Assignment:

NGS, or Next-Generation Sequencing, is a powerful tool used in genomics to quickly and cheaply determine the sequence of nucleotides in DNA. However, one of the key challenges in NGS is aligning the millions of short reads that are generated to a reference genome.

In this assignment, you will implement a solution to the NGS alignment problem using Python. Specifically, you will write a program that takes in a reference genome and a set of short reads, and outputs the alignments of the reads to the reference genome.

To complete this assignment, you will need to do the following:

1. Write a function `score_alignment` that takes in two strings and returns the alignment score of the two strings. The alignment score is calculated by summing the number of matching characters and subtracting the number of mismatching characters. For example, the alignment of "ACGGT" and "ACGGC" would have a score of 4 (3 matching characters and 1 mismatching character).
2. Write a function `find_best_alignment` that takes in a reference string, a query string, and the scoring function from step 1, and returns the best alignment of the query string to the reference string. To find the best alignment, you can use a dynamic programming approach similar to the one used in the Smith-Waterman algorithm.
3. Write a function `align_reads` that takes in a reference genome and a list of query strings, and returns the alignments of the query strings to the reference genome. You can use the `find_best_alignment` function from step 2 to find the best alignment for each query string.
4. Test your implementation on a small example to ensure that it is correct. For example, you could use the following reference genome and query strings:

```
reference = "GATCGTGGCTCTAGA" queries = ["GATC", "GGCT", "CTAG"]
```

The expected output of the `align_reads` function for this example would be a list of alignments, such as:

```
[("GATC", "GATC", 4), ("GGCT", "GGCT", 4), ("CTAG", "CTAG", 4)]
```

Submitting the Assignment:

To submit your assignment, you can either email it to your instructor or upload it to a class website. Please include your code, as well as any test cases or explanations of how you tested your implementation.