The use of Bioinformatics Techniques for Time-Series Motif-Matching: A Case Study

Mark Transell and Carl Sandrock
Department of Chemical Engineering
University of Pretoria
Pretoria
marktransell@gmail.com

***Abstract*—Process engineers have more access to historical plant data than ever before. Finding recurring patterns in process data, also referred to as motif-matching, may reveal diagnostic information to engineers and operators. Dynamic Time Warping (DTW) is one of the most widely used techniques for performing these motif matches. Sequence matching is also an important part of bioinformatics; a field which has received a marked increase in research funding and attention in recent times. Therefore, the techniques developed in bioinformatics may be beneficial to the field of time-series motif matching. In this study, a combination of the Symbolic Aggregate Approximation (SAX) algorithm and the PSI-BLAST bioinformatics algorithm is compared to DTW as a potential method to perform time-series matches. Preliminary results suggest that this combination may be faster than global DTW techniques for large datasets. Details of the implementation are given, along with preliminary results confirming that this method is feasible. Due to implementation difficulties, accuracy and robustness remain uninvestigated. More research is recommended into the potential for this technique as an alternative to Dynamic Time Warping techniques.**

Index Terms*—Dynamic Time Warping; BLAST; motif-matching; time series; PAA

I. Introduction

It is often necessary for chemical and control engineers to diagnose a recurring plant behaviour, and for operators to receive alerts when undesirable plant behaviour is occuring. Dynamic Time Warping (DTW) was first applied to speech-recognition [1], but has since been applied to many fields [2] [3].

Sequence matching is also an important part of bioinformatics; a field which has received a marked increase in research funding and attention in recent times, helped in part by the publicity received by the Human Genome Project. This has led to the development of highly efficient algorithms and automated software implementations.

These freely avalable algorithms promise easier implementation, benefits to computational load and improved matching accuracy than DTW and similar methods. Particular attention is paid to the applicability of the PSI-BLAST algorithm combined with Symbolic Aggregate Approximation (SAX) to match process data. Time series need to be converted into character strings before Bioinformatics techniques can be applied to them. In this work, Symbolic Aggregate Approximation is used to convert the time series into strings, and PSI-BLAST is used to match these sequences.

II. Theory

A. Piecewise Aggregate Approximation (PAA)

In order to reduce the search space, it is often required to resample an original time series of length n to a reduced length w. A simple approach, known as Piecewise Aggregate Approximation (PAA), is to divide the entire time range into blocks and average over the blocks, as in Equation 1 [4].

$$\bar{c_i} = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} c_j \tag{1}$$

B. Symbolic Aggregate Approximation (SAX)

These continuous values need be quantized into character strings. The Symbolic Aggregate Approximation (SAX) method [4] uses breakpoints which will result in an equal probability of letters if the data were normally distributed. This is a highly desirable characteristic for sequence-matching algorithms as it makes the scoring matrices easy to calculate. Although the full procedure is outlined in [4] and [5], Figure 1 illustrates the basic concept of SAX: using the average value for a window of time series data to allocate a specific character.

PAA may acheive similar results to the more complex and computationally demanding methods [5], so it and SAX are used as the baseline technique for converting time-series data into character strings.

C. Wavelets

It has been shown that Haar Wavelet Transforms (HWT) can outperform discrete fourier transforms (DFTs) when reducing dimensionality in time series [6]. Although wavelets have the helpful multiresolution property, they are only defined for time series which are an integer power of two in length [4].

One important feature of wavelets and DFTs is that they are real valued. This limits the algorithms, data structures and definitions available for them. In anomaly detection, we cannot meaningfully define the probability of observing any particular set of wavelet coefficients, since the probability of observing any real number is zero [4].

D. Triangular Episodic Representation

Complete, correct, robust and compact models can be constructed using a small base of primitive representations for

 ISBN: 978-1-61208-237-0

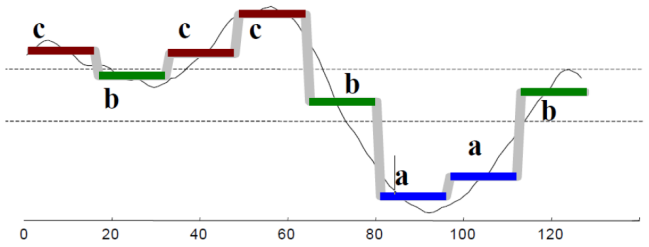

Fig. 1. An illustration of the SAX technique. The different colours are added for clarity, and characters are assigned to each window in the time series. Datapoints are numbered on the x-axis [5].

process behaviours [7]. The qualitative representations can be defined by any time over which the qualitative state of the process variable x is constant. This qualitative state is defined as in Equations 2 to 5.

$$\mathrm{QS}(x,t) = \begin{cases} \text{undefined if } x \text{ is discontinuous at } t \\ < [x(t)], [\partial x], [\partial^2 x] > \text{elsewhere} \end{cases} \quad (2)$$

Where:

$$[x(t)] = \begin{cases} + \text{ if } x > 0 \\ - \text{ if } x < 0 \\ 0 \text{ if } x = 0 \end{cases} \quad (3)$$

$$[\partial x(t)] = \begin{cases} + \text{ if } \partial x > 0 \\ - \text{ if } \partial x < 0 \\ 0 \text{ if } \partial x = 0 \end{cases} \quad (4)$$

$$[\partial^2 x(t)] = \begin{cases} + \text{ if } \partial^2 x > 0 \\ - \text{ if } \partial^2 x < 0 \\ 0 \text{ if } \partial^2 x = 0 \end{cases} \quad (5)$$

This triangular episodic representation gives seven basic types of episodes that can describe an interval of process behaviour [8].

This method is also currently being investigated, but preliminary results show no improvement over SAX techniques.

E. Shapelets

Time series shapelets [9] are used in image recognition, and rely on libraries of stored prominent motifs found within time-series data. Matches are classified using decision trees which compare subsequences that are maximally representative of a class.

Shapelets are selected based either on learning sets, or by using knowledge of the process in question to manually build decision trees. Process data may not necessarily be classified by this method appropriately; online chemical processes do not always have large enough learning sets to ensure accurate shapelet identification. Due to the unpredictable nature of process disturbances, operator knowledge of which motif to select as a relevant shapelet may be incomplete.

F. Dynamic Time Warping

Dynamic Time Warping [1] is often used as a faster and more robust method than Euclidian Distance to quantify the similarity of two time series [10]. It can also be used as a method to match subsequences [11], as implemented in the Machine Learning Python library (mlpy) [12].

It was selected as the baseline for time-series motif-matching, since there is a large amount of literature which employs this technique and it is popular in several applications, including speech and image recognition [1].

This algorithm employs a lower bounding technique based on a warping window or envelope. The most commonly used warping constraints are the Sakoe-Chiba band [1] and the Itakura parallelogram [13] [14].

G. BLAST (PSI-BLAST)

The BLAST algorithm [15] was developed to match strings representing nucleotide or protein fragments to large databases more efficiently than the FASTP [16] algorithm. The most recent iteration of the BLAST library includes the PSI-BLAST program [17], which matches queries in a gap-tolerant fashion. This makes the matching process more robust, and therefore better suited to finding *similar* strings as opposed to exact matches.

The PSI-BLAST algorithm uses scoring matrices to determine the similarity of two strings, the most popular being the BLOSUM62 substitution matrix [18].

H. Scoring matrices

The BLOSUM62 matrix [18] is based on amino-acid substitution; the probability that one amino-acid would replace another in a particular protein string. This scoring matrix is therefore not suitable for normally distributed time-series data. However, a scoring matrix can be calculated directly during the SAX procedure, based on the assumption that all characters are equiprobable [5].

I. Performance Metrics

The metrics used to quantify the performance of the matching algorithms compared are:

- Computing time taken to complete a search, and the ability of the algorithm to scale with database size.
- The accuracy of the matches found, with respect to Euclidean distance and DTW cost for the matches [10], [11], [19].
- The ease by which each algorithm can be implemented. (Does an algorithm require system file access? How many hardware and software dependencies are there?)
- Tolerance to datasets with noise, time-axis shifting, vertical shifting or time warping.

III. Implementation

Figure 2 shows how a library of time-series and query data is processed by each algorithm to produce a match with quantifiable accuracy.

The development framework necessitated the use of the NCBI BLAST+ executable library, available on their website [17]. The language selected for the coding framework was Python 2.7, due to the availability of the Biopython library

 ISBN: 978-1-61208-237-0

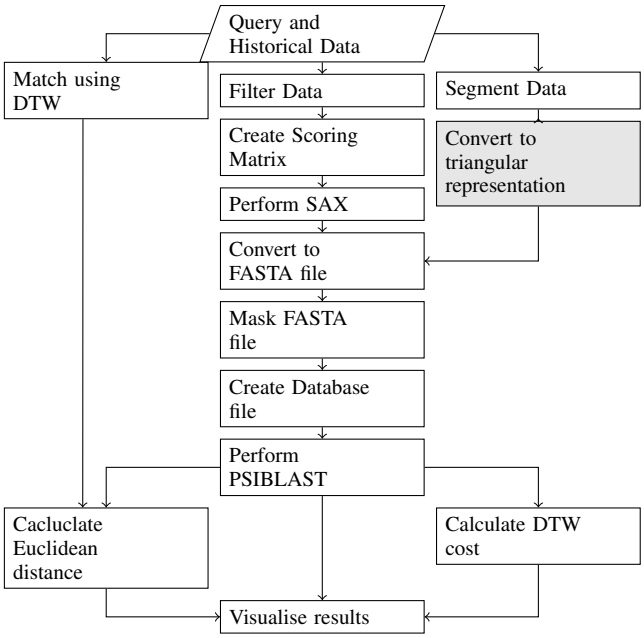

Fig. 2. A simplified overview of the data flow for testing

for use with bioinformatics programs such as BLAST [20]. Additional python libraries were used:

- numpy [21] for data handling
- matplotlib [22] for plotting purposes
- mlpy [12] for the DTW algorithm
- scipy [23] for data filtering.

The breakpoints for SAX conversion are calculated based on the cumulative distribution of the database data, thus ensuring that every character in the database string is equiprobable. [5] The implementation of BLAST matching requires the operating system used to be a Linux-based system, as it is necessary to overwrite the BLOSUM scoring matrix directly with a custom scoring matrix generated during the SAX process. [4]

The SAX-PSIBLAST method has four paramaters, namely the size of time window and number of breakpoints used for SAX, and the expected error value and search word size given as inputs to PSI-BLAST. These parameters enable the user to adjust the degree to which fuzzy matches can be found, increase or decrease noise tolerance in the dataset, or to adjust the number of potential matches listed for any given query.

The SAX-PSIBLAST procedure does not make use of training sets to perform matches. This allows non-mutated test data to be used for speed tests. In order to test whether a matching algorithm is performing correctly, two base cases are examined. These cases are:

1) A query time-series is matched to itself.
2) A query time-series is matched to a time-series which includes, but is not restricted to the query itself.

IV. Results and Discussion

The PSI-BLAST technique is faster than the global DTW algorithm on the self-matching task for large datasets. Figure 3

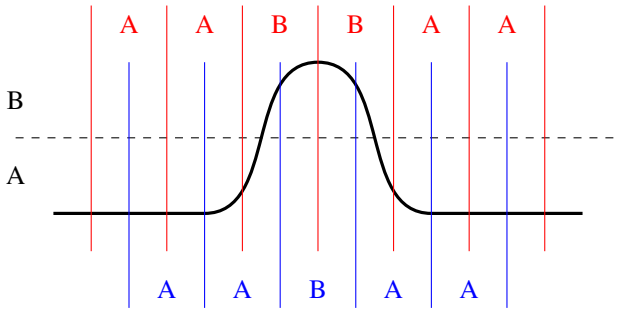

Fig. 4. An illustration of how shifting a PAA time window can alter the string sequence produced by SAX. The upper sequence output is AABBAA, but the lower sequence is AABAA, even though they represent the same time series. This can cause the PSI-BLAST algorithm not to match these series correctly.

shows the processing times for global DTW and the SAX-PSIBLAST method for different PAA average-value windows. The mlpy DTW algorithm fails on the two largest test data sets.

However, the PSI-BLAST algorithm does not reliably find a full match for the second base case after dimensionality reduction. In some cases a partial match is returned, or the match is erroneously extended to include adjacent data. Possible reasons for this failure include:

- The internal workings of the PSI-BLAST executable do not use the modified scoring matrix correctly.
- The modified scoring matrix is in a form which, for statistical reasons, does not fit the specifications for PSI-BLAST scoring matrices. (This could possibly be remedied by intelligent selection of the expected error value and word-size)
- The queried data is shifted during the PAA step for any time window, which may cause the string representation of the query and dataset to be slightly different.

Figure 4 illustrates how the string representation may be altered by shifts in the time-windows for PAA.

V. Conclusions and Recommendations

Preliminary results suggest that a motif-matching algorithm which employs a combination of Symbolic Aggregate Approximation (SAX) and the use of the PSI-BLAST algorithm is faster than conventional Dynamic Time Warping (DTW) techniques. Additional metrics remain to be compared.

It is currently not known whether the partial failure of the SAX-PSIBLAST method to find embedded sequences when large time windows are used is due to implementation error, or to inherent limitations in the PSI-BLAST algorithm when processing dimensionally reduced sequences.

References

[1] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, pp. 43–49, 1978.

[2] T. M. Rath and R. Manmatha, "Word image matching using dynamic time warping," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, vol. 2, p. 521.

 ISBN: 978-1-61208-237-0

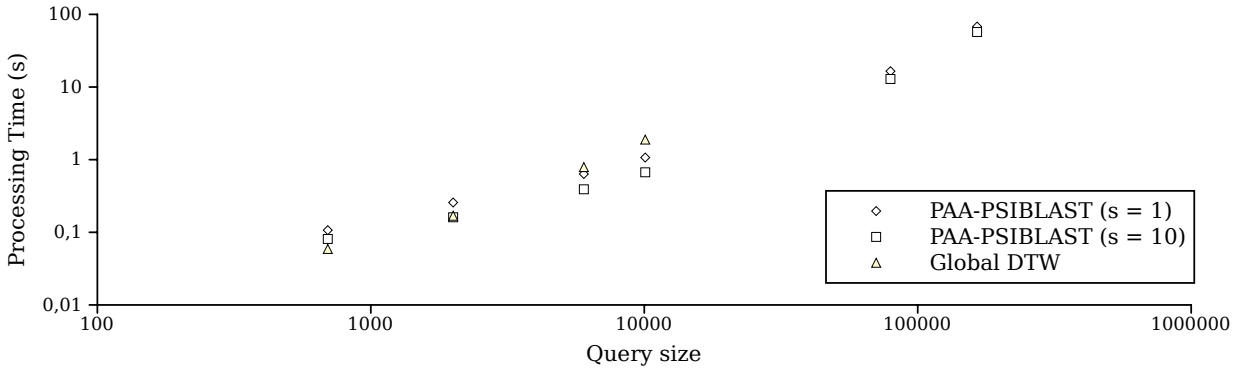

Fig. 3. Base case processing time comparison on Intel Core i5 processor

[3] K. Santosh, "Use of dynamic time warping for object shape classification through signature," *Kathmandu University Journal of Science, Engineering and Technology*, vol. 6, pp. 33–49.

[4] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, ser. DMKD '03. New York, NY, USA: ACM, 2003, pp. 2–11. [Online]. Available: http://doi.acm.org/10.1145/882082.882086

[5] B. Chiu, E. Keogh, and S. Lonardi, "Probabilistic discovery of time series motifs," in *The 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 1, 2003, pp. 493–498.

[6] K. Chan and A. Fu, "Efficient time series matching by wavelets," in *Proceedings of the 15th IEEE International Conference on Data Engineering*, Sydney, Australia, March 1999, pp. 126–133.

[7] J.-Y. Cheung and G. Stephanopoulos, "Representation of process trends part i. a formal representation framework," *Computers and Chemical Engineering*, vol. 14 (4/5), pp. 495–510, 1990.

[8] S. Kivikunnas, "Overview of process trend analysis methods and applications," in *Proceedings of the ERUDIT Workshop on Applications in Pulp and Paper Industry*, University of Oulu, Finland, 1998.

[9] L. Ye and E. Keogh, "Time series shapelets: A new primitive for data mining," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. DMKD '03. New York, NY, USA: ACM, 2009.

[10] E. J. Keogh and M. J. Pazzani, "Scaling up dynamic time warping for datamining," University of California, Tech. Rep., 2000.

[11] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," Stern School Of Business, New York University, Tech. Rep., 1994.

[12] D. Albanese, S. Merler, G. Jurman, R. Visintainer, and C. Furlanello, "Mlpy machine learning py," 2012, http://mloss.org/software/view/66/.

[13] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-23, pp. 52–72, 1975.

[14] C. A. Ratanamahatana and E. Keogh, "Everything you know about dynamic time warping is wrong," in *3rd Workshop on Mining Temporal and Sequential Data, in conjunction with 10th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2004.

[15] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of Molecular Biology*, vol. 215, pp. 403–410, 1990.

[16] D. Lipman and W. Pearson, "Rapid and sensitive protein similarity searches," *Science, New Series*, vol. 227(4693), pp. 1435–1441, 1985.

[17] S. Altschul, T. Madden, A. Schffer, J. Zhang, Z. Zhang, W. Miller, and D. Lipman, "Gapped blast and psi-blast, a new generation of protein database search programs," *Nucleic Acids Research*, vol. 25(17), pp. 3389–3402, 1997.

[18] J. Setubal and R. Braeuning, *Similarity Search*, A. Gruber, A. Durham, and C. H. et al., Eds. National Center for Biotechnology Information (US), 2006.

[19] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: Experimental comparison of representations and distance measures," *VLDB*, 2008.

[20] P. J. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski1, and M. J. L. de Hoon, "Biopython: freely available python tools for computational molecular biology and bioinformatics," *Bioinformatics*, vol. 25, pp. 1422–1423, 2009.

[21] P. F. Dubois, K. Hinsen, and J. Hugunin, "Numerical python," *Computers in Physics*, vol. 10, no. 3, May/June 1996.

[22] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, May-Jun 2007.

[23] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–. [Online]. Available: http://www.scipy.org/

 ISBN: 978-1-61208-237-0