [93]:
```
sbatch ./fastqc.sbatch
# Success. None of the 12 fastqc files report any bad sequences. Proceeding to STAR
```

```
Submitted batch job 45700538
```

In [95]:
```
squeue -u df2463
```

```
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          45697119     short ood-jupy   df2463  R    2:10:50      1 cm001
```

STAR Alignment

In [96]:
```
module avail star
```

```
--------------------------- /share/apps/modulefiles ----------------------------
   star/intel/2.7.6a    star/intel/2.7.11a    starpu/openmpi/intel/20230822
```

In [97]:
```
module load star/intel/2.7.11a
```

In [1]:
```
#cp /scratch/df2463/AppGen2024/Homework2/STAR_script.sbatch ./
```

In [4]:
```
#mv ./STAR_script.sbatch STAR_Sc.sbatch
```

In [98]:
```
mkdir STAR_index
```

In [99]:
```
# .fa and .gff files from Dr. Andreas Hochwagen
ls | grep *.fa
ls | grep *.gff
```

```
SK1.genome.fa
SK1.all_feature.gff
```

In [103…
```
## Genome Generation (using .fa)
cat ./STAR_GG.sbatch
# recommended genomeSAindexNbases was 10
# This one is specifically tailored towards our Sc .fa
```

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=16
#SBATCH --time=1:00:00
#SBATCH --mem=16GB

#SBATCH --job-name=STAR_GG

module purge
module load star/intel/2.7.11a

## NOTE: Pls make a STAR_index directory first.
## This is for the Saccharomyces cerevisiae.

## GENOME GENERATE
STAR --runThreadN 16 --runMode genomeGenerate --genomeDir STAR_index \
--genomeFastaFiles SK1.genome.fa --genomeSAindexNbases 10


    # Index building via STAR. 2 threads, run in index generation mode.
        # Store in STAR_index folder and use provided .fa file
        # SA pre-indexing string of 12 bases \
            # (by my understanding, slice in groups of 12)
        # For this data, genomeSuffixLengthMax can be anything > 75, \
            # but the default is -1 (infinite) anyway
```

In [115…
```
sbatch ./STAR_GG.sbatch
```

```
Submitted batch job 45700808
```

In [104…
```
touch STAR_Sc_12.sbatch
```

In [1]:
```
## Read alignment (BAM from fastq files)
cat ./STAR_Sc_12.sbatch
## Final perfected version of the STAR_Sc sbatches.
# This one properly runs on only the 12 files we need.

## Sorted by coordinate. So in htseq, specify that the --order = pos
```

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=16
#SBATCH --time=6:00:00
#SBATCH --mem=32GB

#SBATCH --job-name=STAR_Scfinal

module purge
module load star/intel/2.7.11a

## This script is made to run one-by-one on 12 fastq files. For anything more than t
his, it is advisable to use arrays or splitting the fastq files into multiple folder
s and running this on each of the folders.
## Final perfected version of the STAR_Sc sbatches. This one properly runs on only t
he 12 files we need.


## READ ALIGN

for file in *.fastq
do
STAR --runMode alignReads --genomeDir ./STAR_index --outSAMtype BAM SortedByCoordina
te --readFilesIn ${file} --runThreadN 16 --outFileNamePrefix STAR_SK
done

    # Reading alignments via STAR.
        # w/o specification, runMode argument defaults to alignReads.
        # (runThreadN appears to be for genome generation only.)
```

In [2]:
```
sbatch ./STAR_Sc_12.sbatch
```

```
Submitted batch job 45709733
```

In [16]:
```
ls -l | grep ".bam"
```

```
-rw-r--r--. 1 df2463 df2463  693102431 Apr 27 03:08 STAR_SKAligned.sortedByCoord.ou
t.bam
```

In [15]:
```
head ./STAR_SKAligned.sortedByCoord.out.bam
```

```
□□□□BC□□□}□□n□@□□□□□□□□ً□ □pB□□□□□□□□w}□?A-□7ě□□□□mk)□□I□T□□W.□□□□□□
x□□□□□□□□□{□□□J-Y□□□□□ov□□□o=l□□□□□□X□E$□Ě$I□Fq□□r□}&□□5□Hmi□X□J□
□□T`□2b□□□\#
□Vm□□)□X□□uO□4LkXp□P□□□E'□□jlu~□ aF□_PS□fV□□0□□□□\Û□:□D□□Ba□□k
uA□e2□h□□□1□h□□□□□□□URb□□□□□□□[□ig,□□□7□Ej□la□H□□□□□t□□□□□□`□□
□c□□s$□□□□^□□q□6□9□q2□□(E□G□□□□(□□j□□□=□□|n□□□□□5jRCo□□□□□□/Y.□'□
□Q□Ш□□□□□□Bu;□□□□□□□□a□□$]□{□+□*□r□w"-□<E□d6□□□M□□[□□□a□{□□Z^□□□
□□□□F□□□□□□□B□□□□lo□□□|□;Ed□Wm□□=□□□+□<□□k□□<□□□□□□lya□□ ;ehC
□L□}□□□.||□□f□□|□□□□□□l□□□a□o'□□□□G□□□□□^Y□□□□f)fG□□□□□o
□□□□□□□BC□□Y□□kll□~□□g?j/□□□□xf□5□f□□M□□□□*□□□1□□□□□K□E □□
□□4□□□A%3q□i□zZÇ*□pP
□JH□C%$n□M!D}□□x□□□□p□
□□NQ□□□>□□□□□□□□□□□~vm̨□□□9□□□□□□□□□□|□□□W□|□d□□_K□□□□|□wT□□//~□□
□□□□□□b□□□_□□<N□□□□~>□O□□□□d□□q2□□□□Ğ□I□I□□Q□□□□wɯ□;□oz'□□×□=|□□on□□ˣ□□□/□□
□□□|·CqT□□□□□□lvyc(V0ď□□□a□□□□d□;□□□□□□{□.Ţ□q□□□□ □□8E□□□□'□□□□ ð□□5□E□□□□□□□□
□□□□(□□@□1□h□□□□$□□F□$□k+
□a□□□□8□□ڑ□□7□V□□□Q=□□□□Z□=□□□□□□□□8□□g□□s□{□□x6□gg□□˙□M□CƘy□;I`O□□p□□_<□
□□qO□1N□□□□D□□□□□□b□□9;□□□9qI□□□□□□□aq□□=□□□□$□□□□9□Fq□□□□I□
□'□!□□□□□□pà%Ä□Y=□□□□□□T□□}□□W□G3□<□□□□□□□□□{□□□□!□H□m□2□□<□
□□J□□>Y□6ص□z□m□Vo□□□□□□□q□8□;lI□□T□8□8□W□□ /□□□□□□□□□□□□□□□□
□□#%f□□q$Ó□□□□□(3qý□□□□婘□'Lk□□□□w□$□A4□□□V□□□{□z□□□□□□0l#-□□W
□/□□□□□Q
□.□□p_□□□8□.□?ʸ□□□□<□M/Sà□H&f□F□□>$□□□'□□□Ƀ>□□□
&□□□1□^_□ђF□e□B□e}ib□|□□tF□>□%I□1F~l?;□""□□y7□□b□a□^θ□□4R□□□□□^_□
□□□□1□□□8d\8xxA,□□□□ĭ□^□□□?X>□□q?□□/□□e□")w□□□□uƌr5□□□α□Ĉ□ى□
(□□□Ù|6□;□□"&□Y□□ī□□□□□□□□jv□□v□
□q?□m4:□□è□□□□0□□□□n□□□□oH□>□□□□□□□®Ņ□□gN□□◦□□y□ □qOR□;□□~□□□G□□□□U-i
□□□=□[□0□□L□:□□□□□□Ù□□□x|□,□
n□□□□2Fe□□□□□2F□□□□□$□y□□□□□s□□(6s□□□□□□ₔ□□□
```


- What do I do with the BAM files afterwards?
- Get the counts. BAM files have align and what they align (see below)to
- Get COUNT table from the BAM files before RStudio offline (htseq-count. This is a module on the HPC cluster. htseq-count from HTSeq still on the cluster. Keep working on this in Bash/Jupy (Done)tr)
- BAM file and GTF file. A table of counts and GTF annotatio(done)ns.

GFF and GTF files are interchangeable if required by a program.

Can I use BAM files instead for htseq? YES --format

ram.

cluster. s.

STAR: ARRAY job or splitting it into smaller fragments (multiple folders)

samtools: Creating index file for .bam

- so it can be used in htseq

In [13]:
```
module avail samtools
```

```
--------------------------- /share/apps/modulefiles ----------------------------
   samtools/intel/1.11    samtools/intel/1.12    samtools/intel/1.14
```

In [14]:
```
module load samtools/intel/1.14
```

In [17]:
```
samtools index STAR_SKAligned.sortedByCoord.out.bam
```

In [19]:
```
ls | grep ".bam"
```

```
STAR_SKAligned.sortedByCoord.out.bam
STAR_SKAligned.sortedByCoord.out.bam.bai
```

htseq-count: Count tables + annotations line-up

In [6]:
```
module avail htseq
```

```
--------------------------- /share/apps/modulefiles ----------------------------
   htseq/0.13.5 (L)

  Where:
   L:  Module is loaded
```

In [2]:
```
module load htseq/0.13.5
```

In [17]:
```
htseq-count -h
# gff file is what i have. excellent...
        #Syntax: htseq-count -f BAM <alignment_files> <gff_file>
# possibly -s yes
# "For stranded=yes and single-end reads, the read has to be mapped to the same str

#https://htseq.readthedocs.io/en/release_0.11.1/count.html
```

```
usage: htseq-count [options] alignment_file gff_file

This script takes one or more alignment files in SAM/BAM format and a feature
file in GFF format and calculates for each feature the number of reads mapping
to it. See http://htseq.readthedocs.io/en/master/count.html for details.

optional arguments:
  -h, --help  show this help message and exit
  --version   Show software version and exit

Written by Simon Anders (sanders@fs.tum.de), European Molecular Biology
Laboratory (EMBL) and Fabio Zanini (fabio.zanini@unsw.edu.au), UNSW Sydney.
(c) 2010-2020. Released under the terms of the GNU General Public License v3.
Part of the 'HTSeq' framework, version 0.13.5.
```

In [3]:
```
htseq-count -f bam -r pos -t gene -i ID STAR_SKAligned.sortedByCoord.out.bam SK1.al
# [Fixed] .bam.bai index file required. Get this through samtools index <bam>
# There is apparently no gene_id attribute on the .gff. Check the file via head
```

```
23388 GFF lines processed.
100000 alignment records processed.
200000 alignment records processed.
300000 alignment records processed.
400000 alignment records processed.
500000 alignment records processed.
600000 alignment records processed.
700000 alignment records processed.
800000 alignment records processed.
900000 alignment records processed.
1000000 alignment records processed.
1100000 alignment records processed.
1200000 alignment records processed.
1300000 alignment records processed.
1400000 alignment records processed.
1500000 alignment records processed.
1600000 alignment records processed.
1700000 alignment records processed.
1800000 alignment records processed.
1900000 alignment records processed.
2000000 alignment records processed.
2100000 alignment records processed.
2200000 alignment records processed.
2300000 alignment records processed.
2400000 alignment records processed.
2500000 alignment records processed.
2600000 alignment records processed.
2700000 alignment records processed.
2800000 alignment records processed.
2900000 alignment records processed.
3000000 alignment records processed.
3100000 alignment records processed.
3200000 alignment records processed.
3300000 alignment records processed.
3400000 alignment records processed.
3500000 alignment records processed.
3600000 alignment records processed.
3700000 alignment records processed.
3800000 alignment records processed.
3900000 alignment records processed.
4000000 alignment records processed.
4100000 alignment records processed.
4200000 alignment records processed.
4300000 alignment records processed.
4400000 alignment records processed.
4500000 alignment records processed.
4600000 alignment records processed.
4700000 alignment records processed.
4800000 alignment records processed.
4900000 alignment records processed.
5000000 alignment records processed.
5100000 alignment records processed.
5200000 alignment records processed.
5300000 alignment records processed.
5400000 alignment records processed.
5500000 alignment records processed.
```

```
5600000 alignment records processed.
5700000 alignment records processed.
5800000 alignment records processed.
5900000 alignment records processed.
6000000 alignment records processed.
6100000 alignment records processed.
6200000 alignment records processed.
6300000 alignment records processed.
6400000 alignment records processed.
6500000 alignment records processed.
6600000 alignment records processed.
6700000 alignment records processed.
6800000 alignment records processed.
6900000 alignment records processed.
7000000 alignment records processed.
7100000 alignment records processed.
7200000 alignment records processed.
7300000 alignment records processed.
7400000 alignment records processed.
7500000 alignment records processed.
7600000 alignment records processed.
7700000 alignment records processed.
7800000 alignment records processed.
7900000 alignment records processed.
8000000 alignment records processed.
8100000 alignment records processed.
8200000 alignment records processed.
8300000 alignment records processed.
8400000 alignment records processed.
8500000 alignment records processed.
8600000 alignment records processed.
8700000 alignment records processed.
8800000 alignment records processed.
8900000 alignment records processed.
9000000 alignment records processed.
9100000 alignment records processed.
9200000 alignment records processed.
9300000 alignment records processed.
9400000 alignment records processed.
9500000 alignment records processed.
9600000 alignment records processed.
9700000 alignment records processed.
9800000 alignment records processed.
9900000 alignment records processed.
10000000 alignment records processed.
10100000 alignment records processed.
10200000 alignment records processed.
10300000 alignment records processed.
10400000 alignment records processed.
10500000 alignment records processed.
10600000 alignment records processed.
10700000 alignment records processed.
10800000 alignment records processed.
10900000 alignment records processed.
11000000 alignment records processed.
11100000 alignment records processed.
```

```
11200000 alignment records processed.
11300000 alignment records processed.
11400000 alignment records processed.
11500000 alignment records processed.
11600000 alignment records processed.
11700000 alignment records processed.
11800000 alignment records processed.
11900000 alignment records processed.
12000000 alignment records processed.
12100000 alignment records processed.
12200000 alignment records processed.
12300000 alignment records processed.
12400000 alignment records processed.
12500000 alignment records processed.
12600000 alignment records processed.
12700000 alignment records processed.
12800000 alignment records processed.
12900000 alignment records processed.
13000000 alignment records processed.
13100000 alignment records processed.
13200000 alignment records processed.
13300000 alignment records processed.
13400000 alignment records processed.
13500000 alignment records processed.
13600000 alignment records processed.
13700000 alignment records processed.
13800000 alignment records processed.
13900000 alignment records processed.
14000000 alignment records processed.
14100000 alignment records processed.
14200000 alignment records processed.
14300000 alignment records processed.
14400000 alignment records processed.
14500000 alignment records processed.
14600000 alignment records processed.
14700000 alignment records processed.
14800000 alignment records processed.
14900000 alignment records processed.
15000000 alignment records processed.
15100000 alignment records processed.
15200000 alignment records processed.
15300000 alignment records processed.
15400000 alignment records processed.
15500000 alignment records processed.
15600000 alignment records processed.
15700000 alignment records processed.
15800000 alignment records processed.
15900000 alignment records processed.
16000000 alignment records processed.
16100000 alignment records processed.
16200000 alignment records processed.
16300000 alignment records processed.
16400000 alignment records processed.
16500000 alignment records processed.
16600000 alignment records processed.
16700000 alignment records processed.
```

```
16800000 alignment records processed.
16900000 alignment records processed.
17000000 alignment records processed.
17100000 alignment records processed.
17200000 alignment records processed.
17300000 alignment records processed.
17400000 alignment records processed.
17500000 alignment records processed.
17600000 alignment records processed.
17700000 alignment records processed.
17797655 alignments  processed.
```

In [21]:
```
head ./SK1.all_feature.gff
# -i = ID.
```

```
chrI    SK1     Y_prime_element_partial 428     3774    .       -       .       ID=Y
_prime_element_partial:chrI:428-3774:-;Name=Y_prime_element_partial:chrI:428-3774:-
chrI    SK1     X_element       4468    4928    .       -       .       ID=X_elemen
t:chrI:4468-4928:-;Name=X_element:chrI:4468-4928:-
chrI    SK1     gene    5066    6637    .       -       .       ID=SK1_01G00010;Name
=YHR216W
chrI    SK1     mRNA    5066    6637    .       -       .       ID=SK1_01T00010.1;Pa
rent=SK1_01G00010
chrI    SK1     exon    5066    6637    .       -       .       ID=SK1_01T00010.1.ex
on.1;Parent=SK1_01T00010.1
chrI    SK1     CDS     5066    6637    .       -       .       ID=SK1_01T00010.1.CD
S.1;Parent=SK1_01T00010.1
chrI    SK1     gene    11129   12442   .       -       .       ID=SK1_01G00015;Name
=BIO6
chrI    SK1     mRNA    11129   12442   .       -       .       ID=SK1_01T00015.1;Pa
rent=SK1_01G00015
chrI    SK1     exon    11129   12442   .       -       .       ID=SK1_01T00015.1.ex
on.1;Parent=SK1_01T00015.1
chrI    SK1     CDS     11129   12442   .       -       .       ID=SK1_01T00015.1.CD
S.1;Parent=SK1_01T00015.1
```

In [6]:
```
head ./SK1.AlignedReadCounts.txt
```

```
SK1_01T00010.1.exon.1   6
SK1_01T00015.1.exon.1   104
SK1_01T00020.1.exon.1   26
SK1_01T00030.1.exon.1   10
SK1_01T00040.1.exon.1   8
SK1_01T00050.1.exon.1   6
SK1_01T00060.1.exon.1   3
SK1_01T00070.1.exon.1   252
SK1_01T00080.1.exon.1   1029
SK1_01T00090.1.exon.1   696
```

And the rest goes to RStudio...

In [ ]:
```

```