



**DNA SEQUENCE PATTERN ANALYSIS USING HADOOP  
MAPREDUCE(Demonstration of Word Count program)**

**SANJAY R**

**2348055**

**22<sup>nd</sup> October 2024**

**MDS571**

**Big Data Analytics**

**Department of Statistics and Data Science**

# CHRIST (Deemed to be University)

## Department of Statistics and Data Science

Course:MDS571-Big Data Analytics

Lab :2

R.SANJAY

2348055

### *DNA SEQUENCE PATTERN ANALYSIS USING HADOOP MAPREDUCE*

*(Demonstration of Word Count program)*

#### ✧ INTRODUCTION:

In the intricate world of genetics, DNA sequences serve as the fundamental blueprint for all living organisms. Each DNA molecule is composed of a series of four nucleotides, represented by the letters A (adenine), T (thymine), C (cytosine), and G (guanine). Understanding the patterns within these sequences is crucial for a variety of scientific disciplines, including genomics, bioinformatics, and medical research. The advancement of sequencing technologies has led to the generation of vast amounts of DNA sequence data, creating opportunities for deep analysis and discovery.

This report explores the use of Apache Hadoop, a powerful framework for processing large datasets, to analyze DNA sequences specifically by identifying hexamer patterns—sequences composed of six nucleotides. The goal is to detect and quantify these patterns across vast DNA datasets, which can provide insights into genetic functions and potentially reveal links to genetic disorders or characteristics. Processing DNA sequences presents unique challenges due to the sheer volume of data generated. Therefore, this project adapts a classic program, the Hadoop Word Count, transforming it to identify hexamer patterns. By leveraging Hadoop's

distributed computing capabilities, we can efficiently process large-scale DNA data, ultimately revealing valuable information about the genetic sequences.

#### ✧ **PROBLEM DESCRIPTION:**

The primary problem addressed in this project is the identification and quantification of hexamer patterns within extensive DNA sequences. Each hexamer, a sequence of six nucleotides, is treated as a unique "word" in this context. Identifying these patterns is vital because they can signify functional roles in genes, including coding for proteins or regulating genetic expression. Genetic research often focuses on specific sequences or markers, making the identification of these hexamers crucial for understanding the underlying mechanisms of biological processes.

The challenge arises from the need to process vast amounts of DNA sequence data efficiently. Traditional data processing methods often struggle to handle such large datasets, leading to long processing times and the potential for errors. The volume of data generated from modern sequencing technologies can be overwhelming, necessitating a solution that is not only scalable but also efficient. By utilizing the Hadoop framework, this project aims to create a scalable solution capable of analyzing significant volumes of DNA data, counting the occurrences of hexamer patterns, and providing a clear output of their frequencies and distributions.

#### ✧ **DATASET DESCRIPTION:**

The dataset for this project can be sourced from publicly available genetic databases, such as the National Center for Biotechnology Information (NCBI) or generated through simulations of DNA sequences. These databases provide rich resources for genomic data, making them invaluable for research. The availability of diverse datasets allows for comprehensive analysis, enabling researchers to compare patterns across different organisms or conditions.

The DNA sequence data is formatted as plain text files, where each line contains a continuous sequence of nucleotides. This straightforward format facilitates easy parsing and processing during analysis. Each nucleotide sequence may represent an

entire genome, a specific gene, or even experimental data from studies, depending on the chosen dataset. Before analysis, the DNA sequences undergo preprocessing to ensure accuracy. This includes converting all nucleotides to uppercase to avoid discrepancies in pattern recognition due to casing differences, and trimming any leading or trailing spaces, as well as extraneous characters, to ensure that only valid nucleotide sequences are analyzed. These preprocessing steps are crucial to maintaining the integrity of the data and ensuring that hexamer patterns are accurately identified and counted.

#### ✧ CODE:

```
import java.io.IOException;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;

// Import necessary Hadoop libraries for configuration, job setup, and MapReduce
// framework
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

/**
 * HexamerAnalysis: This MapReduce program is designed to analyze DNA
 sequences,
 * specifically focusing on counting and calculating the frequency distribution of
 * 6-nucleotide sequences (hexamers) in large datasets.
 * The program extracts each hexamer from DNA sequence data, counts
 occurrences,
 * and provides the percentage of each pattern relative to all hexamers.
 */
public class HexamerAnalysis {

    /**
     * Mapper Class: HexamerMapper
     * This class processes lines of DNA sequences from the input dataset. For
     each

```

**a**

- \* line, it extracts 6-nucleotide sequences (hexamers) and outputs each with
- \* count of 1. This serves as the input for the reducer, where counts for each
- \* unique hexamer are aggregated.

\*/

```
public static class HexamerMapper extends Mapper<Object, Text, Text,
IntWritable> {
    // Initialize an IntWritable with a constant value of 1, representing each
occurrence of a hexamer
    private final static IntWritable one = new IntWritable(1);
    // Text object to hold hexamer sequences
    private Text hexamer = new Text();

    @Override
    public void map(Object key, Text value, Context context) throws
IOException, InterruptedException {
        // Convert the DNA sequence line to uppercase to ensure uniformity
across patterns
        String line = value.toString().toUpperCase().trim();
        int length = line.length();

        // Iterate through the line to extract each 6-nucleotide sequence
for (int i = 0; i <= length - 6; i++) {
            // Extract the hexamer starting from index i with a length of 6
characters

            String hex = line.substring(i, i + 6);
            // Set the hexamer in the Text object
            hexamer.set(hex);
            // Emit the hexamer with an occurrence count of 1
            context.write(hexamer, one);
        }
        // This map function thus produces a <hexamer, 1> output for each
hexamer in each line,
        // setting up the reducer to aggregate counts of each unique hexamer.
    }
}
```

- \* **Reducer Class: HexamerReducer**
- \* The reducer aggregates the counts for each hexamer pattern. It maintains a total count
- \* of all hexamers, calculates the occurrence percentage of each hexamer, sorts hexamers
- \* by count in descending order, and outputs the hexamer pattern, count, and percentage.

\*/

```
public static class HexamerReducer extends Reducer<Text, IntWritable,
Text, Text> {
    // Total count of all hexamers to calculate percentage later
```

```

private int totalHexamers = 0;
// HashMap to store each unique hexamer and its total count
private HashMap<Text, Integer> hexamerCounts = new HashMap<>();
// Variable to ensure that the header is written once to the output
private boolean headerWritten = false;

@Override
public void reduce(Text key, Iterable<IntWritable> values, Context context)
    throws IOException, InterruptedException {
    int sum = 0;

    // Aggregate counts for each hexamer by summing values
    for (IntWritable val : values) {
        sum += val.get();
    }

    // Add this hexamer's count to the overall total
    totalHexamers += sum;

    // Store the hexamer pattern and its total count in the map
    hexamerCounts.put(new Text(key), sum);
}

// Cleanup function is executed after all reduce calls, finalizing output
@Override
protected void cleanup(Context context) throws IOException,
InterruptedException {
    // Convert HashMap to a List for sorting by hexamer count in
    descending order
    List<Map.Entry<Text, Integer>> hexamerList = new
    ArrayList<>(hexamerCounts.entrySet());

    // Sort list so hexamers with the highest occurrences are listed first
    Collections.sort(hexamerList, new Comparator<Map.Entry<Text,
    Integer>>() {
        @Override
        public int compare(Map.Entry<Text, Integer> o1,
        Map.Entry<Text, Integer> o2) {
            return o2.getValue().compareTo(o1.getValue());
        }
    });

    // Write a header line if it hasn't already been written
    if (!headerWritten) {
        context.write(new Text("Pattern/Sequence"), new Text("No of
        occurrences\tPercentage"));
        headerWritten = true;
    }
}

```

```

        // For each hexamer pattern, calculate the percentage and write it to
output
        for (Map.Entry<Text, Integer> entry : hexamerList) {
            Text hexamer = entry.getKey();
            int count = entry.getValue();

            // Calculate percentage of this hexamer relative to all hexamers
            double percentage = (count * 100.0) / totalHexamers;

            // Output the hexamer, occurrence count, and percentage in a
formatted string
            context.write(hexamer, new Text(count + "\t" +
String.format("%.2f", percentage) + "%"));
        }
        // The cleanup function thus finalizes the reducer's output, providing a
sorted, formatted list
        // of hexamers with their counts and percentage representation.
    }
}

```

**\* Driver Class (Main): Sets up and initiates the Hadoop job.**  
**\* This main method validates input, configures the job, and defines the mapper**  
**\* and reducer classes, as well as input and output paths.**  
**\*/**

```

public static void main(String[] args) throws Exception {
    // Check that the correct number of arguments are provided for input and
output paths
    if (args.length != 2) {
        System.err.println("Usage: HexamerAnalysis <input path> <output
path>");
        System.exit(-1);
    }

    // Set up configuration for Hadoop job
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "Hexamer Analysis");

    // Specify the main class, mapper, and reducer classes for the job
    job.setJarByClass(HexamerAnalysis.class);
    job.setMapperClass(HexamerMapper.class);
    job.setReducerClass(HexamerReducer.class);

    // Define the output data types for keys and values
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    // Define the input and output paths from command-line arguments

```

```

FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));

// Execute the job and exit with code 0 if successful, 1 if an error occurs
System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}

```

## ✧ PROGRAM DESCRIPTION:

This initial block of code imports necessary Java libraries along with Hadoop-specific classes. These imports include utilities for handling input/output, map-reduce functionality, and collection classes like HashMap, ArrayList, and Comparator. These are required for configuring the Hadoop job, implementing the mapper and reducer functions, and managing hexamer counting and sorting.

The main class, HexamerAnalysis, encapsulates the entire program. It contains three main components: the HexamerMapper class, the HexamerReducer class, and the main driver method. Each component handles a specific part of the map-reduce process, managing data extraction, counting, and output formatting for DNA hexamer patterns.

The HexamerMapper class extends Hadoop's Mapper class. In this class, each DNA sequence is processed to find 6-nucleotide patterns (hexamers). The hexamer variable is defined as a Text object to store each 6-nucleotide substring, and one is an IntWritable constant used to signify each occurrence.

In the map method, each line of input is converted to uppercase and trimmed of whitespace. The method then iterates through each DNA sequence to extract 6-character substrings, or hexamers. For each hexamer, it emits a key-value pair where the hexamer is the key, and 1 is the value, representing a single occurrence of that hexamer. This approach enables the subsequent reducer step to count occurrences across all input lines.

The HexamerReducer class extends Hadoop's Reducer class and handles the aggregation of hexamer counts. The variable totalHexamers tracks the total count of



all hexamers processed, and hexamerCounts stores each hexamer with its aggregated count. headerWritten ensures that a header row is outputted once for clarity.

Within the reduce method, the reducer aggregates the values for each hexamer key by summing the occurrences from the mapper output. Each unique hexamer's count is stored in hexamerCounts, while totalHexamers keeps a cumulative count of all hexamers, which is necessary for calculating percentages later in the cleanup step.

The cleanup method runs after the reduce method completes. It converts hexamerCounts to a list, sorting it in descending order based on occurrences for more readable output. The context.write method outputs each hexamer sequence, count, and calculated percentage. The percentage is computed by dividing each hexamer's count by totalHexamers and converting it to a percentage format. The header is written once to label the columns for clarity.

The main method sets up and configures the MapReduce job for Hadoop. It verifies the input and output arguments, which specify the paths for the DNA dataset and the output directory. It configures the Hadoop job to use HexamerMapper for mapping and HexamerReducer for reducing, setting the output types as Text for the hexamer keys and IntWritable for the count values. The job then executes within the Hadoop environment, exiting with a status indicating success or failure.

## ✧ PROJECT SETUP AND EXECUTION:

### ➤ Dataset Preparation:

The initial step in conducting the hexamer analysis involved preparing the dataset. The data was stored in a text file named **Lab2\_data**, which contained DNA sequences organized specifically for this analysis.

### ✧ Lab2\_data:

A screenshot of a text editor window titled "Lab2\_data" with a file path of "~/Lab2/Lab\_2". The window displays three lines of DNA sequences, each preceded by a line number (1, 2, 3) in the left margin. Line 1: ATGCCACAGCTAGATACATCCACCTGATTATTATATCTTTTCATATTTCTCACCTCTTCATCCTATTTCAACTAAAAATTTCAATCACTACTACCCAGAAAACCCGATACCAATCTGCTAAAAATTGCTGGTCAACATAATCTTGAGAAAACAAATGAACAAAATCTATTGCTTTCTTGCTGCCCTCAAT. Line 2: ATGTGCACTAAATGAACAGCCCTTTACACGACGACTCATACGAGCGGGGATACGGCGGGCTCCGGCGGCTTTCTCTACAGACTACAARCTCTGAACCCAGCTGGCGCTCAACCTGGCCGACCCCTACCGAAGTCTCAAAAGCCCCGGGGCGGGGCCGACGAGGCGAGCGGTGGCAGCAGCT. Line 3: ATGAGCCGGCAGCTAAACAGAAGCCAGAACTGCTCTTCAGTGACGTCGATGAGCTGATGAAGAGCGGTGCAGTTGGCTGTCCACATCCCCACCTTCTCTGGGCTCTCTCTCAACCTGCTGGCCATCCGAGGCTTCAGACCTTCTTGAGGAAGAGGAGGTGGCGGATTATGCCGCCACCGCCATCTACATGATCAACT. The window includes standard menu options (Open, Save) and window control buttons (minimize, maximize, close) in the top right corner.

The dataset file was placed within the Hadoop directory structure at **/MDS2024/Datasets/LAB2**, ensuring easy access and integration with the Hadoop ecosystem for processing. This setup was critical to enable seamless data flow during the execution of the MapReduce job.

```
hadoop@Ubuntu22:~$ start-dfs.sh
Starting namenodes on [localhost]
localhost: namenode is running as process 4215. Stop it first and ensure /tmp/hadoop-hadoop-namenode.pid file is empty before retry.
Starting datanodes
localhost: datanode is running as process 4340. Stop it first and ensure /tmp/hadoop-hadoop-datanode.pid file is empty before retry.
Starting secondary namenodes [Ubuntu22]
Ubuntu22: secondarynamenode is running as process 4587. Stop it first and ensure /tmp/hadoop-hadoop-secondarynamenode.pid file is empty before retry.
hadoop@Ubuntu22:~$ start-yarn.sh
Starting resource manager
resource manager is running as process 5079. Stop it first and ensure /tmp/hadoop-hadoop-resource manager.pid file is empty before retry.
Starting node managers
localhost: nodemanager is running as process 5234. Stop it first and ensure /tmp/hadoop-hadoop-nodemanager.pid file is empty before retry.
hadoop@Ubuntu22:~$ jps
5234 NodeManager
4340 DataNode
5079 ResourceManager
4215 NameNode
4587 SecondaryNameNode
12428 Jps
11503 org.eclipse.equinox.launcher_1.6.900.v20240613-2009.jar
```

```
hadoop@Ubuntu22:~$ hadoop fs -ls /MDS2024/Datasets
Found 3 items
drwxr-xr-x - hadoop supergroup 0 2024-10-24 23:54 /MDS2024/Datasets/LAB2
-rw-r--r-- 3 hadoop supergroup 541 2024-10-24 08:22 /MDS2024/Datasets/combiner
-rw-r--r-- 3 hadoop supergroup 295 2024-10-22 10:40 /MDS2024/Datasets/word
hadoop@Ubuntu22:~$ hadoop fs -ls /MDS2024/Datasets/Lab2_data
ls: '/MDS2024/Datasets/Lab2_data': No such file or directory
hadoop@Ubuntu22:~$ hadoop fs -ls /MDS2024/Datasets/LAB2
Found 1 items
-rw-r--r-- 3 hadoop supergroup 3947 2024-10-24 23:54 /MDS2024/Datasets/LAB2/Lab2_data
```

## ✧ Eclipse Setup:

For the implementation, the project was set up in Eclipse, a widely used integrated development environment (IDE) for Java development. The following steps were followed to prepare the project:

- A new Java project named LAB2 was created in Eclipse, setting the foundation for developing the MapReduce application for hexamer analysis.
- **Main Class and MapReduce Components:** Within the project, a main class named HexamerAnalysis was implemented, containing the Mapper and Reducer classes:
  - The HexamerMapper class extracted every possible 6-nucleotide substring, or hexamer, from each line in the input data. For each hexamer found, it emitted the sequence with a count of 1, which was used in the aggregation step.
  - The HexamerReducer class consolidated the counts for each hexamer, calculating total occurrences and determining the percentage of each

hexamer relative to the overall count. The data was then organized into a HashMap, sorted by occurrence frequency, and written to the output in a structured format.

- After the code was developed and thoroughly tested, the necessary Hadoop library JAR files were added to the project to ensure Hadoop-specific functionalities during execution. The project was then exported as a JAR file named HexamerAnalysis.jar, making it ready for deployment within the Hadoop environment.

### ✧ Execution Process:

- The next step involved executing the program within the Hadoop environment, using the following command:

```
$ hadoop jar /home/hadoop/Labs/Jar_Files/HexamerAnalysis.jar HexamerAnalysis /MDS2024/Datasets/LAB2/Lab2_data /MDS2024/Outputs/Lab2_Output
```

```
hadoop@ubuntu22:~$ hadoop jar /home/hadoop/Labs/Jar_Files/HexamerAnalysis.jar HexamerAnalysis /MDS2024/Datasets/LAB2/Lab2_data /MDS2024/Outputs/Lab2_Output
2024-10-26 10:15:09,093 INFO Impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2024-10-26 10:15:09,203 INFO Impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2024-10-26 10:15:09,203 INFO Impl.MetricsSystemImpl: JobTracker metrics system started
2024-10-26 10:15:09,382 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to re
2024-10-26 10:15:09,626 INFO Input.FileInputFormat: Total input files to process : 1
2024-10-26 10:15:09,680 INFO mapreduce.JobSubmitter: number of splits:1
2024-10-26 10:15:09,873 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1872796646_0001
2024-10-26 10:15:09,873 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-10-26 10:15:09,978 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2024-10-26 10:15:09,999 INFO mapreduce.Job: Running job: job_local1872796646_0001
2024-10-26 10:15:10,020 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2024-10-26 10:15:10,054 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2024-10-26 10:15:10,055 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2024-10-26 10:15:10,055 INFO output.FileOutputCommitter: skip cleanup_temporary folders under output directory:false, ignore cleanup failures: false
2024-10-26 10:15:10,069 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2024-10-26 10:15:10,172 INFO mapred.LocalJobRunner: Waiting for map tasks
2024-10-26 10:15:10,172 INFO mapred.LocalJobRunner: Starting task: attempt_local1872796646_0001_m_000000_0
2024-10-26 10:15:10,281 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2024-10-26 10:15:10,282 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2024-10-26 10:15:10,282 INFO output.FileOutputCommitter: skip cleanup_temporary folders under output directory:false, ignore cleanup failures: false
2024-10-26 10:15:10,349 INFO mapred.Task: Using ResourceCalculatorProcessTree: [ ]
2024-10-26 10:15:10,368 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/MDS2024/Datasets/LAB2/Lab2_data/0+3947
2024-10-26 10:15:10,500 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2024-10-26 10:15:10,500 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2024-10-26 10:15:10,500 INFO mapred.MapTask: soft limit at 83886080
2024-10-26 10:15:10,500 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2024-10-26 10:15:10,501 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2024-10-26 10:15:10,505 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2024-10-26 10:15:10,663 INFO mapred.LocalJobRunner:
2024-10-26 10:15:10,665 INFO mapred.MapTask: Starting flush of map output
2024-10-26 10:15:10,665 INFO mapred.MapTask: Spilling map output
2024-10-26 10:15:10,665 INFO mapred.MapTask: bufstart = 0; bufend = 43854; bufvoid = 104857600
2024-10-26 10:15:10,665 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26198744(104794976); length = 15653/6553600
2024-10-26 10:15:10,684 INFO mapred.MapTask: Finished spill 0
2024-10-26 10:15:10,713 INFO mapred.Task: Task:attempt_local1872796646_0001_m_000000_0 is done. And is in the process of committing
2024-10-26 10:15:10,738 INFO mapred.LocalJobRunner: map
2024-10-26 10:15:10,738 INFO mapred.Task: Task 'attempt_local1872796646_0001_m_000000_0' done.
2024-10-26 10:15:10,752 INFO mapred.Task: Final Counters for attempt_local1872796646_0001_m_000000_0: Counters: 23
File System Counters
  FILE: Number of bytes read=5275
  FILE: Number of bytes written=697150
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=3947
  HDFS: Number of bytes written=0
  HDFS: Number of read operations=5
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=1
UNEC: Number of bytes read=arsene-rated=0
```

- Upon execution, this command initiated the Hadoop job to process the DNA sequences, and the output was saved in the designated output directory. To verify

successful job completion, a quick directory listing was conducted using the following command:

**\$ hadoop fs -ls MDS2024/Outputs/Lab2\_Output/**

```
bytes written=31248
hadoop@Ubuntu22:~$ hadoop fs -ls /MDS2024/Outputs/Lab2_Output
Found 2 items
-rw-r--r--  3 hadoop supergroup      0 2024-10-26 10:15 /MDS2024/Outputs/Lab2_Output/_SUCCESS
-rw-r--r--  3 hadoop supergroup 31248 2024-10-26 10:15 /MDS2024/Outputs/Lab2_Output/part-r-000000
hadoop@Ubuntu22:~$
```

- The output displayed two items:
- `_SUCCESS`: A file indicating the successful completion of the job.
- `part-r-000000`: A file containing the processed output data.

To examine the contents of the output, the following command was issued:

**\$ hadoop fs -cat /MDS2024/Outputs/Lab2\_Output/part-r-000000**

Pattern/Sequence			No of occurrences	Percentage
GCCCCC	12	0.31%		
GGCGGC	11	0.28%		
GAGCGC	9	0.23%		
GCCACC	9	0.23%		
GCTCAA	9	0.23%		
CGCCCC	8	0.20%		
GGAGCG	8	0.20%		
CTGGGC	8	0.20%		
GGCGCC	8	0.20%		
CCTTCC	8	0.20%		
GCGGCC	8	0.20%		
GCGGCG	8	0.20%		
CGGGGC	7	0.18%		
GAGCAG	7	0.18%		
CCCTTC	7	0.18%		
AGCAGG	7	0.18%		
CCGGGG	7	0.18%		
CCCCCG	7	0.18%		
GGCAGC	7	0.18%		
GCTGCC	7	0.18%		
CTCAAG	7	0.18%		
AGCGCA	7	0.18%		
CGGGGG	6	0.15%		
CCCGGG	6	0.15%		
GCGCCC	6	0.15%		
AGGAGC	6	0.15%		
CTGCTG	6	0.15%		
GGTGGC	6	0.15%		
CGGCTG	6	0.15%		
CGGCGG	6	0.15%		
CCCCAG	6	0.15%		
TCCTGG	6	0.15%		
CTGGAG	6	0.15%		
TCAATA	6	0.15%		
CCACCT	6	0.15%		
CCAGCT	6	0.15%		
CAGCTA	6	0.15%		
GGGCGC	6	0.15%		
AGGCGC	6	0.15%		
CCTGGG	6	0.15%		
CCGACG	6	0.15%		
CCTCCT	6	0.15%		
CCCAGG	6	0.15%		
AGCTGG	6	0.15%		
GGCTGC	6	0.15%		

## ✧ CONCLUSION:

The resulting output revealed each hexamer pattern alongside its count of occurrences and the percentage representation relative to the total hexamers processed. This structured output provided valuable insights into the distribution of hexamer patterns within the dataset, highlighting frequently occurring sequences and potentially significant motifs within the DNA data.

## ✧ OUTPUT:

Pattern/Sequence	No of occurrences	Percentage
GCCCCC	12	0.31%
GGCGGC	11	0.28%
GAGCGC	9	0.23%
GCCACC	9	0.23%
GCTCAA	9	0.23%
CGCCCC	8	0.20%
GGAGCG	8	0.20%
CTGGGC	8	0.20%
GGCGCC	8	0.20%
CCTTCC	8	0.20%
GCGGCC	8	0.20%
GCGGCG	8	0.20%
CGGGGC	7	0.18%
GAGCAG	7	0.18%
CCCTTC	7	0.18%
AGCAGG	7	0.18%
CCGGGG	7	0.18%

CCCCCG	7	0.18%
GGCAGC	7	0.18%
GCTGCC	7	0.18%
CTCAAG	7	0.18%
AGCGCA	7	0.18%
CGGGGG	6	0.15%
CCCGGG	6	0.15%
GCGCCC	6	0.15%
AGGAGC	6	0.15%
CTGCTG	6	0.15%
GGTGGC	6	0.15%
CGGCTG	6	0.15%
CGGCGG	6	0.15%
CCCCAG	6	0.15%
TCCTGG	6	0.15%
CTGGAG	6	0.15%
TCAATA	6	0.15%
CCACCT	6	0.15%
CCAGCT	6	0.15%
CAGCTA	6	0.15%
GGGCGC	6	0.15%
AGGCGC	6	0.15%
CCTGGG	6	0.15%
CCGACG	6	0.15%
CCTCCT	6	0.15%

CCCAGG	6	0.15%
AGCTGG	6	0.15%
GGCTGC	6	0.15%
AGCGGC	6	0.15%
GTCGCC	6	0.15%
CCCGAC	5	0.13%
CGGCCA	5	0.13%
GGAGCA	5	0.13%
CGCCAC	5	0.13%
CGCTCA	5	0.13%
CGCCGG	5	0.13%
GCGCCT	5	0.13%
GCGCAA	5	0.13%
GAGGAG	5	0.13%
GGACAC	5	0.13%
CGCGCC	5	0.13%
GAAGAC	5	0.13%
TGCTGC	5	0.13%
TGGGCC	5	0.13%
TTCGCT	5	0.13%
CAGCGG	5	0.13%
CTAATC	5	0.13%
CACGCC	5	0.13%
TGGCGG	5	0.13%
TCCTCC	5	0.13%

TCAAGC	5	0.13%
TCAAGG	5	0.13%
CAGGAG	5	0.13%
CCCCCT	5	0.13%
CTGGCC	5	0.13%
CTGGCG	5	0.13%
CCCCGG	5	0.13%
CCAGGA	5	0.13%
CGACGG	5	0.13%
AACCTG	5	0.13%
GCTGCG	5	0.13%
GGCGCG	5	0.13%
TTTATT	5	0.13%
GCTGTC	5	0.13%
GCCGCG	5	0.13%
GCCGCC	5	0.13%
ACCTGG	5	0.13%
GGCTTC	5	0.13%
AGGTGG	5	0.13%
CTGAGC	5	0.13%
CCGAGG	5	0.13%
CCCAGC	5	0.13%
CCTGAG	5	0.13%
GACGGC	5	0.13%
CTACTT	5	0.13%



GGAAGA	5	0.13%
TGGAGC	5	0.13%
GGGGCC	5	0.13%
GGGGCG	5	0.13%
CGGAGC	5	0.13%
GCCGGG	5	0.13%
GAGCCT	4	0.10%
CGGCCT	4	0.10%
CGCCCA	4	0.10%
CGCCCG	4	0.10%
GCGCCG	4	0.10%
GAGGCG	4	0.10%
CGGGCG	4	0.10%
TTCATC	4	0.10%
GCGCAT	4	0.10%
ATGGAA	4	0.10%
AACAGC	4	0.10%
CGCGGG	4	0.10%
GCGCTG	4	0.10%
ATGGCT	4	0.10%
CCTAAT	4	0.10%
CAGCCA	4	0.10%
GCAGGA	4	0.10%
CTTCTT	4	0.10%
CTGCTC	4	0.10%

TGCCCC	4	0.10%
GCAGCT	4	0.10%
GCAGCC	4	0.10%
CAGGTG	4	0.10%
GCAGAA	4	0.10%
CCGCGG	4	0.10%
TGAAGA	4	0.10%
GGTCGC	4	0.10%
AGCAGA	4	0.10%
CTCCTC	4	0.10%
CTCCTG	4	0.10%
GCCAGC	4	0.10%
CTGCGC	4	0.10%
TGGCCA	4	0.10%
CCGGGA	4	0.10%
TCAAAG	4	0.10%
TCAACC	4	0.10%
TCAACT	4	0.10%
CCGCCA	4	0.10%
ACAGCT	4	0.10%
TTCTAC	4	0.10%
CGGCGC	4	0.10%
CCCCAA	4	0.10%
CCAACG	4	0.10%
GGAGTG	4	0.10%

CGCCTG	4	0.10%
CCCTCT	4	0.10%
CCCCCC	4	0.10%
CACCAT	4	0.10%
CCGGCC	4	0.10%
TCGCCC	4	0.10%
TTCCTC	4	0.10%
TCGCCT	4	0.10%
CCATCT	4	0.10%
CCACCG	4	0.10%
GCCATC	4	0.10%
GCTTCG	4	0.10%
GCTGGC	4	0.10%
GCTGGA	4	0.10%
GCTGGG	4	0.10%
CAAGCT	4	0.10%
GCTGCT	4	0.10%
TCTACA	4	0.10%
CGCAAG	4	0.10%
CTTCAT	4	0.10%
G TTCCT	4	0.10%
CAGCTG	4	0.10%
CTTCCT	4	0.10%
CGAGGA	4	0.10%
TCGCTG	4	0.10%

CTTCGC	4	0.10%
ACGCCC	4	0.10%
TCTATT	4	0.10%
CGAGCA	4	0.10%
ACGGCG	4	0.10%
GTGGCG	4	0.10%
GGGCCC	4	0.10%
GGGCCT	4	0.10%
GGGCCG	4	0.10%
GGGCGG	4	0.10%
AGCCTG	4	0.10%
CCTGGC	4	0.10%
TCAGCA	4	0.10%
ACCCAG	4	0.10%
CCTGGT	4	0.10%
ACCTCC	4	0.10%
CCTCAA	4	0.10%
CATCAA	4	0.10%
ATTCCC	4	0.10%
ATGAGC	4	0.10%
CCTCCA	4	0.10%
CAGAAC	4	0.10%
CAGAAG	4	0.10%
GCGGGG	4	0.10%
CAACCT	4	0.10%

GCCCCGA	4	0.10%
CAATAT	4	0.10%
TAATTC	4	0.10%
GCCTGG	4	0.10%
GCCTGA	4	0.10%
GGCCCC	4	0.10%
GCCCCA	4	0.10%
CCACGC	4	0.10%
AGCTCG	4	0.10%
GGCTGT	4	0.10%
AGCCAC	4	0.10%
GGCCGC	4	0.10%
CGCATC	4	0.10%
CCAGTC	4	0.10%
CTCAAC	4	0.10%
AGCGAG	3	0.08%
GACCTG	3	0.08%
TCTGCT	3	0.08%
CTCCAG	3	0.08%
CTAAAA	3	0.08%
CTCGGG	3	0.08%
CGGCCG	3	0.08%
CGGCCC	3	0.08%
TCTGGA	3	0.08%
GCGGTG	3	0.08%

ACTTTT	3	0.08%
TAATCT	3	0.08%
CGGCAG	3	0.08%
CCCGGC	3	0.08%
TAATCC	3	0.08%
GCGCGC	3	0.08%
CTCGCC	3	0.08%
GAGGGC	3	0.08%
GGAGCT	3	0.08%
TGTCTT	3	0.08%
GGAGCC	3	0.08%
TCTTCT	3	0.08%
GCAATT	3	0.08%
TTTTAT	3	0.08%
TCCATT	3	0.08%
TTTCAA	3	0.08%
AAGACG	3	0.08%
CGCCGC	3	0.08%
AATTCA	3	0.08%
GGAGGA	3	0.08%
AATTCC	3	0.08%
CGGGAC	3	0.08%
GACTGG	3	0.08%
CGGGAG	3	0.08%
CGCTGC	3	0.08%

CGCTGA	3	0.08%
CGCTGG	3	0.08%
ACCATC	3	0.08%
CGCGCT	3	0.08%
AGTCTA	3	0.08%
GGGCTG	3	0.08%
GTCTTC	3	0.08%
GAGCTC	3	0.08%
CGCGGC	3	0.08%
GAGCTG	3	0.08%
ATCTAC	3	0.08%
TCTTTC	3	0.08%
AACACC	3	0.08%
ACTGCT	3	0.08%
CATCTG	3	0.08%
ACTGGG	3	0.08%
GAGGTG	3	0.08%
GCGCTC	3	0.08%
CCTCTG	3	0.08%
TTTCTC	3	0.08%
ACCACG	3	0.08%
ATAATC	3	0.08%
CTTTTA	3	0.08%
TGCGGG	3	0.08%
CAGTCT	3	0.08%

CTTCTA	3	0.08%
ATCCTA	3	0.08%
CAGCAG	3	0.08%
GTCAAC	3	0.08%
CAGCAA	3	0.08%
GCAGGT	3	0.08%
AGGAGG	3	0.08%
CCGTTC	3	0.08%
AAATCT	3	0.08%
ACCAAA	3	0.08%
GGGACA	3	0.08%
AGGACA	3	0.08%
GCAGCG	3	0.08%
TGCTGG	3	0.08%
GGTCCA	3	0.08%
GCAGAC	3	0.08%
GGGAGC	3	0.08%
CCGCGC	3	0.08%
CCCCTA	3	0.08%
CCCCTC	3	0.08%
CTGGTC	3	0.08%
ACACGC	3	0.08%
CTGGTG	3	0.08%
CACCTC	3	0.08%
CACCTT	3	0.08%



CACCTG	3	0.08%
CTCCTT	3	0.08%
ACACCC	3	0.08%
TACCTT	3	0.08%
CTAATT	3	0.08%
TACTTT	3	0.08%
TGGCCC	3	0.08%
TACTTA	3	0.08%
TTCCCT	3	0.08%
TGGTCA	3	0.08%
TTCCCC	3	0.08%
AATATT	3	0.08%
GCCACG	3	0.08%
TCCGGT	3	0.08%
TGCTTC	3	0.08%
CTGCCC	3	0.08%
AGCAAT	3	0.08%
TCAACA	3	0.08%
CACGAC	3	0.08%
CCGCCC	3	0.08%
TCCCCA	3	0.08%
CCGCCG	3	0.08%
CTGGGT	3	0.08%
CCCTAT	3	0.08%
CGGCGA	3	0.08%

AGACGG	3	0.08%
CACCGG	3	0.08%
CGCCTC	3	0.08%
GCACGG	3	0.08%
GGCACC	3	0.08%
CAGGAC	3	0.08%
CCCCCA	3	0.08%
GGCACG	3	0.08%
CAAAGC	3	0.08%
CTCCGG	3	0.08%
CACCCT	3	0.08%
CACCCA	3	0.08%
CACTCA	3	0.08%
TTGGCT	3	0.08%
TCGGGG	3	0.08%
CGCAGC	3	0.08%
TACCCC	3	0.08%
CGCAGA	3	0.08%
TGGCTC	3	0.08%
CTACGC	3	0.08%
CCACAG	3	0.08%
TTCCTG	3	0.08%
TACTAG	3	0.08%
GCGAGC	3	0.08%
TTCTTT	3	0.08%

CCACCA	3	0.08%
TGGCTG	3	0.08%
CTACCC	3	0.08%
GCTCCT	3	0.08%
TCCTTC	3	0.08%
GCTTCT	3	0.08%
CCATGG	3	0.08%
GCTTCC	3	0.08%
GCTTCA	3	0.08%
GATGAA	3	0.08%
GATGAG	3	0.08%
GCTGGT	3	0.08%
CGACGA	3	0.08%
CCAGAA	3	0.08%
GACACC	3	0.08%
TCTACC	3	0.08%
CCAGCA	3	0.08%
TCTACG	3	0.08%
GGCGCT	3	0.08%
CAGCTC	3	0.08%
CTTTCT	3	0.08%
CTTCCA	3	0.08%
CTTCCC	3	0.08%
GTTTCT	3	0.08%
GGTGTC	3	0.08%

ATCAAA	3	0.08%
ACTAAT	3	0.08%
CTTTGT	3	0.08%
ACCTTC	3	0.08%
AAGCAG	3	0.08%
GGTTTC	3	0.08%
AACCGC	3	0.08%
AAGGCC	3	0.08%
TGAGCC	3	0.08%
ATCAAT	3	0.08%
ACGTCA	3	0.08%
ATCACT	3	0.08%
ACGGCC	3	0.08%
ACGGCT	3	0.08%
CCTCGG	3	0.08%
AACTAA	3	0.08%
GGGTCC	3	0.08%
ATTTAT	3	0.08%
CAACTA	3	0.08%
ATTCAA	3	0.08%
AACCAA	3	0.08%
GCCGAG	3	0.08%
GTGGAG	3	0.08%
CATGAC	3	0.08%
GTCCCC	3	0.08%

CACAGC	3	0.08%
GTCTAC	3	0.08%
ACCTGA	3	0.08%
ACCTGC	3	0.08%
AGCTTC	3	0.08%
ATGAAC	3	0.08%
GGCCTG	3	0.08%
GGCCTC	3	0.08%
GTCCAC	3	0.08%
GTGCCC	3	0.08%
CCTGGA	3	0.08%
GTGTCC	3	0.08%
ACAACA	3	0.08%
CCTTCT	3	0.08%
CATCAG	3	0.08%
ACCGCC	3	0.08%
CCTCCG	3	0.08%
CATGGC	3	0.08%
TATTCC	3	0.08%
GCGGGC	3	0.08%
TCATCG	3	0.08%
TCATCA	3	0.08%
TTCAAG	3	0.08%
TTCAAC	3	0.08%
GACCAG	3	0.08%

AGCCGC	3	0.08%
GCCCGG	3	0.08%
CAACAG	3	0.08%
CAACAC	3	0.08%
GGCCAT	3	0.08%
TGGACA	3	0.08%
GACGGT	3	0.08%
AGGGAC	3	0.08%
GGCCAC	3	0.08%
AGAAGG	3	0.08%
GCAAGC	3	0.08%
CAAGGC	3	0.08%
CCTGCT	3	0.08%
CTATTT	3	0.08%
CAATAA	3	0.08%
CTATTC	3	0.08%
GGCCCG	3	0.08%
GCGGAG	3	0.08%
TGTCCC	3	0.08%
GGCTGG	3	0.08%
TGGAGG	3	0.08%
AGCCAG	3	0.08%
CCACGT	3	0.08%
CGTCAA	3	0.08%
GGCCGA	3	0.08%

TTATTC	3	0.08%
GACGAC	3	0.08%
TCCACA	3	0.08%
TCCACC	3	0.08%
AGCTAC	3	0.08%
GGGGCT	3	0.08%
GCCTCG	3	0.08%
GCCTCC	3	0.08%
CACAAC	3	0.08%
GGCCGG	3	0.08%
GCCCAG	3	0.08%
GCCGGA	3	0.08%
AAGATG	3	0.08%
CCCAAC	3	0.08%
GCCTAA	3	0.08%
CAACGT	3	0.08%
GGGGGC	3	0.08%
CAACGG	3	0.08%
GGCGTC	3	0.08%
AATCTA	3	0.08%
AGCGCC	3	0.08%
CCCGAG	2	0.05%
ACTCTC	2	0.05%
TCACTA	2	0.05%
CTCTCC	2	0.05%

CGCGTC	2	0.05%
TTTCGC	2	0.05%
GCTACC	2	0.05%
GCTACT	2	0.05%
CCCGCG	2	0.05%
GAGTGC	2	0.05%
CCCGCC	2	0.05%
CTCTAC	2	0.05%
GCTAAA	2	0.05%
GAGCCA	2	0.05%
GAGCCG	2	0.05%
TCTGGC	2	0.05%
TAACCA	2	0.05%
GAGCAC	2	0.05%
TTTGTC	2	0.05%
GAGCAT	2	0.05%
CGGTCG	2	0.05%
GCGCGG	2	0.05%
GCGCGA	2	0.05%
AATCCT	2	0.05%
TAATCA	2	0.05%
TCTTCA	2	0.05%
AAGAGG	2	0.05%
GCCCTT	2	0.05%
TCTCCA	2	0.05%



TCTCCC	2	0.05%
AATTGC	2	0.05%
GCCCTC	2	0.05%
TCTCCT	2	0.05%
CTCGAG	2	0.05%
GCCTTC	2	0.05%
CGAACC	2	0.05%
TTTTAC	2	0.05%
TCGAGC	2	0.05%
TGTTTC	2	0.05%
TGTTTT	2	0.05%
GCGCCA	2	0.05%
TCCATG	2	0.05%
GGAGGT	2	0.05%
GAGGCT	2	0.05%
CGCCGT	2	0.05%
CGGGCT	2	0.05%
GGAGGG	2	0.05%
GGAGGC	2	0.05%
GCGCAG	2	0.05%
GCGTCC	2	0.05%
TTTCCG	2	0.05%
TTTCCC	2	0.05%
GAGGAA	2	0.05%
GAGGAC	2	0.05%

GCCGTT	2	0.05%
GGACAA	2	0.05%
CGGGAT	2	0.05%
GGACAG	2	0.05%
GGGTTC	2	0.05%
GTGGTA	2	0.05%
GGGTTT	2	0.05%
ATGGAC	2	0.05%
GGGCTT	2	0.05%
AGTCTG	2	0.05%
CGCGAC	2	0.05%
GGGCTC	2	0.05%
TCTGTT	2	0.05%
ACGACG	2	0.05%
ACGACT	2	0.05%
AACAGA	2	0.05%
CGAAAA	2	0.05%
GTGCGG	2	0.05%
ACTTAC	2	0.05%
ATCTAT	2	0.05%
CATCGC	2	0.05%
CATCGA	2	0.05%
ACTCAT	2	0.05%
ATAACC	2	0.05%
GTGTTT	2	0.05%

GTGCTG	2	0.05%
TCTTTG	2	0.05%
CTTAGC	2	0.05%
ATCGTC	2	0.05%
TCTCTA	2	0.05%
ACTGGC	2	0.05%
ACGAAA	2	0.05%
ACCAGG	2	0.05%
CCTTTC	2	0.05%
ATCTGC	2	0.05%
ATCTGG	2	0.05%
CCTCTA	2	0.05%
TGCGCA	2	0.05%
ACCAGC	2	0.05%
AACAAA	2	0.05%
TTTCTT	2	0.05%
AAAGTG	2	0.05%
GCACAA	2	0.05%
CCGGTT	2	0.05%
CAGCCT	2	0.05%
GTCACC	2	0.05%
CTTTTC	2	0.05%
GGACTG	2	0.05%
GCATCG	2	0.05%
GGTGCC	2	0.05%

CCTACT	2	0.05%
GCATCC	2	0.05%
CAGCCG	2	0.05%
GAAGAT	2	0.05%
ATCTTT	2	0.05%
ATCCTT	2	0.05%
CTTCTG	2	0.05%
GGTGCT	2	0.05%
CAGCAT	2	0.05%
AGCATG	2	0.05%
AGCATC	2	0.05%
CAGCAC	2	0.05%
TAGCTT	2	0.05%
GCAGGG	2	0.05%
GTCAAG	2	0.05%
CTGCTA	2	0.05%
TGCTCC	2	0.05%
CACGTC	2	0.05%
CAGGGT	2	0.05%
AAACAA	2	0.05%
AGTTCC	2	0.05%
AAACAG	2	0.05%
CTGTTG	2	0.05%
CCTAGA	2	0.05%
ACAGTA	2	0.05%

AAACCC	2	0.05%
GGTCAA	2	0.05%
GCAGCA	2	0.05%
GTACGG	2	0.05%
TGAACG	2	0.05%
TGCCAG	2	0.05%
GTACTG	2	0.05%
ACATGA	2	0.05%
TGCTGT	2	0.05%
ATTACC	2	0.05%
GGTTCG	2	0.05%
CCAATA	2	0.05%
TGCCGC	2	0.05%
AGGAAG	2	0.05%
TGGGCG	2	0.05%
GGTCCT	2	0.05%
ATTAGC	2	0.05%
AAATAC	2	0.05%
TATAAA	2	0.05%
CTGGTA	2	0.05%
CCCCTT	2	0.05%
TAGGTC	2	0.05%
ACATCA	2	0.05%
ACATCC	2	0.05%
AGTGCC	2	0.05%

CAGCGA	2	0.05%
GTCAGC	2	0.05%
TGGGGC	2	0.05%
CACTTT	2	0.05%
CAAATC	2	0.05%
ACACCT	2	0.05%
AGCAGC	2	0.05%
ACACCA	2	0.05%
CCGGTC	2	0.05%
TTCCGC	2	0.05%
GCCAGA	2	0.05%
TACCTC	2	0.05%
GCTGAG	2	0.05%
CTGCGG	2	0.05%
CACTAA	2	0.05%
CACGGC	2	0.05%
TGGCCG	2	0.05%
TGGTAG	2	0.05%
TGGTAC	2	0.05%
CCGGGC	2	0.05%
GAACGA	2	0.05%
AGCACT	2	0.05%
TGGCAG	2	0.05%
TCCGGG	2	0.05%
TTCCCG	2	0.05%

AGCACC	2	0.05%
TTCCCA	2	0.05%
CTGCCT	2	0.05%
GGACGA	2	0.05%
ACAGAA	2	0.05%
CTGCCA	2	0.05%
GCAGTT	2	0.05%
TTCCAT	2	0.05%
TGGCGC	2	0.05%
CCGCAG	2	0.05%
TCCTCA	2	0.05%
CCGCAT	2	0.05%
AGCAAC	2	0.05%
CTGTCC	2	0.05%
ACAGCG	2	0.05%
TAGCAA	2	0.05%
GAACCA	2	0.05%
GAACCG	2	0.05%
GAACCT	2	0.05%
TTCGGG	2	0.05%
TGGTGC	2	0.05%
TCCCCC	2	0.05%
CGCTTC	2	0.05%
CTGGGG	2	0.05%
TCCCCT	2	0.05%

CAGGCG	2	0.05%
CCAACA	2	0.05%
TCAAGA	2	0.05%
CGCCTA	2	0.05%
TTGTCA	2	0.05%
CACCGC	2	0.05%
TCCTGA	2	0.05%
CACCGA	2	0.05%
CCCTCC	2	0.05%
CCCTCA	2	0.05%
AGACGC	2	0.05%
TTGCAG	2	0.05%
TTGCAC	2	0.05%
TGCCTC	2	0.05%
AGATGG	2	0.05%
CGGTGC	2	0.05%
GAAGGT	2	0.05%
CGGTGG	2	0.05%
CGGTGT	2	0.05%
CCAAGG	2	0.05%
CCGGAG	2	0.05%
ACTATT	2	0.05%
CCCTGG	2	0.05%
CTCTGG	2	0.05%
GCACCA	2	0.05%



TGGGTC	2	0.05%
CGCCTT	2	0.05%
TACTGA	2	0.05%
CACCAC	2	0.05%
CTGGAC	2	0.05%
CCCCGA	2	0.05%
TGGGTT	2	0.05%
CCCCGC	2	0.05%
GCATCT	2	0.05%
TCGGGA	2	0.05%
GCACCT	2	0.05%
CACCAA	2	0.05%
TTCCGG	2	0.05%
CCGGCT	2	0.05%
GCACCG	2	0.05%
CCAGGC	2	0.05%
CCAGGG	2	0.05%
TCGTCA	2	0.05%
AAGCTG	2	0.05%
AAGCTT	2	0.05%
CCAGGT	2	0.05%
GCTCGG	2	0.05%
TTTACC	2	0.05%
GAGAAA	2	0.05%
GATGGC	2	0.05%

CCACAA	2	0.05%
TAAAAT	2	0.05%
TACTAC	2	0.05%
CCATCA	2	0.05%
GTTTTT	2	0.05%
CCATCC	2	0.05%
TCGCCG	2	0.05%
TTCTTG	2	0.05%
TTCTTC	2	0.05%
TACGGC	2	0.05%
CCACCC	2	0.05%
TTGTGG	2	0.05%
AGAGCC	2	0.05%
CTACCT	2	0.05%
TGGCTT	2	0.05%
TCGCGC	2	0.05%
GAACTG	2	0.05%
TTGCCT	2	0.05%
TCCCTA	2	0.05%
CGTGCC	2	0.05%
TCCCTC	2	0.05%
TACGCC	2	0.05%
GCGACG	2	0.05%
TACGCG	2	0.05%
CTTGAG	2	0.05%

CTACAC	2	0.05%
TCCCTT	2	0.05%
GACAGT	2	0.05%
AAGGTG	2	0.05%
AAGGTC	2	0.05%
CTATAA	2	0.05%
CTTGCA	2	0.05%
CAAGCG	2	0.05%
GCGAAC	2	0.05%
CAAGCC	2	0.05%
AATAAT	2	0.05%
CAAGCA	2	0.05%
AATACT	2	0.05%
AAAATT	2	0.05%
CAAGAG	2	0.05%
CAAGAT	2	0.05%
AAGCGG	2	0.05%
GACACG	2	0.05%
AAGCGC	2	0.05%
CCAGCG	2	0.05%
ACTACT	2	0.05%
CCAGCC	2	0.05%
AACCTC	2	0.05%
ACTAGT	2	0.05%
TCTACT	2	0.05%

CGCAAC	2	0.05%
CTAGAT	2	0.05%
GGCGCA	2	0.05%
CTTCAG	2	0.05%
CAGCTT	2	0.05%
ACGGTG	2	0.05%
CTTTCG	2	0.05%
ACTAAA	2	0.05%
AAGCCC	2	0.05%
GTCATG	2	0.05%
GTCATC	2	0.05%
CTTCCG	2	0.05%
CAGTTC	2	0.05%
CCTATG	2	0.05%
GTTTCC	2	0.05%
CCTATT	2	0.05%
CGAGGC	2	0.05%
AAGTGC	2	0.05%
AAAATC	2	0.05%
TCGCTT	2	0.05%
ACCCTC	2	0.05%
AAGGGA	2	0.05%
ACCTTA	2	0.05%
GTGACG	2	0.05%
CGATGA	2	0.05%

ACGCGC	2	0.05%
AACGTG	2	0.05%
ACGCGG	2	0.05%
CTTCGG	2	0.05%
GTGAAG	2	0.05%
ACCCTT	2	0.05%
TTGCTG	2	0.05%
CGACCT	2	0.05%
ACGCCG	2	0.05%
TGAGCT	2	0.05%
GGTCTC	2	0.05%
ATCAGC	2	0.05%
ACGCCA	2	0.05%
ACGTGC	2	0.05%
ATATTA	2	0.05%
TGAGCG	2	0.05%
TGAGCA	2	0.05%
GATCGT	2	0.05%
AACCCA	2	0.05%
ACGGGC	2	0.05%
ACGCAG	2	0.05%
TCTATA	2	0.05%
TATTTA	2	0.05%
TATTTC	2	0.05%
GTTCGG	2	0.05%

AACTGC	2	0.05%
TGAGGC	2	0.05%
AACTGG	2	0.05%
CTGATC	2	0.05%
GCCGCA	2	0.05%
GGGCAC	2	0.05%
GTGGCC	2	0.05%
TTAGCT	2	0.05%
AAAAAT	2	0.05%
ATGACC	2	0.05%
ACCCGA	2	0.05%
CCTCGC	2	0.05%
TGATCA	2	0.05%
ATTTCA	2	0.05%
TGATCG	2	0.05%
CAACTG	2	0.05%
AAAACA	2	0.05%
AAAACC	2	0.05%
CAATTG	2	0.05%
CAATTC	2	0.05%
GTCTGG	2	0.05%
CATGAT	2	0.05%
GCCGAC	2	0.05%
ACCCCA	2	0.05%
ATACTA	2	0.05%

ACCCCT	2	0.05%
ATACTT	2	0.05%
GTCCCT	2	0.05%
TGATGA	2	0.05%
ACGGAG	2	0.05%
AGCTTG	2	0.05%
AGGTGT	2	0.05%
AGAATT	2	0.05%
GTCTAT	2	0.05%
ATGAAG	2	0.05%
AGCCTC	2	0.05%
GTCTCC	2	0.05%
TGACGC	2	0.05%
GGCCTA	2	0.05%
GTCCAG	2	0.05%
ACCGGC	2	0.05%
ACCGGT	2	0.05%
GTGCCG	2	0.05%
ACAAAT	2	0.05%
TTACCT	2	0.05%
TGCACA	2	0.05%
TCAGCC	2	0.05%
CATCCG	2	0.05%
ACCCAC	2	0.05%
CATCCT	2	0.05%

ACCTCA	2	0.05%
CCTTAG	2	0.05%
CCGACC	2	0.05%
CATCCC	2	0.05%
CATCCA	2	0.05%
AGGTCT	2	0.05%
CCTCAC	2	0.05%
AACGAA	2	0.05%
TTATCA	2	0.05%
CCTTCA	2	0.05%
GTGCAG	2	0.05%
ACCGCA	2	0.05%
CTCATA	2	0.05%
CATTCA	2	0.05%
CGTCTA	2	0.05%
TACATG	2	0.05%
CCTTCG	2	0.05%
CTGAGG	2	0.05%
CAGACG	2	0.05%
TGCAGT	2	0.05%
AGGCAC	2	0.05%
CATGGG	2	0.05%
TCATAC	2	0.05%
CATGGA	2	0.05%
CCCATC	2	0.05%



ATTCGC	2	0.05%
CCTTGA	2	0.05%
TATTCG	2	0.05%
CCCATG	2	0.05%
TATCAC	2	0.05%
TCATCC	2	0.05%
TATTAT	2	0.05%
TATTAC	2	0.05%
TATTAG	2	0.05%
TCACCA	2	0.05%
TCACCC	2	0.05%
GGCGGA	2	0.05%
AGGGCT	2	0.05%
GGCGGT	2	0.05%
GCCCGC	2	0.05%
GCGGCA	2	0.05%
TCATGG	2	0.05%
CCTGAT	2	0.05%
CAACAT	2	0.05%
TGTGGT	2	0.05%
CCATTT	2	0.05%
GAAAAC	2	0.05%
CCTGCA	2	0.05%
GGCTCC	2	0.05%
AGCTGC	2	0.05%

GAAAAT	2	0.05%
GCGGCT	2	0.05%
GGCCAG	2	0.05%
GGAACC	2	0.05%
CTACTA	2	0.05%
GCGGAA	2	0.05%
GGCCCA	2	0.05%
AGCTCT	2	0.05%
GGCTCT	2	0.05%
CAAGGT	2	0.05%
GGCCCT	2	0.05%
GCCTGT	2	0.05%
TGGAAA	2	0.05%
CTATTA	2	0.05%
AGCCCG	2	0.05%
CAAGGA	2	0.05%
TGGAAG	2	0.05%
GCCTGC	2	0.05%
GGAAGC	2	0.05%
TGTCCG	2	0.05%
TGTCCA	2	0.05%
GACGCC	2	0.05%
AGCTCA	2	0.05%
CGTCAG	2	0.05%
GACGCT	2	0.05%

TCCAAC	2	0.05%
TGGAGT	2	0.05%
AGAACA	2	0.05%
GCCCAC	2	0.05%
CACAAG	2	0.05%
TTATTA	2	0.05%
TGTCAT	2	0.05%
AGCTAA	2	0.05%
CGGAAG	2	0.05%
GCCTCT	2	0.05%
GCCTCA	2	0.05%
AGAAAG	2	0.05%
CGGAAT	2	0.05%
GCAACG	2	0.05%
AGAAAA	2	0.05%
GCCGGC	2	0.05%
TTCAGC	2	0.05%
ATTCTA	2	0.05%
CTCACC	2	0.05%
CGGAGG	2	0.05%
CCCAAG	2	0.05%
CTCAAT	2	0.05%
CTCAAA	2	0.05%
CCCACC	2	0.05%
CCCACA	2	0.05%

TGTTGC	2	0.05%
TCCAGG	2	0.05%
TCCAGC	2	0.05%
CCCACG	2	0.05%
GGGGGT	2	0.05%
TCCAGT	2	0.05%
GGGGGG	2	0.05%
CGTCCC	2	0.05%
TTGAGG	1	0.03%
CTCCCC	1	0.03%
CTCCCA	1	0.03%
TTGAGA	1	0.03%
GAGCGA	1	0.03%
GAGCGG	1	0.03%
TTTTGC	1	0.03%
TGTGTT	1	0.03%
CTCTCT	1	0.03%
TCATTT	1	0.03%
CCCGAT	1	0.03%
CGCGTA	1	0.03%
CGGGGT	1	0.03%
CTCTCA	1	0.03%
CTCCAT	1	0.03%
CTCCAA	1	0.03%
CTAACA	1	0.03%

GCTACG	1	0.03%
TCACTC	1	0.03%
CTCCAC	1	0.03%
TCTGCG	1	0.03%
TCACTT	1	0.03%
GAGTGA	1	0.03%
CTCTAT	1	0.03%
CTCTAG	1	0.03%
GAGTGT	1	0.03%
CTCTAA	1	0.03%
GACTTG	1	0.03%
TTGACC	1	0.03%
CTCGGC	1	0.03%
CTCGGA	1	0.03%
CTAAAC	1	0.03%
ACGATC	1	0.03%
GAGCCC	1	0.03%
GCTAAG	1	0.03%
TCGAAA	1	0.03%
TGGATA	1	0.03%
TCTGGT	1	0.03%
GTGCTT	1	0.03%
CCCGGA	1	0.03%
TCTGGG	1	0.03%
TAACCG	1	0.03%

TTTGTG	1	0.03%
TCGACA	1	0.03%
CGGCAC	1	0.03%
CCAAAA	1	0.03%
TCTCAA	1	0.03%
TAACAT	1	0.03%
CCAAAG	1	0.03%
GGAGAA	1	0.03%
TAATCG	1	0.03%
CGGTCC	1	0.03%
TCTTAT	1	0.03%
CCCTAA	1	0.03%
ACTTTC	1	0.03%
GAGTCT	1	0.03%
CCAAAT	1	0.03%
ACTTTG	1	0.03%
CCCTAC	1	0.03%
CTCGCG	1	0.03%
TCTTCG	1	0.03%
TCTTCC	1	0.03%
TCTCAT	1	0.03%
GCCTTT	1	0.03%
AAGAGC	1	0.03%
TCTCAC	1	0.03%
GAGGGG	1	0.03%

AACATA	1	0.03%
AACATC	1	0.03%
ATCGCG	1	0.03%
GCCCTG	1	0.03%
GACGTC	1	0.03%
GACGTG	1	0.03%
CGCCCT	1	0.03%
AACATG	1	0.03%
TAATAG	1	0.03%
GCGCGT	1	0.03%
AACATT	1	0.03%
AATTGT	1	0.03%
CGCTAT	1	0.03%
TTTGGC	1	0.03%
ATCGCT	1	0.03%
TAAGGC	1	0.03%
TCGAGG	1	0.03%
GCCTTA	1	0.03%
CGCCAA	1	0.03%
CGCCAG	1	0.03%
TGTTTG	1	0.03%
GCAATG	1	0.03%
CGAACT	1	0.03%
CGCCAT	1	0.03%
AATCGG	1	0.03%

TCTTGG	1	0.03%
AATTAC	1	0.03%
GACCGC	1	0.03%
TCTTGC	1	0.03%
GAGGCA	1	0.03%
TTTTCA	1	0.03%
AAGACC	1	0.03%
CGCCGA	1	0.03%
AAGACA	1	0.03%
CGGGCC	1	0.03%
GGATAC	1	0.03%
TTTTCC	1	0.03%
CGGGCA	1	0.03%
GACCGT	1	0.03%
AAGACT	1	0.03%
GGATAG	1	0.03%
TCTCGC	1	0.03%
AATCAA	1	0.03%
AATCAC	1	0.03%
GCGCAC	1	0.03%
GCGTCA	1	0.03%
GCGTCG	1	0.03%
AATCAT	1	0.03%
GAAAGA	1	0.03%
GCGTCT	1	0.03%



TAAGCA	1	0.03%
AATTCT	1	0.03%
AATCCA	1	0.03%
GGACAT	1	0.03%
ATCGAA	1	0.03%
GAAAGT	1	0.03%
ATCGAC	1	0.03%
TTCATA	1	0.03%
CGAAGT	1	0.03%
ACCATG	1	0.03%
ATCTCC	1	0.03%
ACTTGC	1	0.03%
ATCCAT	1	0.03%
ATCTCG	1	0.03%
ATCCCC	1	0.03%
CGCGCA	1	0.03%
ACTCCT	1	0.03%
CGCGCG	1	0.03%
GCTATT	1	0.03%
GGTAGG	1	0.03%
GTGGTC	1	0.03%
CATGTG	1	0.03%
GGTAGC	1	0.03%
CATGTC	1	0.03%
ACGAGC	1	0.03%

TCTGTA	1	0.03%
CATGTA	1	0.03%
GTCCTG	1	0.03%
GCTATA	1	0.03%
ATCCCT	1	0.03%
AAGAAA	1	0.03%
AATGGA	1	0.03%
AGTCTC	1	0.03%
TTGATT	1	0.03%
AAGAAT	1	0.03%
ACTTGG	1	0.03%
CGCGAA	1	0.03%
CGCGAG	1	0.03%
GGGCTA	1	0.03%
AATGGC	1	0.03%
GTAATT	1	0.03%
GAGTTT	1	0.03%
ACTCAG	1	0.03%
GTCTTT	1	0.03%
CTTAAT	1	0.03%
ACTCAA	1	0.03%
CGCGGA	1	0.03%
ATCGGA	1	0.03%
CGAAAC	1	0.03%
GAGCTA	1	0.03%

GTCCTC	1	0.03%
AATGAA	1	0.03%
ACGACA	1	0.03%
ATAAAC	1	0.03%
AATGAC	1	0.03%
AGTTTG	1	0.03%
ATAAAA	1	0.03%
CTTACA	1	0.03%
ACTTAT	1	0.03%
ACGACC	1	0.03%
CTTACG	1	0.03%
AAATTG	1	0.03%
CATCGT	1	0.03%
AAATTT	1	0.03%
CTTACT	1	0.03%
ATCGGT	1	0.03%
CATCGG	1	0.03%
ATCCAC	1	0.03%
ACTTCA	1	0.03%
ATCCAG	1	0.03%
GATAGA	1	0.03%
AGGATG	1	0.03%
AAACTC	1	0.03%
AAACTA	1	0.03%
TCGATG	1	0.03%

GTGTGG	1	0.03%
TCGATT	1	0.03%
AACACA	1	0.03%
ACACTT	1	0.03%
TCTTTT	1	0.03%
AACACG	1	0.03%
TATATC	1	0.03%
CATCTT	1	0.03%
ATGGGA	1	0.03%
CATCTC	1	0.03%
CATCTA	1	0.03%
ATGTAC	1	0.03%
ATGGGT	1	0.03%
ATCGTG	1	0.03%
CCGATG	1	0.03%
CATTTG	1	0.03%
CCGATA	1	0.03%
GATACG	1	0.03%
AGTGTC	1	0.03%
CATTTT	1	0.03%
GATACA	1	0.03%
CATTTA	1	0.03%
ACGAAC	1	0.03%
ATCTGT	1	0.03%
ATGGCA	1	0.03%

GATAAC	1	0.03%
ATCCGG	1	0.03%
ATGGCC	1	0.03%
TTTTTA	1	0.03%
CCTCTC	1	0.03%
TTTTTG	1	0.03%
TGCGCG	1	0.03%
ACCAGA	1	0.03%
TGCGCC	1	0.03%
ATCCGA	1	0.03%
GCTAGA	1	0.03%
CCTCTT	1	0.03%
TTTCTA	1	0.03%
GTTAAT	1	0.03%
ACTGCA	1	0.03%
ACTGAT	1	0.03%
ACATTA	1	0.03%
ACATTC	1	0.03%
TTTCTG	1	0.03%
ACACTC	1	0.03%
CTGATT	1	0.03%
TGCGAC	1	0.03%
ACCACC	1	0.03%
ATCTTC	1	0.03%
ATAATA	1	0.03%

GTAGTC	1	0.03%
AGTCGC	1	0.03%
GGATTA	1	0.03%
AGCATT	1	0.03%
ATGTGT	1	0.03%
GAAGAA	1	0.03%
GGACTA	1	0.03%
AGTTGG	1	0.03%
GGTGCG	1	0.03%
GAAGAG	1	0.03%
AAATGA	1	0.03%
CAGCCC	1	0.03%
GGTGCA	1	0.03%
AAATGG	1	0.03%
TGAAAC	1	0.03%
GCATCA	1	0.03%
TGCGGA	1	0.03%
CCTACC	1	0.03%
GCACAC	1	0.03%
ACCACA	1	0.03%
AGGAGA	1	0.03%
CAGTCG	1	0.03%
CAGTCC	1	0.03%
GTCAAA	1	0.03%
AGGAGT	1	0.03%

CTGCTT	1	0.03%
ATGTCT	1	0.03%
CAGTAG	1	0.03%
TGCTCA	1	0.03%
TAGCTC	1	0.03%
TGCTCG	1	0.03%
ATGTCC	1	0.03%
AGTCCA	1	0.03%
CACGTG	1	0.03%
CAGTAA	1	0.03%
CAGTAC	1	0.03%
CTGTTT	1	0.03%
AAACAC	1	0.03%
ATGCCG	1	0.03%
AGTCAT	1	0.03%
CAGGGA	1	0.03%
CAGGGC	1	0.03%
GGTGGA	1	0.03%
CTGTTC	1	0.03%
AAATCA	1	0.03%
GGGAAG	1	0.03%
ATGCCA	1	0.03%
TGCCCCG	1	0.03%
TGCTAA	1	0.03%
TGCTAC	1	0.03%

ACCAAC	1	0.03%
GGGACC	1	0.03%
ACCAAG	1	0.03%
GGGACG	1	0.03%
CCGTTG	1	0.03%
GGTGGT	1	0.03%
GGTTAA	1	0.03%
AAACCA	1	0.03%
CCGCTA	1	0.03%
GGTCAG	1	0.03%
AGGACT	1	0.03%
GGTCAC	1	0.03%
ATGTGC	1	0.03%
TGAACC	1	0.03%
TGCCAA	1	0.03%
TGCCAC	1	0.03%
GGGACT	1	0.03%
AAACCT	1	0.03%
AGGACC	1	0.03%
TTCGGA	1	0.03%
CATAAT	1	0.03%
ATTACT	1	0.03%
ACATGC	1	0.03%
GTTGCC	1	0.03%
AGTGGA	1	0.03%



ACATGG	1	0.03%
AGTGGC	1	0.03%
CATAAC	1	0.03%
GTACTT	1	0.03%
GGTCAT	1	0.03%
CAGGTC	1	0.03%
GGTTCC	1	0.03%
CCGTGC	1	0.03%
TGCCGA	1	0.03%
TGCCGG	1	0.03%
GCAGAG	1	0.03%
TATACT	1	0.03%
TGGGCA	1	0.03%
AGGAAC	1	0.03%
ACACGA	1	0.03%
TGCCCT	1	0.03%
GGGAGG	1	0.03%
CGGCTT	1	0.03%
CCCTTG	1	0.03%
AGACTT	1	0.03%
CCGCGA	1	0.03%
ACATGT	1	0.03%
AAAGGG	1	0.03%
CCGCGT	1	0.03%
AGTGCT	1	0.03%

GTTGCA	1	0.03%
GGGAGT	1	0.03%
TGGGAA	1	0.03%
TTCGCG	1	0.03%
TATAAT	1	0.03%
TTCGCC	1	0.03%
ACACGG	1	0.03%
TGGGAG	1	0.03%
CCCCTG	1	0.03%
TTCGCA	1	0.03%
CACCTA	1	0.03%
AGATTC	1	0.03%
AGTGCA	1	0.03%
AGTGAG	1	0.03%
AAAGAA	1	0.03%
TTCGAT	1	0.03%
ACACAT	1	0.03%
AAAGAC	1	0.03%
AGTGAT	1	0.03%
ACACAG	1	0.03%
GTCAGT	1	0.03%
TCCGAG	1	0.03%
ACACAA	1	0.03%
CAAATA	1	0.03%
TGGGCT	1	0.03%

TCCGAT	1	0.03%
CAGTGA	1	0.03%
AGTGAA	1	0.03%
ATTAAC	1	0.03%
CAGTGG	1	0.03%
AGTGAC	1	0.03%
ATTAAG	1	0.03%
CAAATG	1	0.03%
AAAGCC	1	0.03%
AAAGCA	1	0.03%
AAAGCG	1	0.03%
AGCAGT	1	0.03%
TCCGCG	1	0.03%
CCGGTG	1	0.03%
TCCGCC	1	0.03%
TCCGCA	1	0.03%
CTCCTA	1	0.03%
GCCAGG	1	0.03%
CAAAAG	1	0.03%
CGTACT	1	0.03%
GTAGCT	1	0.03%
TCCGCT	1	0.03%
CAAAAC	1	0.03%
CACGGG	1	0.03%
GGATCT	1	0.03%

CTGCGA	1	0.03%
TAGCGC	1	0.03%
TGGTCT	1	0.03%
GCTGAC	1	0.03%
CACTAC	1	0.03%
GCATTA	1	0.03%
GCATTC	1	0.03%
TACTTG	1	0.03%
TTCTGT	1	0.03%
CTCTTC	1	0.03%
CTCTTA	1	0.03%
TTCTGG	1	0.03%
TTCTGA	1	0.03%
TTCTGC	1	0.03%
CACGGA	1	0.03%
GGACCA	1	0.03%
GCCAGT	1	0.03%
GGACCC	1	0.03%
TTGGAG	1	0.03%
TTAAGG	1	0.03%
TTGGAC	1	0.03%
TACTTC	1	0.03%
ACAGGC	1	0.03%
CTGTGC	1	0.03%
TAGTGA	1	0.03%

ACATAA	1	0.03%
GAACGC	1	0.03%
CCGGGT	1	0.03%
TGGCAC	1	0.03%
CGTACC	1	0.03%
TCAAAC	1	0.03%
AGCACA	1	0.03%
GGATGA	1	0.03%
TCAAAA	1	0.03%
TCCGGC	1	0.03%
CCCGTC	1	0.03%
GTAGCA	1	0.03%
GCCACA	1	0.03%
CGGTTC	1	0.03%
TGGTGG	1	0.03%
TCAAAT	1	0.03%
CACGCG	1	0.03%
TGGTGT	1	0.03%
TCCTAG	1	0.03%
TCCTAA	1	0.03%
AATATG	1	0.03%
TCCTAC	1	0.03%
CACGCA	1	0.03%
CCCGTT	1	0.03%
CCGCAA	1	0.03%

ATTAGT	1	0.03%
CGGTTT	1	0.03%
TCCTAT	1	0.03%
TAGCCG	1	0.03%
TTCTCT	1	0.03%
CTGTCT	1	0.03%
TTCTCA	1	0.03%
TGGCGA	1	0.03%
TTCTCC	1	0.03%
GAATGG	1	0.03%
ATGTTT	1	0.03%
CCGTCT	1	0.03%
CTGCAT	1	0.03%
TGGCGT	1	0.03%
TTAACC	1	0.03%
TCCCAA	1	0.03%
TTCCAA	1	0.03%
CCGTCG	1	0.03%
TAGTCT	1	0.03%
TCCCAC	1	0.03%
CTGTCG	1	0.03%
TTCCAC	1	0.03%
GCCAAC	1	0.03%
CTGTCA	1	0.03%
ATGCTT	1	0.03%

TAGCAG	1	0.03%
ACAGCA	1	0.03%
TCCTCG	1	0.03%
ACAGCC	1	0.03%
CTGCAC	1	0.03%
TAGTCA	1	0.03%
CTGCAG	1	0.03%
TCCCAT	1	0.03%
TTCTAT	1	0.03%
TAGCAC	1	0.03%
TCCTCT	1	0.03%
CCGCCT	1	0.03%
GCAGTG	1	0.03%
TTCTAA	1	0.03%
GCCAAT	1	0.03%
TAGTAG	1	0.03%
TCCCCG	1	0.03%
CTCGTA	1	0.03%
CGGTTA	1	0.03%
GCCAAG	1	0.03%
CTGTAC	1	0.03%
TTCGGC	1	0.03%
ATTATG	1	0.03%
GTACCT	1	0.03%
ATTATA	1	0.03%

CTGGGA	1	0.03%
AGATGT	1	0.03%
GAACAC	1	0.03%
GAACAG	1	0.03%
CCAACC	1	0.03%
CCCCAC	1	0.03%
CGGCGT	1	0.03%
TTGTCG	1	0.03%
CAGGCA	1	0.03%
GAACAT	1	0.03%
TTGTCC	1	0.03%
TCGGAG	1	0.03%
ATTATT	1	0.03%
CCCTCG	1	0.03%
TCGGAC	1	0.03%
CCCCAT	1	0.03%
TCGGAA	1	0.03%
CAAAGT	1	0.03%
CACTGC	1	0.03%
TGCCTG	1	0.03%
TTGCAA	1	0.03%
TGCCTA	1	0.03%
TGCCTT	1	0.03%
GGAGTT	1	0.03%
GAAGGC	1	0.03%



TCGGCA	1	0.03%
TTCGTC	1	0.03%
AGATGA	1	0.03%
CAGGAA	1	0.03%
TCCCGT	1	0.03%
CAAAGA	1	0.03%
GAATAG	1	0.03%
CTGGCT	1	0.03%
TCGGCG	1	0.03%
GTACCC	1	0.03%
TCCCGG	1	0.03%
TGTACG	1	0.03%
CTGGCA	1	0.03%
CACCCG	1	0.03%
GTAGGT	1	0.03%
TGTACC	1	0.03%
AGATCT	1	0.03%
CCAAGC	1	0.03%
CTCCGC	1	0.03%
TACCGA	1	0.03%
TACCGG	1	0.03%
GCATGT	1	0.03%
CCCTGC	1	0.03%
CCAAGT	1	0.03%
CTCTGT	1	0.03%

CCGGAA	1	0.03%
GCATGG	1	0.03%
CGGGTC	1	0.03%
CACCCC	1	0.03%
AGACCG	1	0.03%
GCATGA	1	0.03%
CCGGAT	1	0.03%
CTGGAT	1	0.03%
AGACCA	1	0.03%
GCACCC	1	0.03%
CTCTGC	1	0.03%
CTAAGT	1	0.03%
CTGGAA	1	0.03%
TACTGC	1	0.03%
CACCAG	1	0.03%
CCGGCA	1	0.03%
CTAAGC	1	0.03%
CAAACCT	1	0.03%
GAAGCC	1	0.03%
TCGGGC	1	0.03%
GAAGCG	1	0.03%
CCGGCG	1	0.03%
GGTGAA	1	0.03%
CAAACA	1	0.03%
TCGTAC	1	0.03%

GAAGCT	1	0.03%
TTCCGT	1	0.03%
AGACAC	1	0.03%
CTCCCT	1	0.03%
TTGGCC	1	0.03%
TACCCG	1	0.03%
GATTAT	1	0.03%
TCCGTT	1	0.03%
TACCCA	1	0.03%
TACTCT	1	0.03%
CGCAGG	1	0.03%
GATCAA	1	0.03%
AGATAC	1	0.03%
GATCAC	1	0.03%
AAGCTC	1	0.03%
CGTGAC	1	0.03%
AGAGGC	1	0.03%
TCGTCC	1	0.03%
AGAGGG	1	0.03%
TCGCAG	1	0.03%
AGAGGA	1	0.03%
TACCAC	1	0.03%
TAAACC	1	0.03%
ATAGAT	1	0.03%
GCTCGA	1	0.03%

TAAACA	1	0.03%
CTACGG	1	0.03%
GCTCGC	1	0.03%
GAGAAC	1	0.03%
TTTACG	1	0.03%
GCTCGT	1	0.03%
TTCCTT	1	0.03%
GATGGA	1	0.03%
CTATGT	1	0.03%
CCACAC	1	0.03%
GCTTGT	1	0.03%
CCATCG	1	0.03%
GATTAA	1	0.03%
ATAGCA	1	0.03%
CCACAT	1	0.03%
TAAAAA	1	0.03%
TACGGG	1	0.03%
CTATGA	1	0.03%
GCGAGG	1	0.03%
GCGAGA	1	0.03%
TTAATT	1	0.03%
GCTTGC	1	0.03%
TACGGA	1	0.03%
TAAAAC	1	0.03%
AGAGCT	1	0.03%

TTAATC	1	0.03%
TCGTGG	1	0.03%
GTGATC	1	0.03%
CATATT	1	0.03%
CTACCG	1	0.03%
TCCTTA	1	0.03%
CGATTA	1	0.03%
CTACCA	1	0.03%
GAATTT	1	0.03%
GCTCCG	1	0.03%
TCCTTG	1	0.03%
AATAGG	1	0.03%
AATAGC	1	0.03%
CCATGA	1	0.03%
TCCTTT	1	0.03%
CTATCT	1	0.03%
CGACTC	1	0.03%
CGACTA	1	0.03%
CGTGCG	1	0.03%
TCGCGT	1	0.03%
CGACTG	1	0.03%
CTACAT	1	0.03%
TTGCCG	1	0.03%
TACGCA	1	0.03%
CTACAG	1	0.03%

ATAGGT	1	0.03%
GCGACC	1	0.03%
ACGCTC	1	0.03%
CTACAA	1	0.03%
TCCCTG	1	0.03%
ACGCTG	1	0.03%
TTGTTT	1	0.03%
TACGAT	1	0.03%
ATACAC	1	0.03%
GCGACT	1	0.03%
GAATTC	1	0.03%
CTATAC	1	0.03%
CTTGCC	1	0.03%
ATACAT	1	0.03%
ATATCA	1	0.03%
AATACA	1	0.03%
CGACGC	1	0.03%
CCAGAG	1	0.03%
CGACGT	1	0.03%
TCTAAT	1	0.03%
GACACA	1	0.03%
TCTAAG	1	0.03%
TCGGTT	1	0.03%
CGCACC	1	0.03%
GACACT	1	0.03%

GGCGAT	1	0.03%
GGCGAC	1	0.03%
GGCGAA	1	0.03%
GGCGAG	1	0.03%
CTTGGA	1	0.03%
CTTGGC	1	0.03%
CTAGCC	1	0.03%
ACTAGC	1	0.03%
CTAGCA	1	0.03%
CTTGGT	1	0.03%
GACAAC	1	0.03%
GACAAG	1	0.03%
GCTGAT	1	0.03%
TTGGTG	1	0.03%
GACAAT	1	0.03%
GCACTG	1	0.03%
GAAGTC	1	0.03%
GCACTA	1	0.03%
GCACTC	1	0.03%
CTTCAA	1	0.03%
CTAGAA	1	0.03%
CTTCAC	1	0.03%
ACGGTC	1	0.03%
CTTTCC	1	0.03%
ACTAAC	1	0.03%

AAGCCA	1	0.03%
AAGCCT	1	0.03%
CGACAC	1	0.03%
TCTAGC	1	0.03%
CCTATC	1	0.03%
GTCATA	1	0.03%
CGACAT	1	0.03%
CGATCC	1	0.03%
GTGAGT	1	0.03%
ACTACA	1	0.03%
GGTGTT	1	0.03%
CAGTTG	1	0.03%
ACTACC	1	0.03%
GGTGTG	1	0.03%
TTTATC	1	0.03%
ATCAAC	1	0.03%
AAGTGG	1	0.03%
GTTCCC	1	0.03%
CGAGGG	1	0.03%
AAAATA	1	0.03%
AAGTGA	1	0.03%
TCGCTC	1	0.03%
AAAATG	1	0.03%
CGATAA	1	0.03%
CTTTGA	1	0.03%



ACAATT	1	0.03%
ACCTTG	1	0.03%
GTTCAA	1	0.03%
GCTCTT	1	0.03%
AACGTC	1	0.03%
CGATGG	1	0.03%
ACGCGA	1	0.03%
TGAGAA	1	0.03%
CATACG	1	0.03%
ATATGG	1	0.03%
CATACT	1	0.03%
CTTCGT	1	0.03%
GCTCTG	1	0.03%
GCTCTA	1	0.03%
GCTCTC	1	0.03%
CGACCC	1	0.03%
CGACCA	1	0.03%
GTTGGC	1	0.03%
GGGATA	1	0.03%
GGGATC	1	0.03%
GATGTC	1	0.03%
GCTTTG	1	0.03%
AAGCAT	1	0.03%
GCTTTT	1	0.03%
AAGTCT	1	0.03%

ATACGG	1	0.03%
ATACGC	1	0.03%
GTTGTC	1	0.03%
ACGCCT	1	0.03%
AGGCTT	1	0.03%
AAGGCA	1	0.03%
AAGGCG	1	0.03%
AACCGG	1	0.03%
CGAGAA	1	0.03%
ATCAGT	1	0.03%
CGAGAG	1	0.03%
GCGATG	1	0.03%
ACGTGA	1	0.03%
GGTCTT	1	0.03%
ATATTC	1	0.03%
ACGTGT	1	0.03%
AGTAGT	1	0.03%
CTTGTT	1	0.03%
ATATTT	1	0.03%
AGGCTG	1	0.03%
TGAGGT	1	0.03%
AACTCT	1	0.03%
AGTAGC	1	0.03%
GAGAGC	1	0.03%
ACCGTG	1	0.03%

AACCCG	1	0.03%
ATCACG	1	0.03%
GACATC	1	0.03%
GACATG	1	0.03%
GATTGG	1	0.03%
TATCTA	1	0.03%
TGACAA	1	0.03%
GCTGTT	1	0.03%
ACGGGA	1	0.03%
GATTCT	1	0.03%
ACGTCG	1	0.03%
GTTTGG	1	0.03%
GTTTGC	1	0.03%
AAGGAG	1	0.03%
AAAAGG	1	0.03%
GTTCGA	1	0.03%
AGTACT	1	0.03%
CGAGCC	1	0.03%
CGAGCG	1	0.03%
TGAGGA	1	0.03%
AAGGAT	1	0.03%
GATCCT	1	0.03%
TAGATT	1	0.03%
CGAGCT	1	0.03%
ACCCGG	1	0.03%

TGACCT	1	0.03%
CTGATG	1	0.03%
TAGATA	1	0.03%
TAGATG	1	0.03%
ATTTAC	1	0.03%
CCTTGG	1	0.03%
ATTGGA	1	0.03%
AACGGA	1	0.03%
ATTGGC	1	0.03%
AGGGTT	1	0.03%
AACGGG	1	0.03%
GTGGCA	1	0.03%
AACGGC	1	0.03%
GTCCGA	1	0.03%
AGTAAT	1	0.03%
GTCCGC	1	0.03%
GGGTCG	1	0.03%
ATGACT	1	0.03%
TTAGCG	1	0.03%
GGGTCA	1	0.03%
CATGCT	1	0.03%
CCTCGA	1	0.03%
GGGCAG	1	0.03%
ATGACA	1	0.03%
TTAGCA	1	0.03%

AACTAT	1	0.03%
ATTCAT	1	0.03%
ATTCAG	1	0.03%
ATTCAC	1	0.03%
CAACTC	1	0.03%
AACCAG	1	0.03%
AACCAC	1	0.03%
AGGGTC	1	0.03%
TGACCA	1	0.03%
TGACCC	1	0.03%
AACTCC	1	0.03%
ATTTCT	1	0.03%
TGACCG	1	0.03%
ATTGCT	1	0.03%
ACCCCC	1	0.03%
TGACGT	1	0.03%
GTCCAT	1	0.03%
TATGTT	1	0.03%
ATTGCA	1	0.03%
GGAATG	1	0.03%
AACGCG	1	0.03%
GGGTGC	1	0.03%
CACAGG	1	0.03%
GGAATA	1	0.03%
ACAAGC	1	0.03%

ACGGAA	1	0.03%
AGCTTT	1	0.03%
ACAAGA	1	0.03%
GGCTTT	1	0.03%
CCTGTT	1	0.03%
CTTATA	1	0.03%
CTTATC	1	0.03%
CCTGTG	1	0.03%
GTCCAA	1	0.03%
AGCCTA	1	0.03%
GTCTCA	1	0.03%
GGGCGT	1	0.03%
AGGCGG	1	0.03%
ACAAGG	1	0.03%
TGCAAT	1	0.03%
CACATC	1	0.03%
GTGTCT	1	0.03%
ACCGGG	1	0.03%
TGAGTC	1	0.03%
CACATT	1	0.03%
ACCTAA	1	0.03%
GGCCTT	1	0.03%
CACATG	1	0.03%
CAGAGC	1	0.03%
GTGCCA	1	0.03%

CAGAGG	1	0.03%
AGGTGC	1	0.03%
CCGAAG	1	0.03%
GTGCCT	1	0.03%
AGGTGA	1	0.03%
TGCACC	1	0.03%
TTACCC	1	0.03%
ACAAAC	1	0.03%
CGGATT	1	0.03%
TTACCG	1	0.03%
TCAGCT	1	0.03%
CCTTAC	1	0.03%
CCTTAA	1	0.03%
ACCCAT	1	0.03%
ACCTCG	1	0.03%
GTGCAC	1	0.03%
TATTGT	1	0.03%
CCTCAG	1	0.03%
CATTCT	1	0.03%
GGGGTC	1	0.03%
CTCATT	1	0.03%
ACAACT	1	0.03%
GGGGTG	1	0.03%
TTACAT	1	0.03%
AGGCCT	1	0.03%

GTGTCA	1	0.03%
TATTGG	1	0.03%
CATCAT	1	0.03%
AGGCCC	1	0.03%
AGGCCG	1	0.03%
GATCTT	1	0.03%
CTCATC	1	0.03%
ACCGCG	1	0.03%
TACATC	1	0.03%
CCTCAT	1	0.03%
ATTCCA	1	0.03%
GGGGTT	1	0.03%
TTATAT	1	0.03%
ATTCCG	1	0.03%
CCTCCC	1	0.03%
GGTACT	1	0.03%
AGGTCG	1	0.03%
GTGGGC	1	0.03%
AGGTCA	1	0.03%
CAGACC	1	0.03%
TGCAGA	1	0.03%
TGCAGC	1	0.03%
AGGTCC	1	0.03%
GGTACG	1	0.03%
TTATAA	1	0.03%



AGGCAG	1	0.03%
CATTAG	1	0.03%
TGCAGG	1	0.03%
ATGAGG	1	0.03%
GATTTA	1	0.03%
TGCACT	1	0.03%
ATTTGC	1	0.03%
CATTAA	1	0.03%
TCATAA	1	0.03%
ATCATC	1	0.03%
ATCATG	1	0.03%
ATTTGA	1	0.03%
CCGAGA	1	0.03%
TCATAT	1	0.03%
CCGAGC	1	0.03%
TATTCT	1	0.03%
CAGAAA	1	0.03%
GATCTG	1	0.03%
TATTCA	1	0.03%
AGGGGC	1	0.03%
CAATGA	1	0.03%
TTCACA	1	0.03%
TTCACC	1	0.03%
TATCAA	1	0.03%
TGACTA	1	0.03%

GCGGGA	1	0.03%
TGTGCA	1	0.03%
TCACAT	1	0.03%
TGTGCC	1	0.03%
CTGACG	1	0.03%
CTGACC	1	0.03%
GACCCA	1	0.03%
GACCCC	1	0.03%
GACCCT	1	0.03%
GCCGTC	1	0.03%
TTAGTC	1	0.03%
TTCAAT	1	0.03%
ATGATC	1	0.03%
GCGTAC	1	0.03%
TCACCG	1	0.03%
CGCATT	1	0.03%
TATGGA	1	0.03%
TTCAAA	1	0.03%
ACCGAA	1	0.03%
CAACCG	1	0.03%
ACCGAC	1	0.03%
TAGAAT	1	0.03%
CAACCC	1	0.03%
ACCGAG	1	0.03%
ATGATG	1	0.03%

CCCAGA	1	0.03%
CTGAAA	1	0.03%
CGTTGT	1	0.03%
GAAACA	1	0.03%
GAAACC	1	0.03%
GACCAC	1	0.03%
TGATTG	1	0.03%
GGCTAC	1	0.03%
TGATTT	1	0.03%
CCTGAC	1	0.03%
CGTCGA	1	0.03%
CCCAGT	1	0.03%
CCTGAA	1	0.03%
GACTCA	1	0.03%
GAAACT	1	0.03%
AGCCGT	1	0.03%
TCATGA	1	0.03%
GGCGGG	1	0.03%
GGAAAA	1	0.03%
GGAAAC	1	0.03%
ATTTTA	1	0.03%
CGTCGC	1	0.03%
AGGGCC	1	0.03%
AGCCGG	1	0.03%
AGGGCA	1	0.03%

CGTCGG	1	0.03%
ATTGTG	1	0.03%
CCATTC	1	0.03%
GGAACA	1	0.03%
TGTGGG	1	0.03%
CGTGTC	1	0.03%
ATTGTC	1	0.03%
TTTGAC	1	0.03%
GAAAAA	1	0.03%
TCACGA	1	0.03%
TTTGAT	1	0.03%
GGCTCA	1	0.03%
CCTGCG	1	0.03%
AGCTGA	1	0.03%
GACTAA	1	0.03%
GACTAC	1	0.03%
TGGACG	1	0.03%
GCCCGT	1	0.03%
GGCCAA	1	0.03%
TGGACT	1	0.03%
TATGCC	1	0.03%
CTACTC	1	0.03%
AGAAGC	1	0.03%
GACTAT	1	0.03%
TTACGC	1	0.03%

TTACGA	1	0.03%
CAAGGG	1	0.03%
GCAAGA	1	0.03%
TTTGCC	1	0.03%
TTTGCA	1	0.03%
CAATAC	1	0.03%
AGCCCT	1	0.03%
CTATTG	1	0.03%
TTTGCT	1	0.03%
TTATGC	1	0.03%
TGGAAC	1	0.03%
AGCCCC	1	0.03%
TAATTA	1	0.03%
TATGAG	1	0.03%
CACACT	1	0.03%
CTCAGT	1	0.03%
CCATGT	1	0.03%
AATCTG	1	0.03%
CACACG	1	0.03%
AGCTAT	1	0.03%
CCACGA	1	0.03%
CTCAGC	1	0.03%
TACAGC	1	0.03%
CGGACA	1	0.03%
AATCTT	1	0.03%

AGAACT	1	0.03%
GACGCG	1	0.03%
TTACTA	1	0.03%
TTACTG	1	0.03%
AGAACG	1	0.03%
GCCCCG	1	0.03%
AGAACC	1	0.03%
GGGGAG	1	0.03%
GTCGGG	1	0.03%
GCAACC	1	0.03%
TCAGTG	1	0.03%
TGTCAC	1	0.03%
TCAGTC	1	0.03%
TCCAAT	1	0.03%
GTCGCT	1	0.03%
TGTCAA	1	0.03%
TCAGTA	1	0.03%
CAAGTG	1	0.03%
TGTTCC	1	0.03%
TCAGTT	1	0.03%
TTATTG	1	0.03%
GACGAA	1	0.03%
GCCTAT	1	0.03%
AGCGGT	1	0.03%
GACGAG	1	0.03%

CGGAAC	1	0.03%
AGCTAG	1	0.03%
GCCCAT	1	0.03%
TTCAGT	1	0.03%
GCAACT	1	0.03%
CGTGGA	1	0.03%
CTCACT	1	0.03%
AGCGGA	1	0.03%
TACACG	1	0.03%
GTCGAT	1	0.03%
ATTCTG	1	0.03%
TACACA	1	0.03%
TACACC	1	0.03%
CGGAGT	1	0.03%
TAAGTG	1	0.03%
GCCTAG	1	0.03%
AAGATC	1	0.03%
GCCTAC	1	0.03%
CTAGTG	1	0.03%
CGCATG	1	0.03%
ATTCTT	1	0.03%
CTAGTA	1	0.03%
CGTTCG	1	0.03%
GGGGGA	1	0.03%
TGCATC	1	0.03%

CGTTCA	1	0.03%
CGTCAT	1	0.03%
CGTTCC	1	0.03%
AATTTG	1	0.03%
AATTTC	1	0.03%
TACAAA	1	0.03%
AGCGCT	1	0.03%
TGTCGA	1	0.03%
TGTCGC	1	0.03%
AATCTC	1	0.03%
GTCGAG	1	0.03%
AGCGCG	1	0.03%
CGTCCA	1	0.03%