Qualitative Analysis of Segmented Time-series by Sequence Alignment

Balazs Balasko, Sandor Nemeth, and Janos Abonyi

University of Pannonia, Department of Process Engineering,
P.O. Box 158, H-8201 Veszprem, Hungary,
mail: *abonyij@fmt.uni-pannon.hu* web: *www.fmt.vein.hu/softcomp*

Abstract. *Data analysis is often associated with quantitative techniques because of the large amount of data and easy-to-use statistical tools. Qualitative trend analysis (QTA) techniques always have to be guided with some data reduction method, e.g. principal component analysis (PCA) or segmentation, and the preprocessed, lowered size data can be analyzed for further aims. This article proposes the application of triangular episode segmentation to compress process data. This geometric language for describing trends is applied to convert time series into symbolic sequences. Its main advantage is that based on this technique, one is able to qualify, compare or classify different time series if there is an adequate distance measure. This article proposes pairwise sequence alignment as a common tool in bioinformatics, but never reported in QTA literature. Distance between two evolving DNA sequences in this field of science means dissimilarity: how many mutation steps are needed to transform the old DNA into the new one. Every step has its mutation penalty value, hence distance is a numerical value defined as the sum of transformation weights. This technique can be applied for alignment of sequences where the optimal alignment is given by the minimal sum of weight, hence it is applicable for time series comparement and qualification. It is shown that aligning episode sequences is able to analyze historical process data of an operating plant.*

Keywords: triangular episodes, pairwise sequence alignment, qualitative trend analysis

1 Introduction

There are several ways in the data mining literature to analyze the large amount of time series. Quantitative methods are widely spread because of their statistical nature, it is always easy to calculate the average, the covariance or the quantiles from a data set but it always claims prior knowledge to analyze the results, i.e. to use these basic statistics of a time series for qualitative analysis.

A common method for decreasing the size of a data set and to get qualitative instead of quantitative information is time series segmentation. Segmentation means finding time intervals where a trajectory of a state variable is homogeneous [1]. Segments can be linear, steady-state or transient, indicative for normal, transient or abnormal operation. So to extract some useful feature from time series of the state variables one needs to lower the size of the data and define a distance measure from a theoretically optimal solution to help operators in their work.

Cheung and Stephanopoulos in [2] proposed a new segmentation method for process trend analysis, the application of episodes with a geometrical representation of triangles. Triangular episodes use the first and second derivatives of a trend

on a geometrical basis, hence seven primitive episode can be achieved as characters. Originally, this framework was developed for trend identification as a guide of wavelet decomposition based analysis.

Lot of researchers in the literature found feature extraction by episodes useful for fault diagnosis, decision support service or system monitoring, but also a lot of them modified the definition of primitives. E.g. Venkatasubramanian defined 9 primitive episodes and identified them as a pattern recognition problem with feed-forward neural networks [3]. Charbonnier et al. declared only 3 primitive episodes as {*Increasing, Decreasing, Steady*}, and classified the aggregated segments into seven temporal shapes {*Increasing, Decreasing, Steady, Positive Step, Negative Step, Increasing Transient, Decreasing Transient*}for trends extraction from noisy, unfiltered signals [4]. Gamero et al. combined Principal component analysis and episode based qualitative trend analysis for predicting instabilities in a blast surface [5]. They applied several set of episodes, extended from {*rapid fall, fall, rise, rapid rise, steady-high, steady-medium, steady low*}. Wong et al. fuzzified these primitive episode representation by magnitude and duration in order to have 57 episodes for trend representation and for Hidden Markov Model-based classification for fault diagnosis [6]. In fact, they have not fuzzified but hard partitioned the seven primitive triangular episodes by the maximal fuzzy membership value, but it was a good innovation to increase the number of representing symbols and making the description of trends more qualitative.

Besides decreasing the amount of data by triangular episode segmentation, a powerful analyzing technique is needed for comparing sequences of symbols, i.e. there is a claim for a distance measure. Sundarraman et al. in [7] defined a distance measure in their work, that calculated the matching degree (MD) by shape, magnitude and duration for two trends. They applied first order trend description (only first derivatives are captured) to dynamically synchronize real valued trends and theoretical ones taken from dictionary: first, it is compared which boundary point of a segment in the dictionary corresponds to which boundary point in the real trend, then these corresponding segments are compared by simple ratio by magnitude and duration. In other words, theoretical and real time trends are *aligned* in their work based on shape matching degree (they assumed that trends are always identical by shape if operation is normal).

This type of synchronization is well known in dynamic time warping (DTW), which allows comparison of data points or intervals at different time points hence extreme features (e.g step) located elsewhere in two trends can easily be sychronized. It is based on a dynamic distance matrix that is filled up with the distance values of every time points (It is an $m \times n$ matrix for comparison of two trends with m and n data points) and the 'cheapest', i.e. shortest path in the distance matrix is the result of DTW. It was originally developed for speech recognition [8][9], but its application in process analysis is also known. Srinivasan et al. showed as an example that DTW is able to compare symbolic sequences, like DNA and applied their technique (called dynamic locus analysis) also for process fault diagnosis[10].

Although DTW is a powerful method for synchronizing time series, it cannot correctly handle nascent or vanishing features, like one step in a trend is executed in two smaller steps in an other. For this purpose this article suggests a technique taken from an other discipline, from Bioinformatics, for comparing sequences that is called pairwise sequence alignment. Sequence alignment is a well known method to compare amino acid or nucleotide sequences, and to measure the distance of two sequences, called (dis)similarity in bioinformatics. There are not only mutation and substitution but deletion and insertion operators as well, so it can handle excess features by injecting gaps, but possible alignments of two sequences increase exponentially by the length of the sequences. Needleman and Wunsch proposed a fast algorithm that finds the optimal alignment of two sequences by a similarity function[11], i.e. it maximizes a similarity index. Based on this idea, several other algorithms came into existence that applies the minimization of a distance function [14] or the sum of transformation weights [12][13], which is the cheapest path in a distance matrix just like DTW. A more objective method is the application of maximum likelihood estimation, which compares all considerable sequence alignment between two sequences, but it needs a stochastic model for describing macromolecule evolution and the parameter values where the alignment has maximal possibility, hence the statistic sequence alignment proposed by Thorne et al.[15] - also known as TKF91 model - is not applicable in our study. For time series analysis, MATLAB's Bioinformatics Toolbox was applied, within local and global sequence alignment is implemented for amino acid or nucleotide sequences.

Second and third section of the article deals with the basic theoretical background of triangular episode representation of a trend and pairwise sequence alignment, Section 4 shows the implementation of the algorithm and an explaining example while Section 5 presents a case study for product changing quality of an operating polymer plant. the article ends with conclusions and future work plans.

2 Triangular Episode Sequences

Cheung and Stephanopoulos created a precise formal framework for trend analysis, hence their nomenclature are used to describe triangular episodes. To get from a quantitative to a qualitative representation of a real-valued $x(t)$ function, it has to fulfill the following properties over a closed time interval $[a\ b]$ (note, that *time* is represented as a sequence of open intervals separated by time points):

1. $x(t)$ is continuous over $[a\ b]$ or it has a finite number of discontinuities in its value or derivetive;
2. $x(t)$ has a finite number of extrema over $[a\ b]$;
3. $x'(t)$ and $x''(t)$ are continuous in $(a\ b)$ and at a and b they have existing onesided limits;

The above requirements satisfying functions are called *reasonable functions*. It is clear that all the psychical variables in a plant operation are reasonable. It is considered if we know the value and the derivatives of a reasonable function, the

state of that function is completely known. The continuous state (CS) over $[a\ b]$ can be defined as a point value, which is a triplet (if $x(t)$ is cont. in t):

$$CS(x,t) \equiv PtVl(x,t) = \langle x(t),\ x'(t),\ x''(t) \rangle \tag{1}$$

If $x(t)$ is discontinuous in t, then left- and right-hand limits of the derivatives are used.
Consequently, a continuous *trend* can be defined as continuous sequence of states over $[a\ b]$. For discrete functions, as an approximation, an underlying continuous function has to be known since the derivatives of single points in $[a\ b]$ cannot be performed. These definitions lead to a qualitative description of a state (QS) and trend if x is continuous at t, otherwise it is undefined:

$$QS(x,t) = \langle [x(t)],\ [\partial x(t)],\ [\partial\partial x(t)] \rangle, \tag{2}$$

where $[x(t)]$,$[\partial x(t)]$ and $[\partial\partial x(t)]$ can be $\{-,0,+\}$, depending if they have negative, zero or positive values. Obviously, a qualitative trend of a reasonable variable is given by the continuous sequence of qualitative states over $[a\ b]$.
When $QS(x,t)$ is constant for a maximal time interval (the aggregation of time intervals with same QS) is called an *episode* (see Fig. 2.(a).) and the final definition of a trend of a reasonable function is a sequence of these maximal episodes.
An ordered sequence of triangular episodes is the geometric language to describe trends. It s composed of seven primitives noted as $\{$ `A,B,C,D,E,F,G`$\}$ illustrated in Fig. 2.(b).

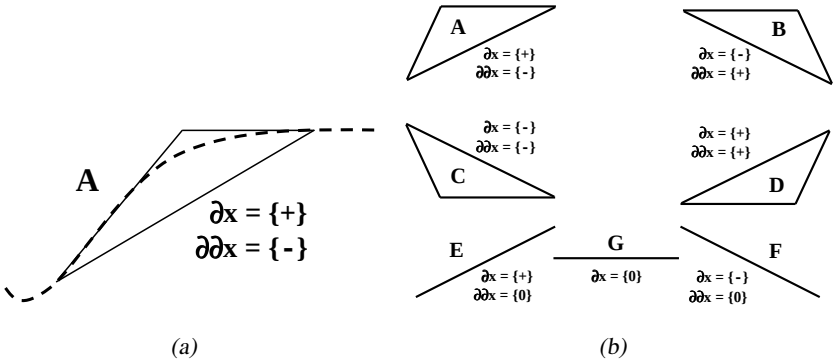


Fig. 1. An example for an episode (a) and the seven primitive episodes proposed by Cheung and Stephanopoulos (b).

Wong et al. fuzzified these seven primitives into a fuzzy set of 57 episodes with a fuzzy membership function . Every episode is assigned to be $\{small(s), medium(m),$

large$(l)\}$ by duration and magnitude with the highest membership value. Actually, this means that they hard partitioned the episodes by values of the intersections of the membership function. Fig. 2 shows how 9 episodes are created from one; in {xyZ} 'fuzzified' episode notation, x means change by magnitude, y means change by duration and Z means the main episode primitive (A to F). Episode {G} is an exception while there is no sense in partitioning this episode by magnitude, it is only partitioned by duration for {sG,mG,lG}

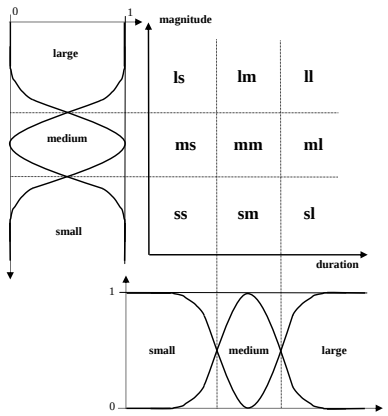


Fig. 2. Episodes partitioned by duration and magnitude.

3 Sequence Alignment

Sequence alignment is typical expression of bioinformatics, where amino acid or nucleotide sequences have to be compared, how far the evolved new sequences are from the elders, i.e. how old they are, and how many mutation steps were needed to result in the new sequence. As the simpliest example to undrstand, the following example shows how two, one-character-long sequences (C and A) can be aligned:

$$\begin{array}{ll} \text{(1.)} & \text{(2.)} \\ -\mathtt{C} & \mathtt{C} \\ \mathtt{A}- & \mathtt{A} \end{array}$$

In bioinformatics, notation on the left means that an Adenosine injection was performed left to a Cytosine while Cytosine was deleted; and notation on the right symbolizes that a Cytosine was substitued with an Adenine amino acid while evolving a sequence. The distance / similarity-based methods seek for an alignment of minimal distance or maximal similarity, hence the sequence evolution on the left is

penalized with $w(- \rightarrow \mathtt{A}) + w(\mathtt{C} \rightarrow -)$ while the other is penalized with $w(\mathtt{C} \rightarrow \mathtt{A})$. The w mutation weights are calculated from their probabilities, that is why usually $(w(- \rightarrow \mathtt{A}) + w(\mathtt{C} \rightarrow -)) < w(\mathtt{C} \rightarrow \mathtt{A})$, so one substitution is more probable than an injection and deletion next to each other. Consequently, in this way the optimal alignment of sequences underestimates injection and deletion.
Applying the *minimal evolution*, one tries to find the least mutation steps between the elder and offspring sequence. Naive algorithms compares all possible alignments and selects one with minimal sum of transformation weights.
Fast algorithms calculate in an other way: Let A_n be a n-element sequence and B_m a m-element sequence, a_n and b_m their nth and mth element; $\alpha * (A_n, B_m)$ denotes the set of optimal pairwise alignments of A_n and B_m and $w(\alpha * (A_n, B_m))$ the sum of transformation weights for these optimal alignment. The basic idea in fast algorithms is that if we know $w(\alpha * (A_{n-1}, B_m))$, $w(\alpha * (A_n, B_{m-1}))$ and $w(\alpha * (A_{n-1}, B_{m-1}))$, then $w(\alpha * (A_n, B_m))$ can be calculated within a constat time period. If we leave the last aligned pair in an optimal alignment of A_n and B_m then we get an optimal alignment of (A_{n-1}, B_m), (A_n, B_{m-1}) or (A_{n-1}, B_{m-1}), depending on that last mutation step was a deletion, injection or substitution, respectively:

$$w(\alpha * (A_n, B_m)) = \min\{ \begin{array}{c} w(\alpha * (A_{n-1}, B_m)) + w(a_n \rightarrow -); \\ w(\alpha * (A_n, B_{m-1})) + w(- \rightarrow b_m); \\ w(\alpha * (A_{n-1}, B_{m-1})) + w(a_n \rightarrow b_m) \end{array} \} \tag{3}$$

The optimal alignment weights are given in a dynamic programming matrix, D with a size of $(n+1) \cdot (m+1)$. The initial conditions for the 0th row and column:

$$d_{0,0} = 0; \tag{4}$$

$$d_{i,0} = \sum_{l=1}^{i} w(a_l \rightarrow -); \tag{5}$$

$$d_{0,j} = \sum_{k=1}^{j} w(- \rightarrow b_k); \tag{6}$$

Equation (3) is the way for filling up D. Optimal alignments are started at $d_{n,m}$ and ended in $d_{0,0}$, while in every step the minimal weight is chosen and stepping left means an injection, stepping upwards means a deletion and stepping diagonally upwards means a substitution.
This method was developed by Needleman and Wunsch[11]. It penalizes more deletion after each other as a simple sum of weights. Latter algorithms use gap functions to penalize longer gaps in a sequence alignment. In out study, the original gap penalty is applied because of the independent episodes.

MATLAB's Bioinformatics Toolbox was developed to handle these operations in a simple MATLAB function. Its origin used popular log odds cost matrices like PAM or Blossum for pairwise alignments of amino acids. Fortunately, manual scoring matrix as an input can be given as well, thus after some modifications the algorithm was able to calculate with the distance matrix of 57 triangular episodes and global alignment of the episode sequences.

4 Implementation of the algorithm

The proposed algorithm is built from the following operations:

1. Filtering and preprocessing of data to be analyzed;
2. Time series segmentation into sequence of primitive triangular episodes;
3. Partitioning of episodes by magnitude and duration;
4. Alignment of sequences achieved by Step 1 to 3.

Process signals are often loaded with noise arised from process equipments faults actuator dynamics external disturbances etc. For smoothing a signal while keeping its main characteristics, the application of the widely known and used Gaussian filter is recommended. An $f(t)$ noisy time series can be transformed into an $F(\sigma,t)$ smoothed function by the following convolution kernel (Gaussian transformation:)

$$F(\sigma,t) = f(t) \circ g(\sigma,t) = \int_{-\infty}^{+\infty} f(u) \cdot \{\frac{1}{\sigma\sqrt{2\pi}} \cdot \exp[\frac{-(t-u)^2}{\sigma^2}]\} du \tag{7}$$

Considering σ as a smoothing parameter, its increasing value results in more and more smoothed signals, where pikes and high frequency features are vanished, hence shorter episode sequences are achieved.

After first and second order derivatives are calculated, segmentation algorithm transforms quantitative data into semi-qualitative episodes description and after that, it distributes episodes into 'fuzzified' episodes by predefined thresholds: these thresholds depend on the range of magnitude and duration of data, hence they have to be adjusted for every problem. Note that duration here means number of data points in an episode, hence it can be only converted back to time if sampling time of data is constant for the whole set.

For sequence alignment, first a manual 7×7 scoring matrix was defined for the basic primitive episodes, where the filling-up-theory was the following:

- Each score of a mutation can be between 0 and 10, except gaps that have a scoring value of -5 (injecting a gap is penalized);
- $w(X \to X) = 10$, perfect alignment gains a score of 10 (unlike orignal alignment, w denotes here the score of an alignment, that is why $w(X \to X) \neq 0$);
- Score of the alignment of an increasing $\{\texttt{A,D,E}\}$ and a decreasing episode $\{\texttt{B,C,F}\}$ is zero;
- Every episode is similar to the steady episode $\{\texttt{G}\}$ with a score of 2.
- Scoring matrix has to be diagonal to fulfill the requirement $w(X \to Y) = w(Y \to X)$;

To have the 63×63 main scoring matrix, scores of the fuzzified episode alignments have to be predefined as well (see Table 4). For episode $\{\texttt{G}\}$ the 9×9 matrix is filled up with the 3×3 submatrix.

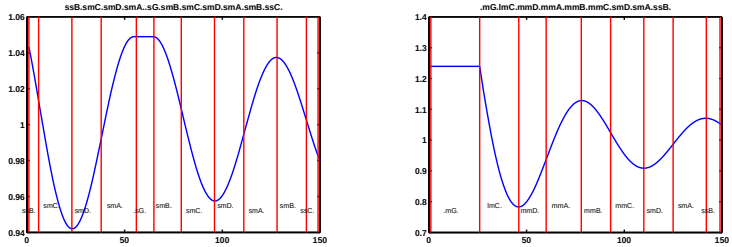
	ss	sm	sl	ms	mm	ml	ls	lm	ll
ss	10	8	6	8	6	4	6	2	1
sm	8	10	8	6	8	6	4	4	2
sl	6	8	10	6	6	8	2	4	4
ms	8	6	6	10	8	6	8	4	2
mm	6	8	6	8	10	8	6	8	4
ml	4	6	8	6	8	10	4	6	8
ls	6	4	2	8	6	4	10	8	6
lm	2	4	4	4	8	6	8	10	8
ll	1	2	4	2	4	8	6	8	10

and for {G}:

	s	m	l
s	10	8	6
m	8	10	8
l	6	8	10

Table 1. Scoring matrices of the fuzzification for primitive episodes.

As a simple application example, Fig. 4(a) shows a $\sin(x)/x$ function where $x = (15 : 0.1 : 29)$ and a 10-data-points-long steady segment is added at the first extrema, while the same function is shown on Fig. 4(b), where $x = (2.5 : 0.1 : 15)$ following a 25-data-points-long constant segment. That means that both time series 'lasts' for 150 data points, the function is the same but located elsewhere in time and 'foreign' steady segments are added to trends. Predefined thresholds are $[0.1, 0.3]$ for change of magnitude and $[10, 30]$ for duration (again, number of data points in an episode).



(a)

Fig. 3. The manually generated time series

Triangular episode segmentation resulted in an 11 and a 9-character-long episode sequence. Alignment of these trends by the seven primitive triangular episodes is as follows(' | ' denotes a perfect match; ' : ' signs a mutation):

```
BCDAGBCDABC
:|||  |||||
GCDA-BCDAB-
```

As one can see, the algorithm recognizes the injected steady-state G episode because it injects a gap into the second sequence and it proposes mutation in the case of the the starting G episode to get the optimal alignment. The ending gap is compulsory to have two sequence with the same length. So, it looks like 8 of 11 episodes are identical (73%), but there are significant differences which can only be recognized if fuzzified episodes are applied ('/' means a mutation in at least one fuzzy attribute):

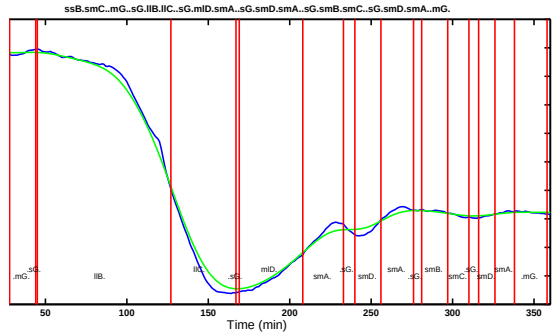
```
ssB.smC.smD.smA..sG.smB.smC.smD.smA.smB.ssC
  :   /   /   /       /   /   |   |   /
.mG.lmC.mmD.mmA.gap.mmB.mmC.smD.smA.ssB.gap
```

Because of the nature of the $\sin(x)/x$ function, the amplitudes decrease by increasing x values, and the algorithm recognizes these changes by magnitude: only 2 of 11 are identical episodes (18.2%) the other 6 one, which seemed to be identical as well, differ in fuzzy attributes. The score of this alignment is 55.2 from 110, which means a scoring percentage of 50%.
To summarize it, the algorithm showed that these trends are very similar by shape (73%) but they differ by change of magnitude and duration of segments (18.2% of episodes are the same).

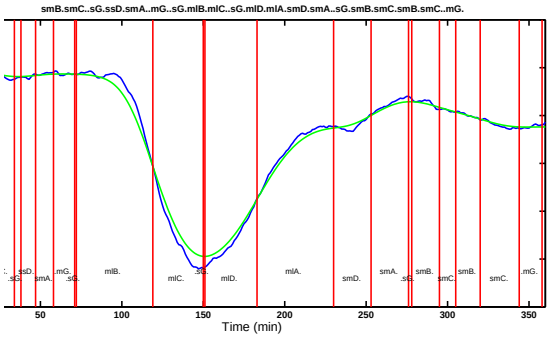
5 Case Study

This section deals with product changing strategies of a polymer plant in Hungary. The plant produces mainly polypropylene homopolymer in two loop reactors in series. Quality of the polymers is indicated by a derived variable called melt flow index (MFI). The value of MFI depends mainly on the average polymer molecule weight, i.e. chain length, which can be controlled with two process variables: (i) hydrogene flow into the reactors and (ii) residence time in the reactors. Residence time is closely correlated with polymer density because the more time the molecules spend in the reactors, the longer chains come into existence thus the higher the density value will be. So to keep product quality in control, operators adjust the setpoint of polymer density during product changes because of the following reasons: Slurry density set point has to be lowered significantly to decrease resident time in reactors, i.e. production rate, and then it has to be stepped up in 2-3 steps onto the level corresponding to the new product. This strategy minimizes the produced byproduct (note that byproduct means polymer with not sufficient MFI).

A time interval of 6 hours (360 data points with a sample time of 1 minute) is analyzed for each product change, 3 hours before and after product change. If a change of density is smaller than 5 kg/m^3,episodes get label 'small', if it is larger than 15 kg/m^3 it is labelled as 'large', else it is 'medium'. These values for duration were 10 and 30 data points (i.e minutes). A product change is shown in Fig. 5 for comparing the densities in the two reactors.



(a)



(b)

Fig. 4. Product change indicated by slurry density in the first(a) and second(b) loop reactor with a scoring percentage of 61.4%.

Alignment of the basic episodes is as follows:

```
BC---GGBCGDAGDAGBCGDAG
||   ||||||| |||||: :|
BCGDAGGBCGDA-DAGBCB-CG
```

```
ssB.smC.gap.gap.gap..mG..sG.llB.llC..sG.mlD ...
   /   |               |   |   /   /   |   |
smB.smC..sG.ssD.smA..mG..sG.mlB.mlC..sG.mlD ...

... smA..sG.smD.smA..sG.smB.smC..sG.smD.smA..mG
      /       |   |   |   |   |   :       :   |
... mlA.gap.smD.smA..sG.smB.smC.smB.gap.smC..mG
```

These alignments show that density values during product change in first and second loop reactor look similar to each other (in almost every cases), 15 of 22 (68 %) episodes are identical in the simple alignment and just 4 from this identitiy set are mutated in fuzzy attributes (50% is identical). Score of this global alignment is 116.6 from a maximal value of 190 (length of the longer sequence multiplied by 10), which means 61.4%. Considering that 5 gaps mean -25 in the score, *one can appoint that these trends are similar qualitatively*, thus product changes are well-coordinated in the two loop reactors.

For further analysis of this case study, the same product changes at different dates may be compared, product changes can be classified by their effect on MFI value or effects of application of other catalyst systems can be analyzed. Unfortunately, lack of place does not allow the presentation of these results, hence this case study is just an example how the algorithm works on areal process data set.

6 Conclusion

The proposed algorithm can extract useful information from quantitative time series. For decreasing the large amount of data and for resulting in a qualitative description of trends, it applies triangular episode segmentation. To qualify and compare these episode sequences, it proposes the application of pairwise sequence alignment, which is a common technique in bioinformatics, but no article reports its usage in the field of process trend analysis. Sequence alignment is an analogous technique to dynamic time warping (DTW) but it can handle nascent or disappearing features in a trend because of its evolutional basics. It has been shown that this tool is able to find similarities in two trends and compare them based on a scoring 'distance' matrix. It has several other potential applications such as qualifying time series based on theoretically optimal trends, search for similar sequences in a time series or classification of trends. Our future work will concentrate on these other application possibilities to show how useful this algorithm is in qualitative process data analysis.

Acknowledgement

The authors would like to acknowledge the support of the Cooperative Research Centre (VIKKK) (project 2004-I) and Hungarian Research Found (OTKA T049534). Janos Abonyi is grateful for the support of the Bolyai Research Fellowship of the Hungarian Academy of Sciences.

References

1. Keogh, E., Chu, S., Hart, D. and Pazzani, M.: An Online Algorithm for Segmenting Time Series, IEEE International Conference on Data Mining, 2001
2. Cheung, J. T., Stephanopoulos, G.: Representation of process trends. Part I. A formal representation framework, Computers and Chemical Engineering, vol. 14, pp. 495-510, 1990
3. Venkatasubramanian, V.: A Syntactic Pattern-recognition Approach for Process Monitoring and Fault Diagnosis, Engineering Applications of Artificial Intelligence, vol. 8, no. 1, pp. 35-51, 1995
4. Charbonnier, S., Garcia-Beltan, C., Cadet, C., Gentil S.: Trends extraction and analysis for complex system monitoring and decision support, Engineering Applications of Artificial Intelligence, vol. 18, pp. 21-36, 2005
5. Gamero, Fco. I., Colomer, J., Melendez, J., Warren, P.: Predicting aerodynamic instabilities in a blast furnace, Engineering Applications of Artificial Intelligence, vol. 19, pp. 103111, 2006
6. Wong, J.C., McDonald, K.A. and Palazoglu, A.: Classification of process trends based on fuzzified symbolic representation and hidden Markov models, J. Proc. Cont. vol. 8, No. 5-6, pp. 395-408, 1998
7. Sundarraman, A., Srinivasan, R.: Monitoring transitions in chemical plants using enhanced trend analysis, Computers and Chemical Engineering, vol. 27, pp. 1455-1472, 2003
8. Itakura, F.: Minimum prediction residual applied to speech recognition, IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-23(1),pp. 67-72, 1975
9. Sakoe, H. and Chiba, S: Dynamic programming algorithm optimization for spoken word recognition, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 27(1), pp. 43-49, 1978
10. Srinivasan, R., Qian, M.S.: Online fault diagnosis and state identification during process transitions using dynamic locus analysis, Chemical Engineering Science, vol. 61, pp. 6109 6132, 2006
11. Needleman, S.B, Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology, vol. 48, pp. 443453, 1970
12. Waterman, M.S, Smith, T.F, Beyer W.A.: Some biological sequence metrics. Advanced Mathematics, vol. 20, pp. 376387, 1976
13. Waterman, M.S.: General methods of sequence comparison, Bulletin of Mathematical Biology, vol. 46, pp. 473500, 1984
14. Sellers, P.H.: On the theory and computation of evolutionary distances. SIAM Journal of Applied Mathematics, vol. 26, pp. 787-793, 1974
15. Thorne J.L., Kishino H., Felsenstein J.: An evolutionary model for the maximum likelihood alignment of DNA sequences. Journal of Molecular Evolution, vol. 33, pp. 114-124, 1991