This script assesses whether any variants in the provided VCF files overlap with the target sequences of gRNAs designed from Task 2. It uses the reference genome, the variant data (in VCF format), and the target gene's exon location to determine if any variants might affect the efficacy or specificity of the gRNA sequences.

1. Loading Necessary Libraries

The libraries used in this script include:

- **`gzip`**: To handle compressed FASTA and VCF files.
- **`Bio.SeqIO`** and **`Bio.Seq`**: For parsing FASTA files and handling genomic sequences.
- **`pandas`**: To parse and handle VCF files efficiently.
- **`tqdm`**: To add progress bars for large file operations.

```
import gzip
import pandas as pd
from Bio import SeqIO
from Bio.Seq import Seq
from tqdm import tqdm
```

2. gRNA Sequences and File Paths

The gRNA sequences designed from Task 2 are listed. The file paths for the reference genome FASTA file and the VCF files are defined here.

```
# Define file paths
genome_fasta = "GCF_000001405.26_GRCh38_genomic.fna.gz"
vcf_files = ["sampled_chr1.vcf", "sampled_chr2.vcf"]
target_chromosome = "NC_000004.12"  # HPSE is on chromosome 4
output_file = "gRNA_variants_impact_results.txt"
```

```
# gRNA sequences from Task 2 exon provided to CHOPCHOP
gRNA_sequences = [
    "CATCTCCGCACCCTTCAAGTGGG", "CCATCTCCGCACCCTTCAAGTGG", "CCGCACCCTTCAAGTGGGTGTGG",
    "ATGGCCGGGATCCAAGCGCCCGG", "TCCAAGCGCCCGGGAGGCCTGGG", "CGCACCCTTCAAGTGGGTGTGGG",
    "CCTCCCGGGCGCTTGGATCCCGG", "ATCCAAGCGCCCGGGAGGCCTGG", "AATCACCCACACCCACTTGAAGG",
    "TGGCCGGGATCCAAGCGCCCGGG", "CCAAGCGCCCGGGAGGCCTGGGG", "ATCACCCACACCCACTTGAAGGG",
    "TGAAGGGTGCGGAGATGGCCGGG", "TTGAAGGGTGCGGAGATGGCCGG", "AGGCCTGGGGAGGAGCGCCCGGG",
    "GAGGCCTGGGGAGGAGCGCCCGG", "CGCTCCTCCCCAGGCCTCCCGGG", "GCGCTCCTCCCCAGGCCTCCCGG"
]
```

3. Loading the Reference Genome

The reference genome is loaded using Biopython's `SeqIO`. Each chromosome's sequence is stored in a dictionary for easy access during the analysis.

```python
# Load genome
def load_reference_genome(genome_fasta):
    genome = {}
    with gzip.open(genome_fasta, "rt") as handle:
        for record in tqdm(SeqIO.parse(handle, "fasta"), desc="Loading genome"):
            genome[record.id] = record.seq
            #print(f"Loaded chromosome: {record.id}")  # Print chromosome name
    return genome
```

4. Parsing VCF Files for Variants

The VCF files are parsed using `pandas`. The variants are stored in a dictionary where the chromosome is the key and the positions, reference alleles, and alternate alleles are stored as values.

```python
# Parse VCF files for variants
def load_variants_from_vcf(vcf_file):
    # Load the VCF file
    df = pd.read_csv(vcf_file, comment='#', sep='\t', header=None)

    # Select the first 8 columns (always present in vcf)
    df = df.iloc[:, :8]

    # Assign names
    base_columns = ['CHROM', 'POS', 'ID', 'REF', 'ALT', 'QUAL', 'FILTER', 'INFO']
    df.columns = base_columns

    # Extract relevant columns (chromosome, position, reference allele, alternate allele)
    variants = df[['CHROM', 'POS', 'REF', 'ALT']]

    # Convert the variants dataframe into a dictionary with chromosome as key
    variants_dict = {}
    for index, row in variants.iterrows():
        chrom = row['CHROM']
        pos = row['POS']
        ref = row['REF']
        alt = row['ALT']
        if chrom not in variants_dict:
            variants_dict[chrom] = []
        variants_dict[chrom].append((pos, ref, alt))

    return variants_dict
```

5. Checking if Variants Overlap with gRNA Sequences

This function checks whether any variants from the VCF files overlap with the gRNA sequences. If there is overlap, it prints a message indicating the affected gRNA sequence and variant.

```python
# Check if variants overlap gRNA target sequences
def check_variants_in_gRNA(gRNA_seq, chrom, start_pos, genome_seq, variants):
    end_pos = start_pos + len(gRNA_seq) - 1
    if chrom in variants:
        for variant in variants[chrom]:
            variant_pos = variant[0]
            if start_pos <= variant_pos <= end_pos:
                print(f"Variant {variant} overlaps with gRNA sequence {gRNA_seq}")
                return True
    return False
```

6. Main Function: Assessing the Impact of Variants on gRNAs

This function integrates the previous functions to assess whether the gRNA sequences overlap with any variants in the VCF files. Results are written to an output file.

```python
# Main function to assess the impact of variants on gRNA sequences
def assess_gRNA_impact(genome_fasta, vcf_files, gRNA_sequences, target_chromosome, output_file):
    genome = load_reference_genome(genome_fasta)

    # Open output file to write results
    with open(output_file, "w") as out_file:
        # Iterate over VCF files
        for vcf_file in tqdm(vcf_files, desc="Processing VCF files"):
            variants = load_variants_from_vcf(vcf_file)

            # Iterate over gRNA
            for gRNA_seq in tqdm(gRNA_sequences, desc="Checking gRNA sequences", leave=False):
                # Define the start positions of gRNA in the genome (first exon of gene)
                start_pos = genome[target_chromosome].find(gRNA_seq)
                if start_pos != -1:
                    result = check_variants_in_gRNA(gRNA_seq, target_chromosome, start_pos, genome, variants)
                    if result:
                        out_file.write(f"gRNA {gRNA_seq} may be impacted by variants. {result}\n")
                    else:
                        out_file.write(f"gRNA {gRNA_seq} is not affected by any variants.\n")
                else:
                    out_file.write(f"gRNA {gRNA_seq} not found in the target region.\n")
```