Analysis of Targeted DNA sequencing data

#Example FastQ files:

P41019091104191\_R1\_001\_trimmed.fastq.gz

P41019091104191\_R2\_001\_trimmed.fastq.gz

P41019091104192\_R1\_001\_trimmed.fastq.gz

P41019091104192\_R2\_001\_trimmed.fastq.gz

**#Align**

module load bedtools

module load samtools

module load python/3.7.2

**#create a script named align.sh**

python3 -m BSBolt Align -DB /u/home/m/mmorsell/project-mcdb/genomes/GRCh38/BSBolt -F1 P41019091104191\_R1\_001\_trimmed.fastq.gz -F2 P41019091104191\_R2\_001\_trimmed.fastq.gz -t 4 -O P4191 >log\_P4191

python3 -m BSBolt Align -DB /u/home/m/mmorsell/project-mcdb/genomes/GRCh38/BSBolt -F1 P41019091104192\_R1\_001\_trimmed.fastq.gz -F2 P41019091104192\_R2\_001\_trimmed.fastq.gz -t 4 -O P4192 >log\_P4192

**#example for alignment submission**

qsub -cwd -V -N algn\_P -pe shared 4 -l h\_data=8G,h\_rt=6:00:00,highp -M mmorsell -m a align.sh

**#option 1: sort and index**

samtools sort -@ 4 P4191.bam -o P4191\_sorted.bam

samtools index -@ 4 P4191\_sorted.bam

samtools sort -@ 4 P4192.bam -o P4192\_sorted.bam

samtools index -@ 4 P4192\_sorted.bam

**#option 2: de-duplicate**

samtools fixmate -p -m P4191.bam P4191.fixmates.bam

samtools sort -o P4191.fix.sort.bam P4191.fixmates.bam

samtools markdup -r P4191.fix.sort.bam P4191.dedup.bam

samtools index P4191.dedup.bam

**#starting from sorted and indexed bam files**

**#calculate % on-off target**

#files needed:

* bam
* TargetProbes\_sorted.bed

#head TargetProbes\_sorted.bed

chr1 778737 778857 latestCombined\_good

chr1 785425 785545 latestCombined\_remove

chr1 826792 826912 latestCombined\_good

chr1 898819 898939 latestCombined\_good

chr1 898859 898979 latestCombined\_good

chr1 898899 899019 latestCombined\_good

chr1 898939 899059 latestCombined\_good

chr1 898979 899099 latestCombined\_remove

chr1 899019 899139 latestCombined\_remove

chr1 920209 920329 latestCombined\_good

* Repeats\_GRCh38\_sorted4col.bed

#head Repeats\_GRCh38\_sorted4col.bed

chr1 10000 10468 (TAACCC)n

chr1 10468 11447 TAR1

chr1 11504 11675 L1MC5a

chr1 11677 11780 MER5B

chr1 15264 15355 MIR3

chr1 15797 15849 (TGCTCC)n

chr1 16712 16744 (TGG)n

chr1 18906 19048 L2a

chr1 19971 20405 L3

chr1 20530 20679 Plat\_L3

#total number of reads

samtools flagstat P4192\_sorted.bam

samtools flagstat P4191\_sorted.bam

#reads on repeats

bedtools intersect -bed -u -abam P4192\_sorted.bam -b Repeats\_GRCh38\_sorted4col.bed | wc -l > on\_Repeats\_P4192.txt

bedtools intersect -bed -u -abam P4191\_sorted.bam -b Repeats\_GRCh38\_sorted4col.bed | wc -l > on\_Repeats\_P4191.txt

#reads on Target

bedtools intersect -bed -u -abam P4191\_sorted.bam -b TargetProbes\_sorted.bed | wc -l > on\_target\_P4191.txt

bedtools intersect -bed -u -abam P4192\_sorted.bam -b TargetProbes\_sorted.bed | wc -l > on\_target\_P4192.txt

# calculate % of on target vs. off-target

e.g.

P4192\_sorted.bam

--> total: 16209846

--> Repeats: 4778980 (29%)

--> Target: 10926275 (67%)

P4191\_sorted.bam

--> total: 8682497

--> Repeats: 2487263 (28%)

--> Target: 5957360 (68%)

#for the plots

head allProbes\_sorted.bed

chr1 10000 10468 repeats

chr1 10468 11447 repeats

chr1 11504 11675 repeats

chr1 11677 11780 repeats

chr1 15264 15355 repeats

chr1 15797 15849 repeats

chr1 16712 16744 repeats

bedtools multicov -bams P4191\_sorted.bam P4192\_sorted.bam -bed allProbes\_sorted.bed > multicov\_all.txt

#you can process a single file

# you can filter based on column 5 (first .bam) or the following (if multiple bams have been processed

awk '$5>9 && $6>9' multicov\_all.txt > multicov\_filter10x.txt

#output

head multicov\_all.txt

chr1 10000 10468 repeats 2 3

chr1 10468 11447 repeats 0 0

chr1 11504 11675 repeats 0 0

chr1 11677 11780 repeats 0 0

chr1 15264 15355 repeats 0 0

chr1 15797 15849 repeats 0 0

chr1 16712 16744 repeats 0 0

chr1 18906 19048 repeats 0 0

chr1 19971 20405 repeats 0 0

chr1 20530 20679 repeats 0 0

#add the header:

chromosome start end probeSet bam1 bam2 …

🡪 plot in R

e.g.

library(ggplot2)

#header: chromosome start end probeSet bam1 bam2

b<-read.table("R\_multicov\_all.txt", header = T)

#plot with regions covered 10x

ggplot(data=b, aes(x=bam1, group=probeSet, fill=probeSet)) +

geom\_histogram(alpha=0.6, binwidth = 5)+

theme\_classic()+facet\_wrap(~probeSet,nrow=2)+xlim(0,1000)+labs(title="Plot of coverage distribution by probe sets",x="Counts", y = "Number of regions")

#adjust the ylim and xlim parameters to zoom in

ggplot(data=b, aes(x=bam1, group=probeSet, fill=probeSet)) +

geom\_histogram(alpha=0.6, binwidth = 5)+

theme\_classic()+facet\_wrap(~probeSet,nrow=2)+ylim(0,100)+xlim(0,1000)+labs(title="Plot of coverage distribution by probe sets",x="Counts", y = "Number of regions")

#for specifics on repeat classes

head Repeats\_GRCh38\_sorted4col.bed

chr1 10000 10468 (TAACCC)n

chr1 10468 11447 TAR1

chr1 11504 11675 L1MC5a

chr1 11677 11780 MER5B

chr1 15264 15355 MIR3

chr1 15797 15849 (TGCTCC)n

chr1 16712 16744 (TGG)n

chr1 18906 19048 L2a

chr1 19971 20405 L3

chr1 20530 20679 Plat\_L3

bedtools multicov -bams P4191\_sorted.bam P4192\_sorted.bam -bed Repeats\_GRCh38\_sorted4col.bed > multicov\_RepeatClass.txt

#output

bedtools multicov -bams P4191\_sorted.bam P4192\_sorted.bam -bed Repeats\_GRCh38\_sorted4col.bed | head -1000 | sort -nrk 5 | head

chr1 632618 632685 tRNA-Ser-TCA(m) 267 749

chr1 180998 181980 TAR1 13 9

chr1 629498 629570 tRNA-Gln-CAA\_ 12 4

chr1 598734 599078 L1MC4a 6 8

chr1 520777 520989 GA-rich 4 10

chr1 286134 286178 (TG)n 3 0

chr1 433276 433374 L1PA8 2 0

chr1 433199 433297 L1PA8 2 0

chr1 433122 433220 L1PA8 2 0

chr1 162181 162393 GA-rich 2 10

#column 4 has the information about the repeat class