Agustin Pardo - Genome Regions – Illumina Coding Project

Requirements

Develop on Python 3.9.6

Running the application

Usage options:

Flag	Description	Default
"-in"	Input regions file	"Regions_Large.txt"
"-part1"	Return result file of part1 task	
"-part2"	Return result file of part2 task	

Usage example

Go to the project root directory:

```
cd illumina-coding-project
```

Get the result of part1 task on "Regions_Large.txt" input file. Execute:

```
python main.py -part1
```

Get the result of part2 task on "Regions_Small.txt" input file. Execute:

```
python main.py -in Regions_Small.txt -part2
```

Show usage options. Execute:

```
python main.py -h
```

Notes

- You can run both -part1 and -part2 options together.

- Output file name is base on input file name (Example for "Regions_Small.txt" is "Regions_Small_part2_result.txt").

Test

Go to test directory and execute:

```
python -m unittest -v
```

Results

The results output files of part1 and part2 tasks over "Regions_Small.txt" and "Regions_Large.txt" are in the results directory.

Algorithm description

The program is object-oriented in Python. There are two classes, Region and Segment to store the information of the regions and "non-overlapping segments".

The input file is parsed on the Parser class creating tow datasets (lists) of the regions and the "non-overlapping segments". Each region on the list save the info of start, end and Y-axis coordinate. All the regions start with a 0 value in Y-axis coordinate since the extend of regions overlapping is not know yet. "non-overlapping segments" are calculated using the regions and set a count number of 0.

The Process class is in charge of solving the part 1 and part 2 tasks. To solve the part 1 task the idea is to compare each region to the others regions to see if they have overlapping coordinates. Once the overlaps are find, Y-axis coordinate region value is updated based on the overlapped regions Y-axis values. First a highest Y-axis value plus 1 is selected (If non of them have a Y-axis value (0) the first level is chosen (1)). Then, if the overlapped regions are in certain level the lowest level available is chosen, if not poible it keeps the highest Y-axis level.

To solve the part 2 task the idea is to first find out the "non-overlapping segments" intervals (This is done on the Parser class). For that is necessary first gather the stop and start coordinates of all the regions and sorted from lowest to highest. It is vital to have them sorted in order to be able to define the intervals correctly, otherwise "non-overlapping segments" intervals cannot be constructed without overlaps between them. Once the segments are defined, each segment is compared agains the regions to count how many of them are in each segment.

Contact

agustinmpardo@gmail.com