For controls:

First transform vcf to bed files and lift from mm9 to mm10 to mm39. Then VEP annotation.

Start with: Exome-seq\_Control\_2\_S2\_indels\_filtered.vcf

vcf file Exome-seq\_Control\_2\_S2\_indels\_filtered.vcf

vcf2bed < Exome-seq\_Control\_2\_S2\_indels\_filtered.vcf > Exome-seq\_Control\_2\_S2\_indels\_filtered.bed

you get that: Exome-seq\_Control\_2\_S2\_indels\_filtered.bed

then run correct\_bed.py after you install pandas: pip install pandas

correct\_bed.py:

import pandas as pd

# Load the original BED file

original\_bed = pd.read\_csv('/mnt/c/Users/agsko/dev/pcm/Exome-seq\_Control\_2\_S2\_indels\_filtered.bed', sep='\t', header=None)

# Create a simplified BED file with only the necessary columns for liftover

simplified\_bed = original\_bed.iloc[:, [0, 1, 2, 3]]

simplified\_bed.columns = ['CHROM', 'START', 'END', 'ID']

# Save the simplified BED file

simplified\_bed.to\_csv('/mnt/c/Users/agsko/dev/pcm/Exome-seq\_Control\_2\_S2\_indels\_filtered\_simplified.bed', sep='\t', index=False, header=False)

you get this: Exome-seq\_Control\_2\_S2\_indels\_filtered\_simplified.bed

then lift the bed from mm9 to mm10 and mm39 in this website: <https://genome.ucsc.edu/cgi-bin/hgLiftOver?hgsid=2313004406_Ql8wcYE3eOhUTakhHIeQ2CaHAtgr>

then you get: Exome-seq\_Control\_2\_S2\_indels\_filtered\_lifted\_simplified.bed

and now you need to get all the extra information from the Exome-seq\_Control\_2\_S2\_indels\_filtered.bed and merge them with the new one.

Merge.py:

import pandas as pd

# Paths to the BED files

original\_bed\_path = "/mnt/c/Users/agsko/dev/pcm/Exome-seq\_Control\_2\_S2\_indels\_filtered.bed"

lifted\_simplified\_bed\_path = "/mnt/c/Users/agsko/dev/pcm/Exome-seq\_Control\_2\_S2\_indels\_filtered\_lifted\_simplified.bed"

output\_merged\_bed\_path = "/mnt/c/Users/agsko/dev/pcm/Exome-seq\_Control\_2\_S2\_indels\_filtered\_lifted\_with\_info.bed"

# Load the original BED file

original\_bed = pd.read\_csv(original\_bed\_path, sep='\t', header=None)

original\_bed.columns = ['CHROM', 'START', 'END', 'ID', 'REF', 'ALT', 'QUAL', 'FILTER', 'INFO', 'FORMAT', 'NORMAL', 'TUMOR']

# Load the lifted simplified BED file

lifted\_bed = pd.read\_csv(lifted\_simplified\_bed\_path, sep='\t', header=None)

lifted\_bed.columns = ['CHROM', 'START', 'END', 'ID']

# Adjust positions: BED files are 0-based while VCF files are 1-based, so adjust the positions accordingly

lifted\_bed['START'] = lifted\_bed['START'] + 1

# Ensure both dataframes have the same length

if len(original\_bed) != len(lifted\_bed):

    raise ValueError("The original and lifted BED files do not have the same number of entries.")

# Merge the coordinates from the lifted BED into the original BED

merged\_bed = original\_bed.copy()

merged\_bed['CHROM'] = lifted\_bed['CHROM']

merged\_bed['START'] = lifted\_bed['START']

merged\_bed['END'] = lifted\_bed['END']

# Save the merged BED file with all the variant information

merged\_bed.to\_csv(output\_merged\_bed\_path, sep='\t', index=False, header=False)

print("Merged BED file has been created.")

you get: Exome-seq\_Control\_2\_S2\_indels\_filtered\_lifted\_with\_info.bed

then convert to vcf. bed\_to\_vcf1.py

import pandas as pd

# Load the merged BED file

merged\_bed\_path = "/mnt/c/Users/agsko/dev/pcm/Exome-seq\_Control\_2\_S2\_indels\_filtered\_lifted\_with\_info.bed"

output\_vcf\_path = "/mnt/c/Users/agsko/dev/pcm/Exome-seq\_Control\_2\_S2\_indels\_filtered\_lifted.vcf"

merged\_bed = pd.read\_csv(merged\_bed\_path, sep='\t', header=None)

merged\_bed.columns = ['CHROM', 'POS', 'END', 'ID', 'REF', 'ALT', 'QUAL', 'FILTER', 'INFO', 'FORMAT', 'NORMAL', 'TUMOR']

# Create a new DataFrame for the VCF

vcf\_df = merged\_bed[['CHROM', 'POS', 'ID', 'REF', 'ALT', 'QUAL', 'FILTER', 'INFO', 'FORMAT', 'NORMAL', 'TUMOR']]

# Load the original VCF header

original\_vcf\_path = "/mnt/c/Users/agsko/dev/pcm/Exome-seq\_Control\_2\_S2\_indels\_filtered.vcf"

with open(original\_vcf\_path, 'r') as file:

    vcf\_header = []

    for line in file:

        if line.startswith("#"):

            vcf\_header.append(line.strip())

        else:

            break

# Save the VCF file

with open(output\_vcf\_path, 'w') as file:

    # Write the VCF header

    for line in vcf\_header:

        file.write(line + '\n')

    # Write the VCF data

    vcf\_df.to\_csv(file, sep='\t', index=False, header=False)

print("VCF file has been created.")

then you get: Exome-seq\_Control\_2\_S2\_indels\_filtered\_lifted.vcf

then you run vep with frameshift, downstream and wildtype plugin:

perl /mnt/c/Users/agsko/dev/pcm/ensembl-vep/vep -i /mnt/c/Users/agsko/dev/pcm/Exome-seq\_Control\_2\_S2\_indels\_filtered\_lifted.vcf -o /mnt/c/Users/agsko/dev/pcm/Exome-seq\_Control\_2\_S2\_indels\_filtered\_lifted\_annotated.vep.vcf --vcf --symbol --terms SO --plugin Wildtype --plugin Downstream --plugin Frameshift --tsl --dir\_plugins /mnt/c/Users/agsko/dev/pcm/ensembl-vep/Plugins --dir\_cache /mnt/c/Users/agsko/dev/pcm/ensembl-vep/cache --fasta /mnt/c/Users/agsko/dev/pcm/vep/fasta/Mus\_musculus.GRCm39.dna.primary\_assembly.fa --species mus\_musculus --cache –offline

then install pvacseq – pip install pvactools, and pvactools –version

then install vatools: pip install vatools

then add a dummy sample using vcf-genotype-annotator: python genotype.py

import pandas as pd

# Paths to your files

vcf\_file = '/mnt/c/Users/agsko/dev/pcm/Exome-seq\_Control\_2\_S2\_indels\_filtered\_lifted\_annotated.vep.vcf'

output\_vcf\_file = '/mnt/c/Users/agsko/dev/pcm/Exome-seq\_Control\_2\_S2\_indels\_filtered\_lifted\_annotated\_corrected.vep.vcf'

# Open and process the VCF file

with open(vcf\_file, 'r') as infile, open(output\_vcf\_file, 'w') as outfile:

    for line in infile:

        if line.startswith('#'):

            outfile.write(line)

        else:

            fields = line.strip().split('\t')

            if not fields[5].replace('.', '', 1).isdigit():

                fields[5] = '.'

            outfile.write('\t'.join(fields) + '\n')

print("VCF file has been corrected and saved.")

command: vcf-genotype-annotator /mnt/c/Users/agsko/dev/pcm/Exome-seq\_Control\_2\_S2\_indels\_filtered\_lifted\_annotated\_corrected.vep.vcf DUMMY 0/1 -o /mnt/c/Users/agsko/dev/pcm/Exome-seq\_Control\_2\_S2\_indels\_filtered\_lifted\_annotated\_with\_dummy.vep.vcf

then generate protein fasta using pvacseq:

pvacseq generate\_protein\_fasta /mnt/c/Users/agsko/dev/pcm/Exome-seq\_Control\_2\_S2\_indels\_filtered\_lifted\_annotated\_with\_dummy.vep.vcf 17 /mnt/c/Users/agsko/dev/pcm/Control\_1\_S1\_output\_class1\_mutant\_peptides.txt --sample-name DUMMY

pvacseq generate\_protein\_fasta /mnt/c/Users/agsko/dev/pcm/Exome-seq\_Control\_2\_S2\_indels\_filtered\_lifted\_annotated\_with\_dummy.vep.vcf 29 /mnt/c/Users/agsko/dev/pcm/Control\_1\_S1\_output\_class2\_mutant\_peptides.txt --sample-name DUMMY