Where Traffic meets DNA: Mobility Mining using Biological Sequence Analysis Revisited*

Ahmed Jawad, Kristian Kersting, and Natalia Andrienko
Knowledge Discovery Dept., Fraunhofer IAIS
53754 Sankt Augustin, Germany
firstname.lastname@iais.fraunhofer.de

ABSTRACT

Traffic and mobility mining are fascinating and fast growing areas of data mining and geographical information systems that impact the lives of billions of people every day. Another well-known scientific field that impacts lives of billions is biological sequence analysis. It has experienced an incredible evolution in the recent decade, especially since the Human Genome project. Although, a very first link between both fields has been established already in the early 90ies, many recent papers on mobility mining seem to be unaware of it. We therefore revisit the link and show that many unexplored and novel mobility mining methods fall naturally out of it. Specifically, using advanced discretization techniques for stay-point detection and map matching, we turn traffic sequences into a "biological" ones. Then, we introduce a novel distance function that enables us to directly apply the rich toolbox for biological sequence analysis to it. For instance, by just looking at complex traffic data through the biological glasses of sequence logos we get a novel, easy-to-grasp visualization of data, called "Traffic Logos". For clustering and prediction tasks, our empirical evaluation on three real-world data sets demonstrates that revisiting the link can yield performance as good as state-of-the-art data mining techniques.

Categories and Subject Descriptors

H.2.8. [**Database applications**]: Data mining, Spatial databases and GIS.

General Terms

Applications, Experimentation

Keywords

mobility mining, visualization, biological sequence analysis

1. INTRODUCTION

Location-based services signify a change in how we as a society use computers to mine data. As Mitchell pointed out [12], we are beginning to analyse 'our reality' — data recording personal activities, conversations, and movements — in space and time in an attempt to improve human health, guide traffic, and advance the scientific understanding of human behaviour in general. In a sense, sensor-equipped computing devices overcome longstanding temporal and spatial boundaries to human perception [**?**]. Therefore it comes as no surprise that mining one type of reality data, namely movement and traffic data, is currently receiving a lot of attention. Another well-known scientific field that is receiving a lot of attention is biological sequence analysis(see e.g [5] for an introduction) especially since the Human Genome project. Additionally, both of these fields have something in common i.e. the identification of relevant patterns in massive sequential information. Indeed, whereas biological sequence analysis focusses on sequences of (few) symbols, mobility mining focusses on sequences of continuous values. Thus, one may argue that building bridges between them is insurmountable. However, it is not the case as the first link between biological sequence analysis and mobility has been established already in the early 90ies (see e.g. [1]). In this paper we revisit the link established as many recent papers on mobility mining seem to be unaware of it. Furthermore, previous efforts were not exploiting the link to the full extent. They either extended standard biological sequences alignment to the multi-dimensional case [**?**] — loosing the biological glasses to view further into this direction for clustering, classification, visualization, and probabilistic modelling of data, among other tasks — or they used standard similarity scores from biology [16, 15] and in turn run the risk of missing the invariances of traffic sequences.

In this paper, we demonstrate that advanced descretization techniques (e.g. map matching [8], stay point extraction, etc.) together with a novel, data-driven similarity score allows one to keep wearing the biological glasses while keeping the traffic invariance intact. Specifically, our main contributions are (1) a novel, data-driven similarity score suitable for traffic data, (2) the introduction of 'Traffic Logos', a novel visualization technique that provides a compact yet descriptive view on the *information content* of traffic sequences, and (3) the demonstration of state-of-the-art performance for important traffic analysis tasks such as *user activity analysis* using 'off-the-shelf' biological techniques with our similarity score. Essentially, *user activity analysis* from geographic data comprises models that abstract a person's movement from raw GPS data to places of interest and analyse travel rhythms [14] between them.

Mobility mining techniques naturally deal with the two most common problems that come with traffic data: **(a)** Traffic data is composed of sequences over continuous time and space and not discrete symbols. **(b)** The amount of raw traffic data is huge. Since we abstract the raw data into sufficiently small alphabets using standard discretization, we can instantly solve **(a)** and in turn **(b)**. Why? Biological sequence analysis will do the rest for us. It was designed to deal with large numbers of variable length sequences.

*The authors would like to thank the three anonymous reviewers for their valuable comments. This work was supported by the Fraunhofer ATTRACT fellowship STREAM and a "Higher Education Commission" (HEC)-Pakistan scholarship.

ACM SIGSPATIAL GIS '11 ,November 1-4, 2011. Chicago, IL, USA
Copyright 2011 ACM 978-1-4503-1031-4/11/11 ...$10.00.

2. FROM RAW TRAFFIC TO SYMBOLS

Let $\mathcal{X}$ be some raw traffic data. We now convert $\mathcal{X}$ into a set $\mathcal{S}$ of traffic sequences using a so-called translation method $\mathcal{M}$ i.e $\mathcal{M} = (\mathcal{A}_{\mathcal{M}}, \Delta_{\mathcal{M}}, \mathcal{F})$ where $\mathcal{A}_{\mathcal{M}} = \{a_1, a_2, ..., a_l\}$ is an alphabet (set of symbols the sequences are composed of) and $\Delta_{\mathcal{M}}$ is an l-by-l matrix of pair-wise similarities between symbols in $\mathcal{A}_{\mathcal{M}}$. Now, a traffic sequence $T_{\mathcal{M}}$ for $\mathcal{M}$ is a temporally tagged sequence of symbols chosen from alphabet $\mathcal{A}_{\mathcal{M}}$, that is $T_{\mathcal{M}} = \{(a_{t_1}, t_1), (a_{t_2}, t_2), ..., (a_{t_e}, t_e)\}$. Finally, $\mathcal{F}$ denotes a discretization function which maps raw traffic data $\mathcal{X}$ to the set of traffic sequences $\mathcal{S}$ according to $\mathcal{M}$, that is $\mathcal{F}(\mathcal{X}) \leftarrow \mathcal{S}_{\mathcal{M}}$. In general, the discretization function to be used in application dependent. Examples include map matching i.e the process of assigning raw trajectories to street segments, see e.g. [8], region based division of Euclidean space in T-Pattern mining, frequency bins from sensor readings and stay point extraction from user trajectories, among others. Let us now touch upon the alphabet and similarity score used in more detail. Every symbol $a \in \mathcal{A}$ corresponds to a set of traffic objects. Therefore, it is natural to assume that for any two symbols $a_i, a_j \in \mathcal{A}_{\mathcal{M}}, a_i \cap a_j = \emptyset$, that is a_i and a_j correspond to disjoint/non-overlapping sets of traffic objects. Note that the symbols usually represent spatial and unary objects like regions of a city or streets in a street network, however they can also represent non-spatial entities of interest like frequency bins for sensor readings or categories of streets like highway, link road, etc. The similarity matrix $\Delta_{\mathcal{M}}$ describes the similarity between symbols in $\mathcal{A}_{\mathcal{M}}$. In the context of computational biology, $\Delta_{\mathcal{M}}$ is driven by the following insight: two molecules have higher similarity if they can be converted through chemical reactions readily and vice versa. Therefore, standard matrices have been developed. For traffic applications, the situation is different. There is a multitude of traffic data sets, all with their own characteristics and invariants. Hence, it is unlikely that there is a single good similarity matrix. Instead, it depends upon the application at hand. For example, we have chosen shortest path distances for the model where the input alphabet consists of streets from a street network and the application of interest is 'trajectory clustering'. For cases, where we do not have such domain knowledge available, we now propose a 'data driven' approach to devise a similarity matrix Δ. To illustrate, we turn a sequence into a graph in the following way. Each unique symbol in the sequence is a node. Then if two symbols are consecutive in the sequence, there is an edge between the corresponding nodes in the graph. Finally, we weight the edge with the average temporal differences between the two symbols in the sequence. Now, we calculate the shortest path distances between all nodes in the graph. If there are muliple sequences, we simply average all resulting distance matrices. Unfortunately, it may very well happen (in particular for rather small data sets) that there are pairs of symbols which never co-occur in a traffic sequence. In turn, the *average temporal difference* distance cannot be computed. For example, in the dataset we used for the *analysis of user activities*, the user never does sports and shopping in a sequence together. In this case, we assign some value larger than the maximum similarity values computed for the 'observed' symbol pairs. In other words, we just ensure that the two symbols are maximally dissimilar. We note that now we are in a very similar situation as the well-known IsoMap approach for computing low-dimensional Euclidean embedding [9]. Simply following it, i.e., we embed the weighted graph into Euclidean space $\mathbf{R^2}$ resulting in distances d_{ij} using *multi-dimensional scaling* [9]. This new distance respects well the intrinsic geometry of the data manifold described by the weighted graph. Finally, we turn the Euclidean distances into similarities by using RBF kernels [8], i.e., $\Delta_{ij} = \exp(-d_{ij})$. Now, we have everything together to run the traffic sequence analysis techniques as discussed in the last section. For instance, we can align a set of sequences. However, we can do even better. For instance, standard alignment assumes that the time lapsed between two consecutive symbols is constant. This is not true for most traffic data. To accommodate for variable-size steps in time, that means to balance between duration of a time step and the Euclidean distance between the two corresponding symbols, we add a penalty term to the Euclidean distance between them. Specifically, let π^* denote the alignment between two traffic sequences s and s' of length m and n respectively. Furthermore, let $d(s_i, s'_j)$ denote the distance after embedding between symbols in s and s' at position i and j. Now, we define a similarity based on d:

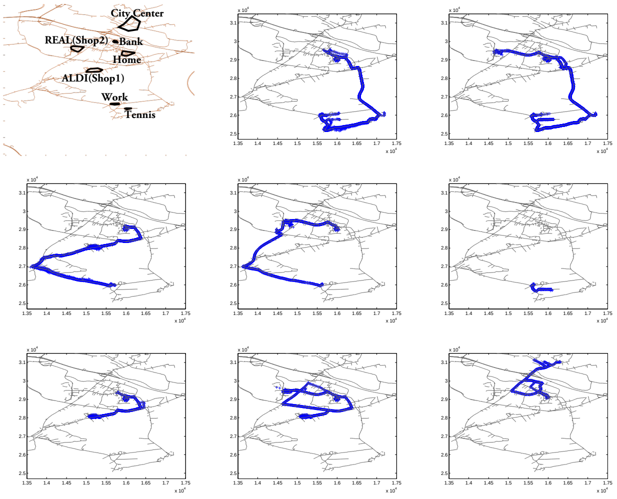


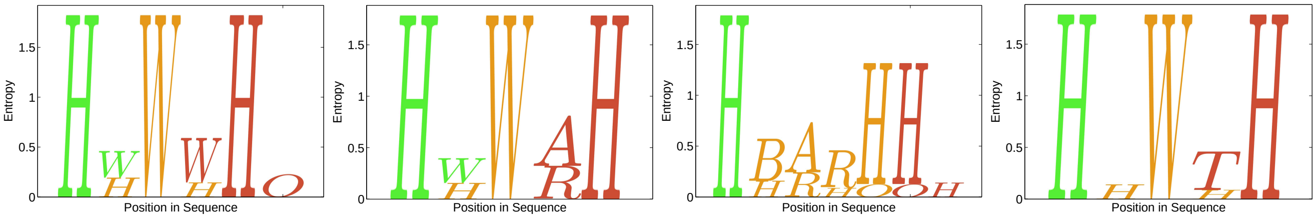
Figure 2: (a). Convex hulls of labelled stay points (blue polygons over gray-edged street network). The main stay points are Home, Office, Bank, Tennis, Shopping-1(ALDI), Shopping-2(REAL) and City Center. (b–i)8 Clusters of raw trajectories (blue points) based on map-matched street segments from street nework (gray edges). Additionally, cluster plots describe specific routes that user follows between stay-points. Clustering algorithm used is DBscan [6](a density based clustering algorithm) with $\epsilon = 0.58$ and $min_n = 3$. Clustering captures 262 objects, describe in order of cluster labels from 'a' to 'h' i.e $\{119, 78, 28, 11, 7, 7, 6, 6\}$. After clustering, trajectories are projected back in the original space and shown collectively for each cluster in a separate plot. Clusters of routes indicate office travel, shopping routine, sports, weekend routine and city center roaming. Finally, intra-cluster distance is low as all trajectories in a cluster are compact and follow same route.

$$d'(s_i, s'_j) = d(s_i, s'_j) + \lambda \cdot (t_i - t'_j)^2 \qquad (1)$$

where $\lambda \in [0, 1]$ denotes the regularizer for variable-size time steps. Its value is application dependent. In case of a gap, we simply fix the gap penalty as a constant i.e $d'(s_i, -) = d'(-, s'_j) = c$
Now, we simply use the alignment algorithm in [4] to compute the optimal alignment π^* and it's score $\theta(\pi^*)$ using the similarity $\Delta' = \exp(-d')$. Moreover, we can naturally turn the score of the alignment into a similarity score among pairs of whole sequences by normalizing it i.e.

$$K_{s,s'} = \frac{\theta(\pi^{\star}_{s,s'})}{sqrt(\theta(\pi^{\star}_{s,s}), \theta(\pi^{\star}_{s',s'}))} \qquad (2)$$

Now, we finally have everything together to employ the toolbox for biological sequence analysis for mobility mining.



(a) Work Routine(87 objects) (b) Work & Shopping(68 objects) (c) Weekend Routine(34 objects) (d) Work & Sports(26 objects)

Figure 1: Traffic logos for stay-point based clustering after *approx* 99% compression of original data. x-axis denotes sequence positions and y-axis denotes 'information' present in each column. Every cluster describes one of the possible routines that user follows in her daily life. The symbols in the figure denote labels of activities based on stay points i.e: H denotes staying at Home; W–Working; A–shopping at ALDI; R–shopping at Real; B–getting cash from Bank, T–playing Tennis and O denotes Other activities for leisure (i.e city center roaming and visiting friends). Colours of symbols denote the time of day i.e green denotes Morning(before 9.am); yellow–daytime(9am-6pm) and blue-evening(after 6pm). The height of a symbol denotes the certainty of an acitivty in the clusters. For example, staying at work place in the evening is less certain in than going Home in the evening (cluster 'a'). Similarly, during weekends, doing other activities in the day after shopping is less certain than coming back home(cluster 'c').

3. MOBILITY MINING USING BIOLOGY

To validate the usefulness of biological sequence analysis method for traffic data, we investigated the following questions: **(Q1)** Is it possible to solve Traffic problems with the help of out-of-the-box biological sequence analysis methods? **(Q2)** If so, how do they perform compared to state-of-the-art methods? **(Q3)** Can we gain interesting insights into traffic data with the help of biological sequence methods? To answer these questions, we choose two mobility mining tasks being investigated by GIS community, namely 'Traffic event detection' and 'Analysis of user activities'. We followed two complementary approaches to analysing user routines at different abstraction levels. In the first approach, we extracted daily sequences of the user's stay points, clustered them using alignments, and analysed the resulting clusters using traffic logos. In the second one, we digged deeper and analysed the user's routines using 'map-matched' trajectories. This helps in grouping functionally relevant trajectories and in turn in identifying specific routes over the street network. Specifically, we used DBscan [6] using the the pair-wise alignment score and then visualized the resulting clusters. The second approach also helps us in comparing the performance to state-of-the-art methods (as we will show). Both approaches were applied to the same dataset of 112k recorded position within 363 trajectories , see [2] for details.

(Q1, Q3) Stay Points Discretization: To extract frequent stay points of user, first we marked GPS positions from raw trajectories, which stayed within a radius $r = 100$ meters for time $t = 10$ minutes. Then we clustered these marked points with the help of DBscan [6] to find area which are more dense among these marked positions. In the end, we took the convex hull of each cluster to get the shape of a stay point. Our next step is stay point labelling. To do this, we first looked at the temporal distributions of the stay points in order to label the most important stay points, in our case 'home' and 'work'. The rest of the stay points are labelled with the help of Google maps (e.g restaurants, post office, bank, shopping markets, Tennis court, etc.). The extracted stay points along with their labellings are shown in Fig. 2; for the sake of keeping privacy, we are omitting the latitudes and longitudes. After the extraction of stay points, we built the similarity matrix as described above using time regularization. These stay points served as the symbols for activities in our traffic sequences. We calculate distance matrix between daily activity sequences from user with pairwise sequence alignment. Then, the sequences were clustered based using DBscan [6]. The sequences of each clustering were additionally aligned and we produced traffic logos for them shown in Fig. 1(a-d). As one can see, traffic logos show a very dense and illustrative view of clusters for user's daily routines. Fig. 1(a) describes the largest cluster. This is an affirmative answer to questions **(Q1)** and **Q3)**.

(Q1-Q3) Map Matching Discretization: To investigate whether our methods can perform comparably with state-of-the-art methods, we focused on clustering trajectories [13]. Whereas the state-of-the-art method clusters raw trajectories, we used the sequences of map matched street segments to cluster our data set. This helps us in analysing specific map routes that user selects during her travels. For map matching we followed [8]. After clustering on the map matched level, we projected the labelled trajectories back into Euclidean space and visualized them over the street networkin Fig. 2. As one can see, we find meaningful clusters. However, are they also as good as state-of-the-art? To see this, i.e., for a quantitative comparison, we computed the Hausdorff distance for both clustering solutions as well as the error. The error term RMS_{STD} is a measure of clustering compactness which gives *sum of average within cluster variances* [11]. For clustering, our alignment-based method used again *DBscan [6]*. As state-of-the-art baseline, we used *OPTICS [3]* route similarity search, see [13]. As both clustering methods are density based and filter outliers, this comparison is fair. Using K-means as baseline for example is not a fair option as it does not filter outliers. In other words, it will produce apriori much higher error. The parameters of both density based algorithms are ϵ (minimum similarity threshold to consider two points as neighbours) and min_n (minimum number of neighbours needed to define a point as a 'core point'). We used a grid search to determine the best parameters for the same number of clusters as found by the baseline, namely 7. Our method produced (using a grid search on $\epsilon \in [0.5 : 0.01 : 1.0]$ and $min_n = [2 : 1 : 6]$) an error of $RMS_{STD} = 197$ with parameter settings $\epsilon = 0.65$ and $min_n = 5$ as compared to $RMS_{STD} = 271$ with $\epsilon = 1KM$ and $min_n = 5$ for the baseline. However, the number of trajectories clustered by our method was 232 whereas the baseline clustered 261 trajectories. The remaining ones of the in total 363 trajectories were filtered out. This, however, is only giving a 'point estimate' of our performance. To see the general picture, i.e., performance over the range of grid search, we provide the performance averaged per number of clusters in Tab.1. As one can see, *pairwise traffic sequence alignment* is able to capture similar number of objects with a better mean of error than the baseline by filtering out the noise in a better way. We believe that this happens because of high gap costs in the alignment computation. They penalize to group together trajectories with dissimilar sub-parts. Consequently, more compact clusters are found. More-

Similarity method	Clusters	μ_{RMS}	σ_{RMS}	$\#objs(\mu \pm \sigma)$
Route Search	7	271.5	NA	261
Pairwise Traffic Sequence Alignment	7	249.24	36.4	255.6 ± 7
	5	211.78	112.43	221.2 ± 21
	6	236.58	75.74	243.6 ± 14
	≥ 8	313.98	20.13	275.1 ± 6

Table 1: Comparison of clustering results using the Hausdorff distance to compute errors. As we can see both capture a similar number of trajectories and similarly good clusters, see also Fig. 2(a-g). Row 3,4 have a lower error but also capture a smaller number of trajectories because of small ϵ and large min_n. Row 5 shows that the alignment-based method can capture more patterns than the original method, cf. Fig. 2(h), at the cost of a higher error with large ϵ and small min_n.

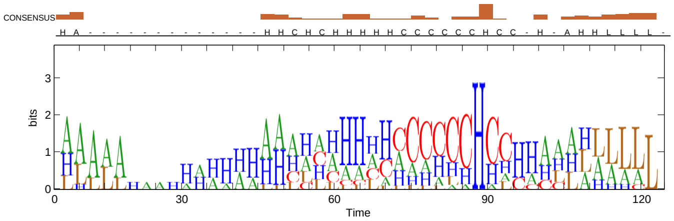


Figure 3: Traffic Logos for Baseballs games in *training set* for '+-1 hour' of game endings. Symbols denote frequency of traffic i.e. L-Low, A-Avg , H-High and C-Congestion. (Top)5-conserved regions. (From left to right) region 1 describes 'low to average' traffic during play. Regions 2–4 collectively describe traffic frequency for game endings and show high density traffic with real congestion starting approx. 15 minutes after game endings. This trend declines approx. 45 minutes after the games where conserved region region no.5 starts showing a tendency towards normal traffic i.e average and low density.

over, it finds a similar number of clusters as the baseline, namely $7-8$. Fig.2 shows the 8-distinct patterns found by our algorithm. Here, the clusters in Figs 2(a-g) were also found by the baseline. The additional cluster shown in Fig.2(h) is Home to City Center. Moreover, we contacted the owners of the dataset and they agreed with the possibility of clusters found. This is clearly an affirmative answer to question **(Q1)- (Q3)**.

(Q3) Visual Analysis of Sensor Data: We consider a real world data sets also used by [7]. This loop sensor data was collected for a free-way in Los Angeles. It is close enough to the stadium to see unusual traffic after a Dodgers game, but not so close that the signal for the extra traffic is overly obvious. The observations were taken over 25 weeks with 288 time slices per day in 5 minute counts. Here, the goal is to analyse the presence of congestion during baseball games at Dodgers stadium. We discretize the traffic frequencies into Low (L), Average (A), High (H) and Congestion (C) and then prepare a profile HMM from the traffic sequences related to base ball games. A comparison of profile HMM with the test data for event detection shows that we get a better recall (lower number of false positives) for event detection compared to [7]. Due to space constraints we cannot provide full results, however we show traffic logos in Fig. 3 to reveal the trends in training data .

4. CONCLUSION AND FUTURE WORK

In this paper, we have revisited the link between 'biological sequence analysis' and 'mobility mining'. Specifically, we demonstrated that advanced discretization techniques can be combined with a novel, data-driven similarity score and used with off-the-shelf biological sequence analysis techniques to get state-of-the-art performance. We introduce Traffic Logos, which provide a condensed, yet illustrative of picture of patterns in traffic sequence data. There are several attractive avenues for future work. First of all, one should investigate the benefits of out-of-the-box biological sequence techniques for other traffic mining applications. We are currently working on generating more complex profiles and diaries of user's activities to compare them for getting user similarity.

5. REFERENCES

[1] A. Abbott. A primer on sequence methods. *Organization Science*, 1(4):375–392, 1990.

[2] G. Andrienko, N. Andrienko, and S. Wrobel. Visual analytics tools for analysis of movement data. *SIGKDD Explor. Newsl.*, 9:38–46, December 2007.

[3] M. Ankerst, M. M. Breunig, H.P. Kriegel, and S. Jörg. Optics: Ordering points to identify the clustering structure. In *ACM SIGMOD*, pages 49–60, 1999.

[4] G. d. Vries and M. Someren. Clustering vessel trajectories with alignment kernels under trajectory compression. In *ECML PKDD*, pages 296–311, 2010.

[5] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis*. Cambridge University Press, 1998.

[6] M. Ester, H.P. Kriegel, S. Jörg, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.

[7] A. Ihler, J. Hutchins, and P. Smyth. Adaptive event detection with time-varying poisson processes. In *ACM SIGKDD*, 2006.

[8] A. Jawad and K. Kersting. Kernelized map matching. In *ACM SIGSPATIAL*, GIS, pages 454–457, 2010.

[9] C. Joh, T.A. Arentze, and H.J.P. Timmermans. Multidimensional sequence alignment methods for activity-travel pattern analysis: A comparison of dynamic programming and genetic algorithms. *Geographical Analysis*, 33(3):247–270, 2001.

[10] J.A. Lee and M. Verleysen. *Nonlinear dimensionality reduction*. Springer, New York; London, 2007.

[11] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu. Understanding of internal clustering validation measures. In *IEEE*, ICDM '10, pages 911–916. IEEE Computer Society, 2010.

[12] T. Mitchell. Mining our reality. *Science*, 326(5960):1644–1645, 2009.

[13] S. Rinzivillo, D. Pedreschi, M. Nanni, F. Giannotti, N. Andrienko, and G. Andrienko. Visually driven analysis of movement data by progressive clustering. *Information Visualization*, 9:225–239, 2008.

[14] A. Schmidt, M. Langheinrich, and K. Kersting. Perception beyond the here and now. *IEEE Computer*, 44(2):86–88, February 2011.

[15] S. Schonfelder and K.W. Axhausen. *Urban Rhythms and Travel Behaviour: Spatial and Temporal Phenomena of Daily Travel (Transport and Society)*. Ashgate, 2010.

[16] N. Shoval and M. Isaacson. Sequence alignment as a method for human activity analysis in space and time. *Annals of the Association of American Geographers*, 97(2):282–297, 2007.

[17] C. Wilson. Analysis of travel behavior using sequence alignment methods. *Journal of the Transportation Research Board*, 1645(-1):52–59, 1998.