UCSC Genomic REST Api Wrapper

مازن محمد النبراوي

إياد خالد حمزة

سهيلة سامر عبدالحميد

ياسمين المتولي أحمد

سلمى أحمد الصاوي

1. Motivation

Researchers and analysts may spend long times trying to access and use genomic data from different services, this overhead gets tedious and problematic really fast, especially when researchers have little knowledge regarding http requests and principals of restful apis, so why can't they only focus on their analysis and their work instead of wasting time in repetitive tasks such as getting or querying data from public restful apis?

Eliminating the overhead of the process may help them advance their research and reach a conclusion much faster.

2. Overview

An open-source python package licensed under the MIT license, the package represents a python Api wrapper on the UCSC genomic database, which makes it much easier for researchers to access and query the database with an elegant and human readable Api.

2.1 Significance of the Project

The package is based upon the restful Api of UCSC which offers a handful of useful genomic data that researchers and data analysts may use to extract information.

2.2 Description of the Project

The university of California offers a public rest Api that can be used to access a variety of Genomic data, such as listing public hubs, genome assemblies, list of available data tracks from a specified hub, list of chromosomes contained in an assembly hub and finally list of chromosomes contained in a specific track.

All of these data can be obtained using specific endpoints offered by the organization, with that in mind, comes the idea of this project, to provide simple, easy to use python programmatic interface to access the restful Api, which means using only method calls we can get all data that mentioned above. That comes from the idea that most researchers won't be fully familiar with restful Api concepts and JSON data format, and may therefore face many issues and difficulties. Fortunately, almost every researcher has now a solid knowledge of programming languages such as python, so to make their research a bit easier and faster, this tool is now exists.

2.3 Background of the Project

Previously, users who wanted to download track or sequence data would need to grab the dataset in its entirety from the UCSC downloads server, or use the Table Browser's graphical interface. Unfortunately, scripting against these resources was never practical for dynamic queries like getting track data for >1000 regions or accessing UCSC data via R/Python/Perl or another programming environment.

To better support those use cases, UCSC have created a RESTful API that returns JSON-formatted data for a variety of data retrieval queries.

The API is accessed via the URL “https://api.genome.ucsc.edu” and has two primary functions: listing available datasets and downloading data from said datasets. Notable endpoints include “/list/tracks/” for listing all of the tracks for a given database or hub, and “/getData/sequence/” and “/getData/track/” for obtaining genome sequence and annotation data, respectively.

Now, using this package functions, researchers can only call a function named for example “getData()” to make the HTTP request, returns a json response then serialize the response to python primitives, then with the new python object they can do any analytical processes or functions.

3. Methodology

The series of steps we will be following to achieve the desired ends and finally release the package is stated down below:

3.1 Learning phase

- The team will first study the topics that are necessary for implementing and testing the package, will also dive in RESTFUL Api concepts, techniques and best practices.
- Then study the UCSC Api itself and related concepts.

3.2 Design phase

Here, there won’t be any user interface, as the software may be used as a package inside a larger tool, but instead there will be software models such as class diagrams to illustrate how the package should be structured and implemented.

3.3 Implementation phase

- The team will operate on windows 10 operating system, with python 3 installed and using packages such as “Requests” library.
- The preferred editor is Pycharm as it may increase the speed of the development.
- The team will use GIT version control and GitHub to easily collaborate and track

changes and pull requests.

- The team may use containers such as docker if there were certain configuration that must be shared.
- A ReadMe file will be added as a documentation of how to use the package.

3.4 Testing phase

- Will rely on unit testing to make sure the package works as expected.
- The team will use “unittest” Unit testing framework.

3.5 Evaluation phase

- The issues tab on GitHub will help the team to evaluate and track developers’ feedback for the tool and any potential errors.
- Monitoring the pull request tab for community contributions

4. Features

- **Expressive Api**
  - Method names that are meaningful and easy to remember
  - A solid documentation for the Api that explains every function, the parameter and the return type
- **Easy to use**
  - All it takes is calling a function with the required parameter to get the required data
- **Can be extended**
  - Taking advantage of the packages and modules concept in python, the package may be extended and forked, may also be used in larger project
- **Can be reused.**
  - The package may be reused in all projects, just require it and all the functionality will be available to the developer

5. Project Planning

The document is actively being updated:

https://docs.google.com/spreadsheets/d/1eWUKbfrAE_IKeX9wJfZjOQTQl4HmpCO7Asi5FNAmsBA/edit?usp=sharing