Software Applications 2022/2023

Project Specification

Andrea Giovanni Nuzzolese

From an high-level point of view the software should address some questions about the human genome.

Input

The input is the human genome annotation file in Generic Feature Format Version 3 (GFF3).
The data can be downloaded here. The file is about 37 MB, which is very small if compared to the information content of a human genome, which is about 3 GB in plain text. That's because the GFF3 file only contains the annotation of the sequences, while the sequence data is usually stored in another file format called FASTA.
can extend the project by adding fasta seq to annotations

Data access

assume all lines starting with # are comments

Taking a look at the head of the GFF file, you will see many metadata/pragmas/directives lines starting with ## or #!. According to the README, ## means the metadata is stable, while #! means it's experimental. Later on you will also see ###, which is another directive with yet more subtle meaning based on the specification. Human readable comments are supposed to be after a single #. For simplicity, we will treat all lines starting with # as comments, and simply ignore them during our analysis.
The data must be read by a *dataset reader* that relies on **Pandas**. The dataset reader is specific to the GFF3 format, however it should be compliant with a general abstract interface that defines the procedure that a dataset reader is provided with. When a dataset reader reads the data it returns a dataset object as output that is a wrapper around a Pandas DataFrame.

Dataset operations

we have a generic dataset, in our peculiar case we want to deal with a GFF3, but the software has to be opened to allow the adding other data types.

A *dataset* is the view over the data. As for the reader the software must distinguish between a generic tabular data and GFF3 data, which is a peculiar case. By means of a dataset object a number of insights over data can be obtained. Each insight is called an *operation*. An operation is executed over a dataset object and the result os an
operation executed on dataset object, returns dataset object
execution is **another dataset object**. There are multiple operation types that can be executed. A dataset object
the dataset is unaware of operations that can be executed, just aware that operation can be executed, not specific type
accesses and executes those operation regardless to the specific operation type. The operations can be executed
not possible in software to execute external operations??
only by means of a dataset object and only if they are marked as *active*. For an operation being active means it is loaded within a registry of active operations that the dataset object can query to check whether or not a certain operation can be executed. **Hint**: Define a decorator for managing the activation of operations.

The operations are:

- getting the some basic information about the dataset. The basic information are the name and data type of each column; name, data type,
- obtaining the list of unique sequence IDs available in the dataset;
- obtaining the list of unique type of operations available in the dataset;
- counting the number of features provided by the same source;
- counting the number of entries for each type of operation;
- deriving a new dataset containing only the information about entire chromosomes. Entries with entire chromosomes comes from source GRCh38;
- calculating the fraction of unassembled sequences from source GRCh38. Hint: unassembled sequences are of type supercontig while the others are of chromosome;
- obtaining a new dataset containing only entries from source `ensembl`, `havana` and `ensembl_havana`;
- counting the number of entries for each type of operation for the dataset containing containing only entries from source `ensembl`, `havana` and `ensembl_havana`;
- returning the gene names from the dataset containing containing only entries from source `ensembl`, `havana` and `ensembl_havana`.

User interface

The application provides the user with web-based interface. The interface consists of the following views:

- Homepage: it is the main page presented by the application when a user accesses it through her preferred browser. The homepage presents an overview over the application by listing, linking and describing the other available views effectively;
- Active operations: a list of active operations is presented. If a user selects an active operation, then such an operation is executed over the dataset;
- Operation: onse a user selects an operation from the previous view and such an operation is executed, then the result of the execution is presented in a different view;
- Project document: it contains the description of the project in terms of software analysis, design and implementation. CRC cards and UML class diagrams must be used for software analysis and desing, respecitvely. Diagrams are not self-explainable. Hence, the **must** be extensively described in the text.

Softare modularisation

It is preferrable that the application is implemented in different python files (aka modules) by aggregating different classes or function coherently.

Project submission

Create a GitHub repository for the project and share this repository with the teacher as final submission.