Towards real-time pre-processing of Mass Spectrometer data using a scalable cluster

Chris Hillman
School of Computing
University of Dundee

chillman@dundee.ac.uk

ABSTRACT

The human genome project was one of the largest and most well known scientific endeavors of recent times. This project characterised the entire set of genes found in human DNA. Following on from genomics, proteomics is the study of proteins, which are the products of genes found in cells. In a typical proteomics experiment, an instrument called a mass spectrometer is used to identify proteins and measure their quantities. This produces large data files that currently take many hours to process. This research is concerned with reducing the time needed to process data files to near real-time by designing a novel algorithm that can run in a parallel fashion and evaluating its performance characteristics across various cluster computing platforms that have not previously been used for this purpose.

1. INTRODUCTION

Proteomics can be defined as the large-scale study of protein properties, such as expression levels, modifications and interactions with other proteins. By studying proteins and their properties, it is possible to gain a deeper understanding of how proteins should function in healthy cells compared with diseased cells. Full proteins are in most cases too large to be measured intact by mass spectrometers, therefore a process of fragmentation breaks the proteins into constituent parts called peptides. It is these peptides that are analysed and identified by software in a data processing step after the mass spectrometer has processed a cell sample. This identification and quantification allows Life Scientists to research how different environments and compounds affect the protein expression in cells.

MapReduce is a framework used to distribute processing across clusters of computers that was first proposed by engineers working for Google [1]. It is designed to allow programming code to run on clusters of commodity hardware and abstracts the complexity of allocating, monitoring and running many parallel tasks away from the programmer. Although the basic methods are simple to understand and implement, a process will need to be re-engineered to fit into the strict system of Map and Reduce tasks.

To answer the question "Can pre-processing of Mass Spectrometer data approach real-time using a MapReduce process on a horizontally scalable cluster", parallel processing is investigated on several platforms offering different types of storage or computation frameworks.

2. RELATED WORK

A review of the current literature reveals that there are currently very few references to the use of MapReduce style parallelization for proteomics data processing. Where it is referenced it is in relation to searches for matching peak-lists with databases to identify proteins [5][6] and not the algorithms required for parallel processing or the suitability of the various types of cluster available.

3. GOALS

In this research work we investigate methods of reducing the elapsed time to process the output from Mass Spectrometers created in the course of proteomics experiments. Several cluster-based technologies that allow parallel computation are to be included.

- Hadoop – Distributed file system
- Cassandra – NoSQL database
- Teradata Aster – MPP SQL database
- Spark – In-memory distributed processing
- Storm – distributed stream processing

Benefits and drawbacks of each system are to be evaluated from a number of different points of view including ease of use and maintenance, ability to integrate with other systems into a proteomics processing pipeline, disk space requirements above and beyond that of the original data file and overall time to process the data.

4. ALGORITHM DESIGN

Much has been published regarding the process of detecting peptides in the spectral data output from mass spectrometers. The algorithm implemented by the software "MaxQuant" [2] has been chosen as the base from which to develop a novel algorithm that can be executed in parallel on a shared-nothing cluster. MaxQuant is used in many Life Sciences laboratories and supports the output from the Thermo-Fischer mass spectrometers that are used at the University of Dundee.

5. CURRENT WORK

The standard data format for the output from proteomics experiments is called mzML which is an XML file format [3] consisting of a header section containing information about the environment in which the experiment was performed, plus a

section containing information about each scan performed. The mzML files are large often in excess of 5Gb and as with all XML files are difficult to distribute across a cluster and process in parallel. In order to process these files efficiently a new file format (.scmi) has been designed that contains only the information relevant to peptide identification. The parser that performs this conversion also copies the data from the source file system to the target cluster. The remainder of the information in the mzML file is still important to the complete understanding of the outcome of an experiment. A hybrid load architecture was discussed in previous work by Hillman [4] with metadata stored in a relational database and the scan information in a distributed file system.

The data processing is broken down into several steps as described in detail below. The first step is to identify peaks in the spectral signal within individual scans, these peaks occurring along the mass dimension and are called 2D peaks. The second step is to link the 2D peaks across the scans in the time dimension, these are know as 3D peaks.

A Map task is used to process the 2D peaks and this is where the scmi file format greatly simplifies processing over the original XML, as each record in the file up to a carriage return represents a single scan from the mass spectrometer. Each scan can be processed completely independently of the others, and therefore the 2D peak-picking process can be made to operate in an embarrassingly parallel fashion. The Map task decodes the Base64 binary arrays storing the mass-to-charge and intensity data and loads them into in-memory arrays of java objects. Each peak is detected by using a slope detection algorithm with overlapping peaks introducing some degree of complexity here. In addition, noise in the signal and the way the instrument measures the peptides mean that the peaks can be shifted slightly; however, it is possible to compensate for this by calculating a theoretical peak by fitting an ideal Gaussian curve to the data. Because of the presence of carbon isotopes, it is necessary to identify peaks within an isotopic envelope that represent the same peptide. This can be done in the same Map task as the 2D peak picking. As peaks are matched within an isotopic window the charge can be calculated which allows us to deduce the final mass of the peptide.

The 2D peaks identified so far are indicators of the presence of a peptide. The mass spectrometer carries out multiple scans over time and any one peptide will take several seconds to pass through the machine. To match 2D peaks across time, the data will need to be re-distributed around the cluster and written out to disk. As the peaks detected are clustered into compact groups across the mass range, a custom partitioner function is required to ensure that correct data distribution takes place. Efficient distribution of processing and avoidance of data skew is key to a performant parallel process. Once this has been done a Reduce task is used to build the 3D peaks across time. An algorithm has been developed for this taking into account certain biological rules such as peaks occurring within a mass window (chosen to be 7 ppm in this work) and the complexity of matching peaks which have missing segments and noise in the signal.

The 3D peaks also require a de-isotoping step, for this process the techniques for 2D isotopic peak detection can be reused with a modification to ensure peaks with similar profiles are matched. At this point we have calculated the mass and intensity of molecules and can either output the results or move on to further processing such as detection of stable isotope labeling by amino acids in cell culture (SILAC) and/or database search.

In this way, the 2D and 3D peak-picking process fits well into the MapReduce programming framework, and where data needs to be redistributed, the dataset has been greatly reduced by the initial peak picking in the Map Task.

6. EVALUATION

For development and testing purposes, a virtual cluster has been constructed on a custom built server using an 8-core AMD processor with 32Gb Ram and 5 individual SSD Drives (SSD Drives were found to produce far more consistent run times than spinning disks)

Currently Virtual Clusters have been created to run Hadoop 1.3, Teradata Aster 6.1, Cassandra 2.07, Hadoop 2.4 and Spark 1.2

The output from the parallel algorithm coded in the mapreduce framework using Java has been checked against 3 current methods of processing this data.

Proteowizard, a common command line tool that is freely available and capable of peak picking in the mz domain (2D picking).

MaxQuant a GUI tool that can produce output peaks in the mz (2D) and also time (3D) domains as well as many other features not discussed here.

The Spectracus system developed in the Lamond Lab at the University of Dundee[7].

In each case the results from the mapreduce code accurately reflect that produced from the other software.

7. FUTURE WORK

Following on from the validation of the output from the parallel algorithm, research into the most efficient platform to perform the processing can begin. Note that this is an evaluation of a parallel algorithm and the MapReduce framework; hardware acceleration technologies such as GPU or FPGA are therefore not included. The platforms to be evaluated are mentioned above and include batch as well as stream processing architectures. As each claim to be linearly scalable, once experimentation into the elapsed time for the processing on different platforms is complete it will be possible to calculate the system requirements to complete the steps in a given time frame that fits within the definition of real-time in this context.

8. REFERENCES

[1] Dean, S. Jeffrey;Ghemawat 2004. MapReduce: Simplified Data Processing on Large Clusters. *OSDI '04: 6th Symposium on Operating Systems Design and Implementation*. (2004).

[2] Jurgen Cox and Matthias Mann. MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantifica- tion. *Nature Biotechnology*, 26(12):1367–1372, November 2008.

[3] Deutsch, F. E.W.; Martens L; Binz P.;Kessner D.;Chambers M; Sturm M;Levander 2009. mzML: Mass Spectrometry Markup Language. (2009).

[4] Hillman, C. 2012. *Investigation of the Extraction, Transformation and Loading of Mass Spectrometer RAW Files*. University of Dundee, January 2012.

[5] Lewis, J. Steven;Csordas Attila;Killcoyne Sarah;Hermjakob Henning;Hoopmann Michael R;Moritz Robert L;Deutsch Eric W;Boyle 2012. Hydra: a scalable proteomic search engine which utilizes the Hadoop distributed computing framework. *Bioinformatics*. 13, (2012).

[6] Mohammed, M. Yassene;Mostovenko Ekaterina;Henneman Alex A.;Marissen Rob J.;Deelder Andre M.;Palmblad 2012. Cloud Parallel Processing of Tandem Mass Spectrometry Based Proteomics Data. *Journal of Proteome Research*. 11, (2012), 5101 ? 5108.

[7] Ahmad, Y; Rasheed 2014 Protein Finger Printing in Teradata. Implementation Documentation, University of Dundee