Analysis of structured RDF/XML data using Spark and GraphX

Claudio Reggiani CLAUDIO.REGGIANI@ULB.AC.BE
Mircea Mudura MMUDURA@ULB.AC.BE
Yann-Aël Le Borgne YLEBORGN@ULB.AC.BE
Gianluca Bontempi GBONTE@ULB.AC.BE
Machine Learning Group,
Faculty of Science, Université Libre de Bruxelles, Boulevard du Triomphe, CP 212, 1050 Brussels, Belgium

Keywords: apache spark, graphx, rdf, scalability

Abstract

Bioinformatics is an evolving field combining multiple domains in order to interpret biological data. The integration of complementary functional annotations is an important step in understanding the gene-disease association and the mechanisms underlying complex diseases. Due to the increasing amount of generated data, the challenges are in both storage and analysis. In this work, we present a unified framework for the processing and analysis of UniProt RDF/XML data, using Big Data technology: Spark and GraphX. We describe the data conversion from RDF to GraphX object and show preliminary analysis about the scalability of centrality analysis.

1. Introduction

The existing knowledge in Bioinformatics is split across several structured text based formats, that are difficult to manage in order to have a unified framework for processing and analysis. UniProt provides a freely accessible resource of genes, protein sequences and other functional annotations. The database makes use of semantic web standards to provide high-quality, high-accessibility of its information and to facilitate cross-database information retrieval with other resources.

While our final objective is an integrated analysis of functional annotations to discover novel gene-disease associations, the contributions presented in this work set the ground in that direction: the development of a standalone tool for RDF data import into GraphX object, with a built-in implementation of the closeness centrality algorithm, and scalability analysis. While recent research are addressing the challenge of querying RDF data on a distributed environment (Schätzle et al., 2015), the software is not released.

In this work we present how Apache Spark combined with GraphX provides a distributed solution for graph storage and analysis, using MapReduce paradigm. We investigated the scalability of our solution at two levels: the creation of a distributed graph and performance (in execution time) of our closeness centrality implementation, by varying the number of nodes.

We will provide source code, documentation and executables on a public repository as soon as we complete our scalability analysis.

2. RDF data import

The framework presented in this paper uses the Resource Description Framework (RDF) format, which is a graph representation of statements about resources, in the form of subject-predicate-object.

```
<rdf:Description rdf:about="foo">
  <si:title>bar</si:title>
  <si:author>baz</si:author>
</rdf:Description>
```

Figure 1. RDF/XML generic representation.

Figure 1 provides an example of a generic RDF/XML file. An alternative (named hereafter *n-triple*) representation of the same information, that better fits our scalable infrastructure, is reported in Figure 2. As the RDF/XML generic format is a set of highly

Appearing in *Proceedings of Benelearn 2016*. Copyright 2016 by the author(s)/owner(s).

structured text (blocks of text have a meaning in a specific context), it is currently impossible to use a distributed storage and processing system in order to process all the data. Due to this shortcoming of the format, we preprocess locally the original UniProt RDF/XML data into a collection of n-triples, while preserving all the information available.

```
<rdf:Description rdf:about="foo">
<si:title> bar
<rdf:Description rdf:about="foo">
<si:author> baz
```

Figure 2. RDF/XML n-triple representation.

Having the data transformed into a n-triple format and stored in HDFS, the final step is to transform it into a GraphX graph. We report in Figure 3 the execution time of conversion by increasing the number of nodes (we ran all tests once).

3. Closeness centrality

To showcase the advantages of using Big Data in the context of Bioinformatics, in particular in RDF processing, the tool described in this article implements the Closeness Centrality Algorithm (CCA) (Oktay et al., 2011). CCA is part of a class of algorithms providing relevant information regarding the importance of a node in a network, and the combination of CCA with UniProt data can help researchers consider new avenues in disease treatment.

The closeness centrality score of a node u is given by

$$C(u) = \frac{1}{\sum_{v \in V} d(u,v)}$$

where $d(u,v)$ denotes the distance between nodes u and v.

The distance between all the nodes in the graph is calculated using GraphX's built in shortest path algorithm. In Figure 4 we report the execution time of the algorithm using different number of nodes (we ran all tests once).

4. Results and discussion

In this work we present our solution for a unified framework for processing and analysis of omics data. It is built around two main modules: a preprocessing component that converts RDF/XML data into a GraphX graph object, and a data analysis component that include our implementation of CCA. We studied

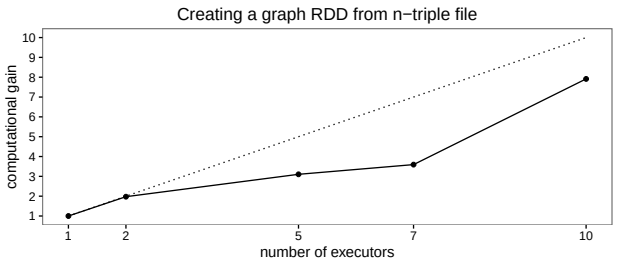


Figure 3. Scalability analysis of creating a GraphX graph object from n-triple file. The dotted line represents the linear scalability.

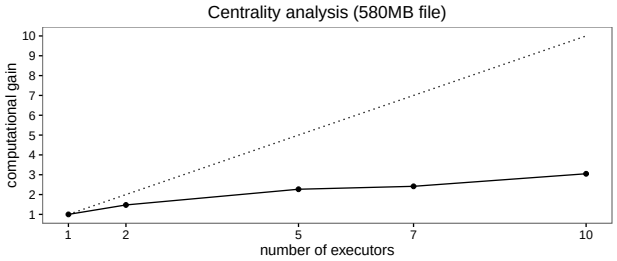


Figure 4. Scalability analysis of the Closeness centrality algorithm. The dotted line represents the linear scalability.

the scalability of the modules by varying the number on executors (i.e. nodes). While we were expecting a linear scalability performance in the creation of a graph RDD object, the internal overhead representation of a graph in GraphX might explain a worse computational gain. On the other hand, it is unclear why the closeness centrality scalability is non-linear. The algorithm is based on the built-in implementation of the shortest path between pairs of nodes and there is no official documentation explaining the details.

Besides a better understanding of the shortest path scalability, our future intents are to integrate multiple functional datasets to UniProt, to study other centrality algorithms and to investigate novel gene-disease associations.

References

Oktay, H., Foster, I., Balkir, A., & Jensen, D. (2011). Distance estimation for very large networks using mapreduce and network structure indices. *Workshop on Information in Networks*, 1–6.

Schätzle, A., Przyjaciel-Zablocki, M., Berberich, T., & Lausen, G. (2015). S2x: Graph-parallel querying of rdf with graphx. *Proc. of 1st International Workshop on Big-Graphs Online Querying.*