index fasta & bam

samtools faidx sequence1.fasta

samtools sort samp\_oar.bam > s\_samp\_oar.bam

samtools index s\_samp\_oar.bam

call genotypes

* -B option not to take read qual into account
* -A option to keep alt if both likely

bcftools mpileup -B -f fastafile -R bedfile bamfile | bcftools call -mV indels -A --ploidy 1 -o tr\_samp\_oar.vcf

reduce vcf

bcftools query -f '%CHROM %POS %REF %ALT\n' vcffile | cat | tr ' ' '\t' > redtr\_samp\_oar.bed

exclude transitions different from references

awk '!( $3 == "T" && $4 == "C" || $3 == "C" && $4 == "T" || $3 == "A" && $4 == "G" || $3 == "G" && $4 == "A" ) ' redtr\_samp\_oar.bed | awk '{print $1,$2-1,$2,$4}' | cat | tr ' ' '\t' > transv\_samp\_oar.bed

see the reads

bedtools bamtobed -i bamfile > output.bed

find shared reads with R script >> sharedoar.bed

intersect shared reads and target positions

\*-wb option to keep ones in a only present in b

bedtools intersect -a sharedoar.bed -b transv\_samp\_oar.bed -wb > int.bed

see each read as how many ref and alt

awk '{print $4,$10}' int.bed |awk '{gsub(/A|T|G|C/,"N",$2)}1' | awk '{gsub(/N,N/,"N",$2)}1'| sort | uniq -c |awk '{ print $1,'\t',$2,'\t',$3 }' > uq.bed