A Study in Hadoop Streaming with Matlab for NMR data processing

Kalpa Gunaratna[1], Paul Anderson[2], Ajith Ranabahu[1] and Amit Sheth[1]
[1]Ohio Center of Excellence in Knowledge Enabled Computing (Kno.e.sis)
Wright State University, Dayton, Ohio 45435
Email: { kalpa, ajith, amit }@knoesis.org
[2]Air Force Research Laboratory, Biosciences & Protection Division
Wright-Patterson AFB, Dayton, Ohio 45433
Email:paul.anderson2@wpafb.af.mil

Abstract

Applying Cloud computing techniques for analyzing large data sets has shown promise in many data-driven scientific applications. Our approach presented here is to use Cloud computing for Nuclear Magnetic Resonance (NMR) data analysis which normally consists of large amounts of data. Biologists often use third party or commercial software for ease of use. Enabling the capability to use this kind of software in a Cloud will be highly advantageous in many ways. Scripting languages especially designed for clouds may not have the flexibility biologists need for their purposes. Although this is true, they are familiar with special software packages that allow them to write complex calculations with minimum effort, but are often not compatible with a Cloud environment. Therefore, biologists who are trying to perform analysis on NMR data, acquire many advantages due to our proposed solution. Our solution gives them the flexibility to Cloud-enable their familiar software and it also enables them to perform calculations on a significant amount of data that was not previously possible. Our study is also applicable to any other environment in need of similar flexibility. We are currently in the initial stage of developing a framework for NMR data analysis.

1 Introduction

NMR spectroscopy is a well known technique to analyze biological samples. NMR spectrometers usually generate outputs in the form of a spectrum. In order to extract useful information from this spectral data, they have to be subjected to numerical processing such as base line correction and normalization that include complex computations.

Biologists who analyze NMR data sets often face the common problem of very large data files. A typical spectrum could be 3 MB (C13 spectrum) or 600 KB (1H spectrum) and one user alone may process several hundred spectra. Performing analysis over these extremely large datasets at once is difficult and sometimes impossible due to the limitations in memory and processing power a single computer can provide. Computing clusters and others types of distributed computing systems are generally used to analyze large datasets.

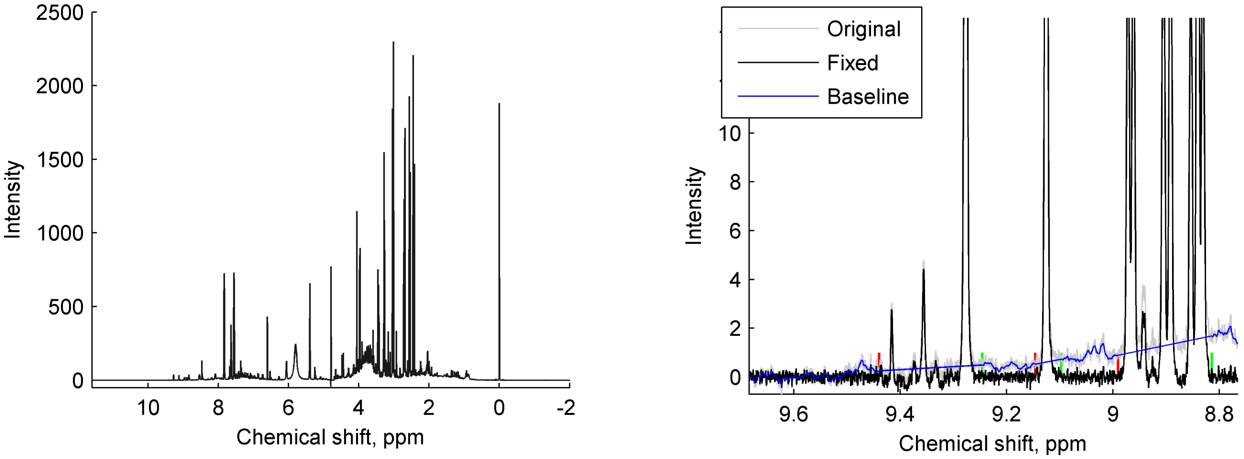
Scientists and biologists are familiar with scientific software tools that provide friendly environments for their particular needs. Matlab [7] is one such commercial software that provides specific data structures and modules that biologists need in their routine workflows. Matlab, however, typically runs as a desktop software and hence, constrained in computational power. Moving to a distributed environment may require mastering a set of new technologies and many scientists are hesitant to move away from the convenience of domain specific tools such as Matlab.

We make two observations in this context.
(1) There is an increase in the available computing power and distributed computing tools. These tools, however, have sharp learning curves and often discourage scientists from adopting them.
(2) User friendly and domain specific tools are deemed important by scientists. The convenience of such tools is often preferred over their apparent lack of performance.

If biologists are given the opportunity to run their familiar software in a distributed environment, then we are able to relieve them from the limitations of the computing power while providing convenient tooling. This represents a *best of both worlds* scenario.

In this paper we present the experience in our preliminary attempt to use Hadoop streaming [1] with Matlab. This enables scientists to use their familiar tools along with Apache Hadoop clusters. We also present the preliminary results obtained by using a baseline correction process.

978-0-7695-4302-4/10 $26.00 © 2010 IEEE
DOI 10.1109/CloudCom.2010.70



(a) Example Spectra with distorted and corrected baselines

(b) Enlarged portion of the spectra highlighting the baseline distortion and correction

Figure 1. Baseline distortion and correction

2 Motivation and Background

Our research is motivated by the difficulty scientists encounter in analyzing large data files conveniently. Matlab is a scientist-friendly tool although it does not take advantage of a distributed computing environment. By enabling the use of Matlab along with a computing cluster or a Cloud, scientists are capable of taking advantage of the available Matlab functionality as well as the power of computing Clouds.

Metabolomics

Metabolomics, the measurement of metabolite concentrations and fluxes in various biological systems, is one of the most comprehensive of all bionomics. Unlike proteomics and genomics that assess intermediate products, metabolomics assesses the end product of cellular function, metabolites. NMR spectroscopy of biofluids has been shown to be an effective method in metabolomics to identify variations in biological states. In contrast to various other methods, NMR spectroscopy is non-invasive, non-destructive, and requires little sample preparation [10].

A typical 1H NMR spectrum of pure proteins, biofluids, or tissue may contain thousands of resonances (sample shown in Figure 1). Analyzing NMR spectroscopic data requires a variety of computationally intensive algorithms that range from signal processing to pattern recognition techniques. The input to a specific processing technique may range from a single spectrum to multiple groups of spectra; however, the parallel nature of the processing steps is seldom realized.

Baseline distortion and correction

Baseline distortions have long been a problem in Fourier Transform NMR, which can arise from a number of hardware and processing sources [8]. These distorted baselines will result in incorrect metabolites quantification, thus, leading to spurious scientific conclusions. Further, subsequent signal processing algorithms, such as peak picking and alignment are also adversely affected by baseline distortions.

In some cases, baseline distortions can be removed by adjusting acquisition parameters, but often, baseline distortions remain, thus, a more general solution is often employed during post-processing [3, 2, 5, 11]. These algorithms employ a diverse set of techniques to model the baseline. In general, these algorithms build a baseline model from previously-detected baseline points, where the ideal baseline should match the baseline distortion while remaining smooth. A variety of techniques have been employed in the literature, including cubic splines [13], Bernstein polynomials [2], or polynomial functions [6] for smoothing, each of them with their own weaknesses. The Whittaker Smoother algorithm (WS) is commonly used to model spectral baseline [12, 4] since its capable of balancing both fidelity to data and the smoothness.

Baseline correction we performed using WS is shown in Figure 1 and a clear view of baseline correction with a closer view of Figure 1(a) is shown in Figure 1(b).

3 Implementation

Baseline correction

Fidelity to the data can be expressed as shown in equation (1) where z_i is the modeled baseline and y_i is the original spectrum. An estimate of smoothness can be expressed as the squared differences between neighbors as shown in equation (2).

$$S = \sum_{i=1}^{N} (y_i - z_i)^2 \tag{1}$$

$$R = \sum_{i=1}^{N} (z_i - z_{i-1})^2 \tag{2}$$

The WS algorithm balances these two goals as the sum: $Q = S + \lambda R$, where λ is a user defined parameter. This is a standard sum of squares problem with penalization, where we optimize z_i to minimize Q. Tunning the parameter λ will balance the relationship between fidelity and smoothness.

Optimizing the equation by setting the partial derivatives with respect to z_i equal to 0 yields:

$z = (I + \lambda D'D)/y$

where D is the derivative of the identity matrix I.

In the first stage of baseline correction, as previously described, baseline points in the spectrum are identified. These are incorporated into the equations using a weight vector w. This vector of weights is introduced in the fidelity term as in equation (3).

$$S = \sum_{i=1}^{N} w_i (y_i - z_i)^2 \tag{3}$$

Thus, the system of equations is updated to $(W + \lambda D'D)z = Wy$. Solving this system of equation results in a modeled baseline.

NMR Data Streaming for Matlab

In order to enable streaming to use Matlab, we compiled Matlab code and created a C++ shared library. A C++ driver application was written as the interface for mapper in Hadoop. The mapper driver invokes the Matlab mapper function, in this case the baseline correction implementation. This architecture is illustrated in Figure 2. The need for these wrappers is discussed in Section 4.1.

The NMR spectra are usually generated as column oriented data files, i.e the data values are present in rows. However Hadoop streaming architecture reads data files line-by-line. Hence we collected all the spectra to a single file and inverted the data, i.e a single row now represents the full spectrum. The wrapper creates the relevant Matlab object for a column and passes it to the Matlab function. The Matlab code for baseline correction was trivially changed from the desktop version.

In this particular case, a reducer was not used since each spectrum was represented in one line. If a spectrum spreads across multiple lines, then a reducer is needed to properly formulate the results.

The relevant Matlab files and the command reference is available in the Metabolink Web site[1].

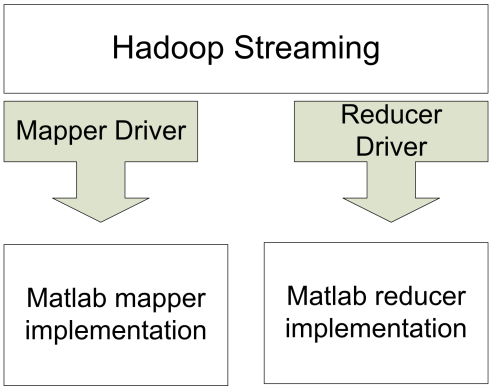


Figure 2. Using a driver to redirect input and output to Matlab implementations

Size	One Machine (Sec)	Cluster (Sec)
292KB (1 spectrum)	22	46
2.9MB (10 spectra)	192	152
28.6MB (100 spectra)	1996	1563

Table 1. Baseline correction performance

4 Discussion

There are two major advantages in using Matlab over Hadoop.

(1) Scientists are relieved from learning new technologies that often have sharp learning curves. For example, declarative scripting languages for distributed computations such as PIGLatin [9] are very verbose and incompatible with their non-distributed counter parts.

(2) Some of the required functionalities have already been implemented for non-distributed systems and are readily available. For example, Matlab already has a large number of statistical processes implemented and available as toolkits. Hence, there is no burden of reimplementing any of the necessary functionality.

The importance of these advantages come to light when the effort to use a raw distributed system is considered. For example, the Matlab distributed version[2] requires specific programming and expensive licensing. Scientists need to *think in parallel*, a paradigm shift that may take significant effort to get used to. The code adoption cost for such a porting is expensive and often repetitive. Furthermore, the capability to directly adopt the standalone implementation to a distributed environment facilitates rapid testing and prototyping.

[1] http://metabolink.org

[2] http://www.mathworks.com/res/distribtb

4.1 Technical challenges

We observed that Matlab had issues accessing the standard input and output provided by Hadoop streaming mechanism. Hence, we implemented the process with a C++ shared library and wrapped the Matlab function. This driver application's task is to read the input from Hadoop streaming and then call the mapper function passing the content as a string.

Another technical issue was the need for Matlab to be present in all the worker nodes in the cluster. The Hadoop streaming system only distributes the data and assumes the executables are present in the worker nodes. Unfortunately the error reporting mechanisms did not properly report this error. Since Matlab is a licensed program, one needs to make sure that proper licensing mechanism allows the given number of Matlab instances to run simultaneously.

4.2 Results

Our results reflect the advantages of running Matlab code for NMR data analysis as shown in Table 1. The cluster consists of 16 nodes, each node having a Quad-Core AMD Opteron cpu and 16GB of RAM. The single computer was a typical desktop with a dual core 3GHz cpu and 4GB of RAM.

For one spectrum the cluster is slower, probably due to the cost of distribution. The advantage of using the cluster becomes obvious with the increasing number of spectra.

5 Conclusion

The power of computing clouds may not be feasible for scientists if they have to significantly deviate from their current practices. Hadoop streaming allows the scientists to use their existing programs in Matlab with Hadoop clusters. Our preliminary experiments with NMR datasets show that using Matlab is indeed feasible and could be extended for various requirements.

References

[1] Apache Hadoop team. Hadoop Streaming. Online at http://hadoop.apache.org/common/docs/r0.15.2/streaming.html.

[2] D. Brown. Fully automated baseline correction of 1D and 2D NMR spectra using Bernstein polynomials. *Journal of Magnetic Resonance, Series A*, 114(2):268–270, 1995.

[3] W. Dietrich, C. Rüdel, and M. Neumann. Fast and precise automatic baseline correction of one-and two-dimensional NMR spectra. *Journal of Magnetic Resonance (1969)*, 91(1):1–11, 1991.

[4] P. Eilers. A perfect smoother. *Anal. Chem*, 75(14):3631–3636, 2003.

[5] S. Golotvin and A. Williams. Improved baseline recognition and modeling of FT NMR spectra. *Journal of Magnetic Resonance*, 146(1):122–125, 2000.

[6] P. Güntert and K. Wüthrich. FLATT-A new procedure for high-quality baseline correction of multidimensional NMR spectra. *Journal of magnetic resonance*, 96(2):403–407, 1992.

[7] D. Hanselman and B. Littlefield. *Mastering MATLAB 5: A comprehensive tutorial and reference*. Prentice Hall PTR Upper Saddle River, NJ, USA, 1997.

[8] J. Hoch and A. Stern. NMR data processing. *Physics in Medicine and Biology*, 42:611, 1997.

[9] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins. Pig Latin: A not-so-foreign language for data processing. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1099–1110. ACM, 2008.

[10] N. Reo. NMR-based metabolomics. *Drug and chemical toxicology*, 25(4):375–382, 2002.

[11] G. Schulze, A. Jirasek, M. Yu, A. Lim, R. Turner, and M. Blades. Investigation of selected baseline removal techniques as candidates for automated implementation. *Applied spectroscopy*, 59(5):545–574, 2005.

[12] E. Whittaker. On a new method of graduation. *Proceedings of the Edinburgh Mathematical Society*, 41:63–75, 1923.

[13] Z. Zolnai, S. Macura, and J. Markley. Spline method for correcting baseplane distortions in two-dimensional NMR spectra. *Journal of Magnetic Resonance (1969)*, 82(3):496–504, 1989.