Ahmadou Bassirou DIALLO

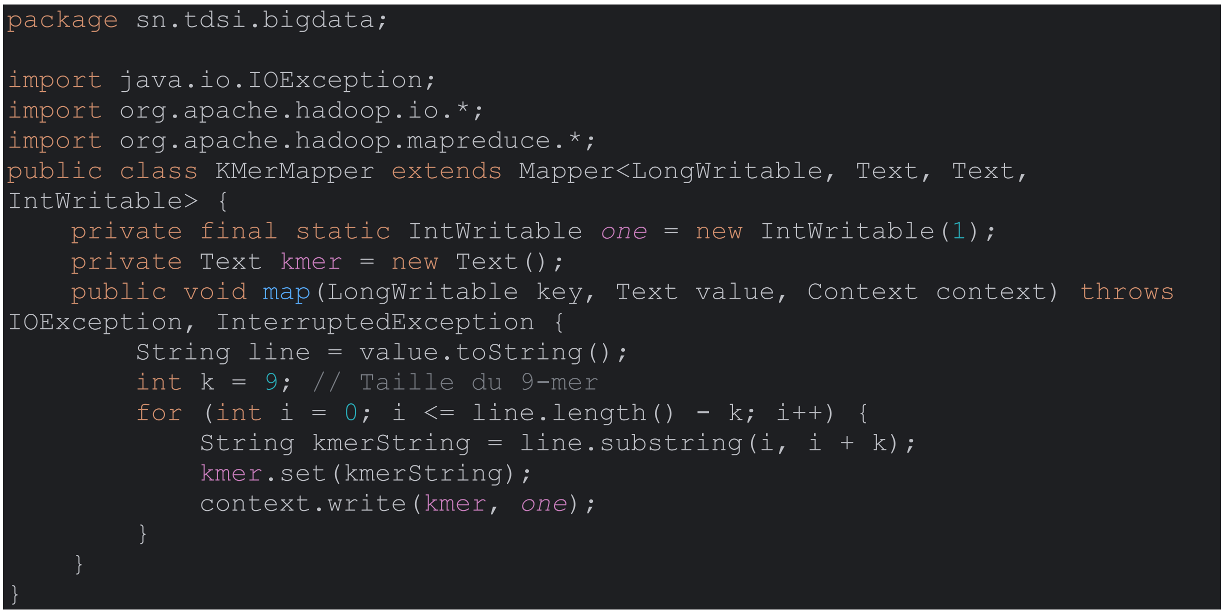
Rapport de devoir sur la création d'un code MapReduce pour compter les séquences de n-mers dans un code génétique.

Le code source :

1. La classe **KMerCountDriver**

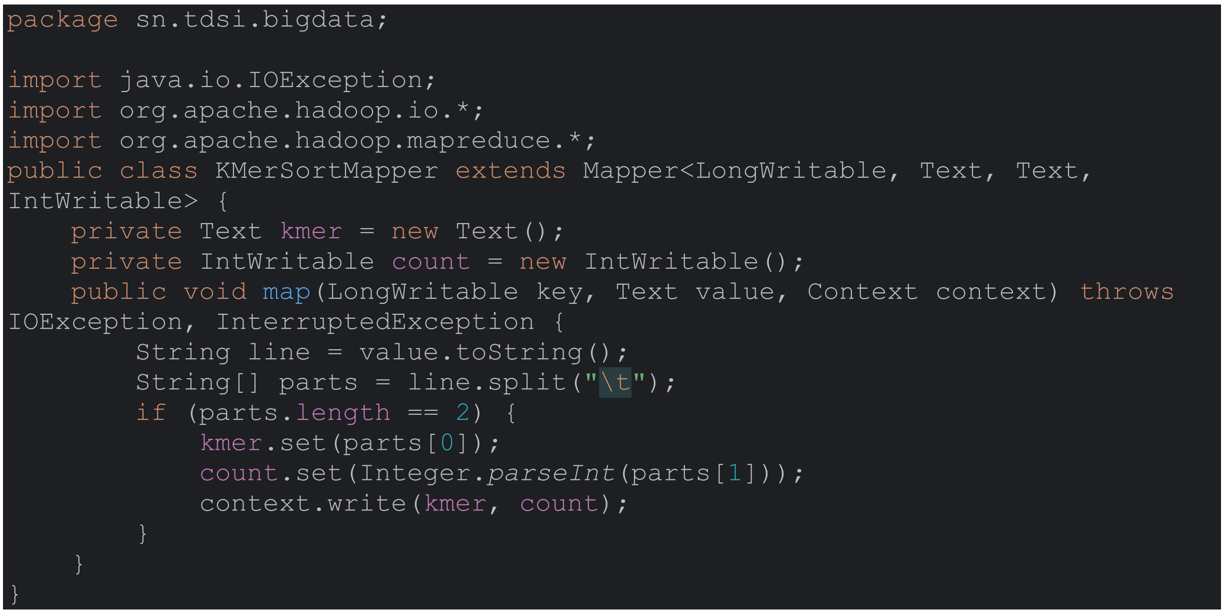
```java
package sn.tdsi.bigdata;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class KMerCountDriver {
    public static void main(String[] args) throws Exception {

        Configuration conf = new Configuration();

        Job job = Job.getInstance(conf, "kmer count");

        job.setJarByClass(KMerCountDriver.class);
        job.setMapperClass(KMerMapper.class);
        job.setCombinerClass(KMerReducer.class);
        job.setReducerClass(KMerReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

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

        job.waitForCompletion(true);
    }
}
```

2. La classe **KMerMapper**



```
package sn.tdsi.bigdata;

import java.io.IOException;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
public class KMerMapper extends Mapper<LongWritable, Text, Text,
IntWritable> {
    private final static IntWritable one = new IntWritable(1);
    private Text kmer = new Text();
    public void map(LongWritable key, Text value, Context context) throws
IOException, InterruptedException {
        String line = value.toString();
        int k = 9; // Taille du 9-mer
        for (int i = 0; i <= line.length() - k; i++) {
            String kmerString = line.substring(i, i + k);
            kmer.set(kmerString);
            context.write(kmer, one);
        }
    }
}
```

3. La classe **KMerSortMapper** pour la trie



```
package sn.tdsi.bigdata;

import java.io.IOException;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
public class KMerSortMapper extends Mapper<LongWritable, Text, Text,
IntWritable> {
    private Text kmer = new Text();
    private IntWritable count = new IntWritable();
    public void map(LongWritable key, Text value, Context context) throws
IOException, InterruptedException {
        String line = value.toString();
        String[] parts = line.split("\t");
        if (parts.length == 2) {
            kmer.set(parts[0]);
            count.set(Integer.parseInt(parts[1]));
            context.write(kmer, count);
        }
    }
}
```

4. La classe **KMerReducer**

```
package sn.tdsi.bigdata;

import java.io.IOException;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;

public class KMerReducer extends Reducer<Text, IntWritable, Text,
IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values, Context
context) throws IOException, InterruptedException {

        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}
```

5. Les 10 "9-mers" les plus fréquents dans E coli avec leurs nombres de répétitions sont :

CCAGCGCCA	**258**
CAGCGCCAG	**252**
GCGCTGGCG	**238**
CGCTGGCGG	**224**
CGCCAGCAG	**221**
CTGGCGCTG	**221**
CGCCAGCGC	**214**
GCCAGCGCC	**213**
TGGCGCTGG	**204**
CCGCCAGCA	**200**

Vous pouvez trouvez aussi mon code source dans le dépôt de mon github en cliquant sur le lien ci-dessous.

https://github.com/BassirouD/Urca-M2/tree/main/big%20data