

A New Measure for Analyzing and Fusing Sequences of Objects.

John Y. Goulermas, *Senior Member, IEEE*, Alexandros Kostopoulos, Tingting Mu, *Member, IEEE*

Abstract—This work is related to the combinatorial data analysis problem of seriation used for data visualization and exploratory analysis. Seriation re-sequences the data, so that more similar samples or objects appear closer together, whereas dissimilar ones are further apart. Despite the large number of current algorithms to realize such re-sequencing, there has not been a systematic way for analyzing the resulting sequences, comparing them, or fusing them to obtain a single unifying one. We propose a new positional proximity measure that evaluates the similarity of two arbitrary sequences based on their agreement on pairwise positional information of the sequenced objects. Furthermore, we present various statistical properties of this measure as well as its normalized version modeled as an instance of the generalized correlation coefficient. Based on this measure, we define a new procedure for consensus seriation that fuses multiple arbitrary sequences based on a quadratic assignment problem formulation and an efficient way of approximating its solution. We also derive theoretical links with other permutation distance functions and present their associated combinatorial optimization forms for consensus tasks. The utility of the proposed contributions is demonstrated through the comparison and fusion of multiple seriation algorithms we have implemented, using many real-world datasets from different application domains.

Index Terms—seriation, sequencing, consensus/ensemble seriation, combinatorial data analysis, positional proximity coefficient, quadratic assignment problem.

1 INTRODUCTION

COMBINATORIAL data analysis (CDA) [1], [2] is the broader field of methodologies operating through the arrangement, ordering, grouping or other discrete structural manipulation with the aim of revealing and summarizing data properties and trends. Examples include data clustering, community detection in graphs, prototype reduction, feature subset selection, sample or feature ranking. This work is concerned with an exploratory CDA methodology referred to as *seriation* (or *sequencing*). Its objective is the linear repositioning of a set of data objects, such that the more similar ones are arranged in proximity whereas the more dissimilar appear further apart. There exist many different seriation algorithms to generate such data orderings, and these are based on different assumptions and cost functions. Their input is typically a set of inter-object relational measurements provided by the user in the form of a (dis)similarity matrix.

Seriation is largely cross-disciplinary and it has its roots in the work proposed by Petrie [3] back in 1899 for archaeology. Some examples of situations where seriation has been very useful include the following. In [3] graves were initially characterised by sets of morphological characteristics of artifacts, such as pottery, found inside them. Subsequently, the graves were chronologically ordered to provide what is referred to as sequence dating. In a similar example in paleontology, we can have as the objects to be ordered to be sites of excavation and their features the presence or absence

of certain mammal genera remains. Using site similarity and also auxiliary site age information the sites can then be ordered in a biochronological order [4]. In machine learning seriation can be used to pre-estimate the number of clusters or assess the tendency of the data patterns to form clusters. This is important for several clustering algorithms whose parameterization makes direct or indirect use of the approximate cluster number, or for the user to analyse and interpret the clustering tendency [5]. A related application is the use of seriation for data visualisation. In bioinformatics and for microarray gene expression data, for instance, the gene-by-array matrix can be reordered either along the row or the column to reveal the latent structure of multidimensional data [6]. Exploratory data analysis of biological data using seriation is very important as it can identify biological dynamics, such as successive cell differentiation stages or cell cycles [7]. Text mining is another application example. Given a corpus, datasets represented as word-by-document matrices can be compiled. Applying seriation to such matrices using, for example, cosine similarities between documents, can facilitate multiway spectral cluster assignments [8], or track the flow of conversation in newsgroup postings [4]. In sociology, tabular arrangements of data measuring social relationships, referred to as sociomatrices, can be subjected to row and column rearrangements in order to produce improved and more standardized representations of the sociometric tests [9]. Seriation has also been applied to psychiatric data where patient-by-symptom matrices are available. For example, two correlation matrices, one between patients and one between symptoms, can be constructed and seriated. The resulting orders can provide the means for interaction linkage analysis between the two types of objects [10]. Overall, seriation has been intensely used and often reinvented in numerous fields such

J.Y. Goulermas is with the Department of Computer Science, The University of Liverpool, Ashton Building, Liverpool, L69 3BX, U.K. Correspondence email: j.y.goulermas@liverpool.ac.uk.

A. Kostopoulos and T. Mu are with the Department of Electrical Engineering and Electronics, The University of Liverpool, Brownlow Hill, Liverpool, L69 3GJ, U.K. Correspondence emails: {a.kost,t.mu}@liverpool.ac.uk.

as archaeology, anthropology, biology, bioinformatics, cartography, sociology, database design, document processing, network analysis, psychology, ecology, linguistics, manufacturing, circuit design, as well as data analysis, visualization and machine learning. For a comprehensive overview of its applications and theoretical details, the reader is redirected to the reviewing works of [1], [2], [9], [11]–[15].

In the following, we demonstrate the exploratory value of seriation using the two very simple synthetic 2D datasets of Figs.1(a,b). The different structures in the two sets have been labeled to aid the subsequent discussion. Using the Euclidean distance as a measure of dissimilarity between the points in each dataset, we obtain the corresponding dissimilarity maps. These are plotted in Figs.1(c,d), by linearly mapping the matrix data range to the available colormap. Because the original sample order does not need to obey any structure, these two maps appear completely unstructured and convey no meaningful information. Using seriation the samples can be reordered so that more similar/proximate ones are placed closer. This new order can be used to simultaneously reorder both rows and columns of the unordered maps in Figs.1(c,d) and produce their seriated versions shown in Figs.1(e,f). Although the latter maps contain exactly the same information as the unordered ones, object sequencing reveals many interesting data properties.

For example, all samples within each of the tight clusters A and C in Fig.1(a) correspond to the darker (of higher similarity or shorter distance) diagonal blocks A and C, respectively, in Fig.1(e). These blocks are very discernible as all points within each associated cluster are far closer to each other than to any other point. By contrast, the points composing the long and thin cluster B cannot correspond to a solid block in Fig.1(e), but a rather patterned one. Specifically, this block contains a band of high similarity values along its main diagonal, that reveals the continuum of few proximate samples along the path of the cluster. The radially decreasing off-diagonal similarities within block B represent samples located far from each other as a result of the span and shape of the cluster. The off-diagonal blocks in Fig.1(e) capture key between-structure variations. For example, block D displays the gradual variation from shorter distances (left side of block) between points in cluster A and bottom part of cluster B, to larger distances (right side of block) between A and the top of B and cluster C. Also, block E in Fig.1(e) contains mostly low-varying large distances. This reflects the fact that clusters A and C are compact and far apart from each other relative to their diameters.

Similar observations can be made for the flower dataset. For example, the leaves G, H, I and the petals J in Fig.1(b) can be clearly identified by the four blocks along the main diagonal of the seriated map in Fig.1(f). Unsurprisingly, the almost linear thin stem structure F gives rise to the small block F containing diagonally concentrated short distances. The patterns in the three top-row blocks between F and K in Fig.1(f) reflect the passing of the stem F by each of the leaves G, H and I. The larger distances in block K show that most of the constituent points of stem F lie far from the petals J. More observations can be made for the remaining blocks of both seriated maps.

Although practical applications of seriation involve data of much higher complexity and dimensionality, analysis of

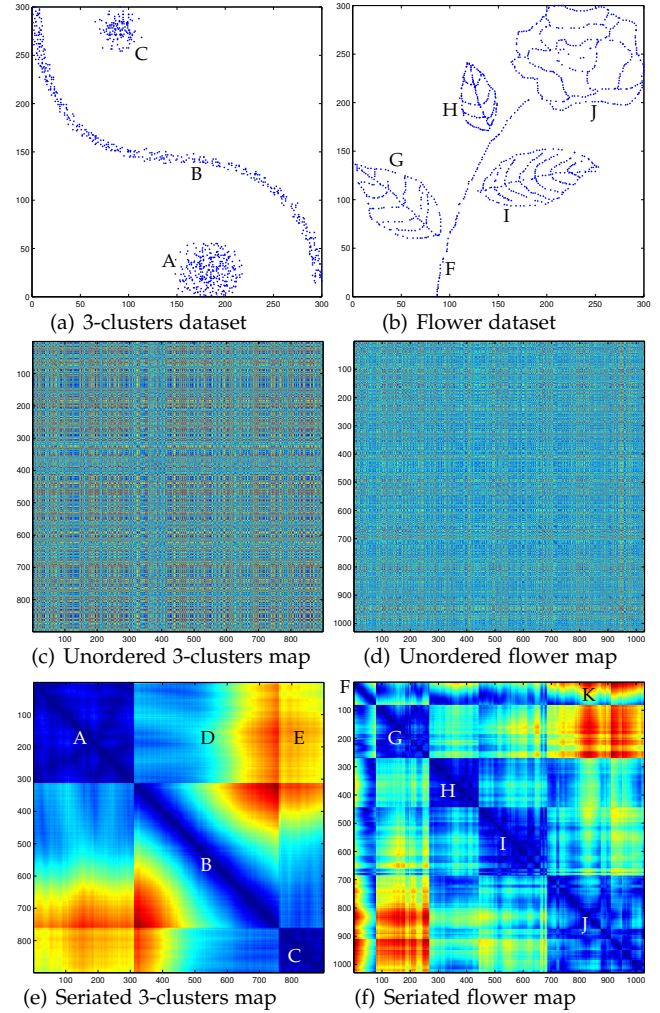


Fig. 1. Data visualization applying seriation to two synthetic datasets. (a,b): Original 2D datasets, with labels A-C and F-J corresponding to structures. (c,d): Symmetric distance matrices using a random initial sample placement. (e,f): Row/column ordered distance maps, with labels A-K corresponding to individual blocks.

the seriated (dis)similarity maps notably aids the understanding of the properties, distributions and tendencies of the data. Many different algorithms to seriate data objects have been proposed (we experiment with a representative set in Section 4.1). These either employ heuristics to seek permutations that maintain proximity of objects which are more similar to each other, or combinatorial optimization procedures that rely on merit or loss functions to locate the exact or approximate optimal, in that sense, ordering. Such functions are typically defined to act on the seriated (dis)similarity maps and evaluate the seriation quality or error by aggregating comparisons between the matrix entries. For a detailed analysis, the reader is referred to [13], [15].

This work introduces novel tools to facilitate both the application and evaluation of seriation results. Specifically, one contribution is a new measure and its associated coefficient to compare arbitrary orderings in terms of how proximately or distantly they arrange the objects. We also present several useful algebraic and statistical properties of this coefficient. Comparing two orderings is very important in order to evaluate the output of different seriation algorithms or when

designing a new one. For example, when for a particular experimental domain, an approximate but fast algorithm is constantly producing results very similar to a slower algorithm that relies on exact combinatorial optimization, the former could be more practical to the user for regular application. The second contribution is a new formulation for fusing multiple arbitrary orderings. This is important for achieving consensus (or ensemble) seriation. For example, when one seriation method cannot be preferred over another one, then multiple ones can be applied and have their outputs combined in a single unifying ordering. We express the underlying model as a quadratic assignment problem and we offer a fast approximate procedure for solving it. The third contribution is that we analyze existing measures for comparing permutations in terms of what type of fusion models they produce and what type of optimization such models require. Finally, we include various experimental case studies to demonstrate the applicability and usefulness of our contributions.

The structure of this paper is as follows. Section 2 revisits some previous methods related to data ordering and consensus analysis. Section 3 contains the main contributions. Specifically, after setting out some basic concepts in Section 3.1, we introduce the new similarity measure in Section 3.2. The coefficient version of the measure and a summary of its key properties are presented in Section 3.3. Section 3.4 contains the proposed consensus seriation approach, while Section 3.5 discusses its relative links with existing measures. Experimentations, comparisons and results are included in Section 4, while Section 5 concludes the work.

2 PREVIOUS WORK

In order to obtain a measure of association between two given arrangements of a collection of objects, we can rely on the comparison of their corresponding permutations. This is because such orderings can be directly represented as permutation vectors. There exist various metrics for comparing permutations, such as the Hamming distance, Spearman footrule, Spearman coefficient, Kendall coefficient, Cayley distance, etc. [16]. However, such measures are mostly applicable to rankings of objects where within each specified arrangement the precedence or priority of some objects over others is of importance. For example, ordering documents subject to search keywords or ordering algorithms according to their performance across multiple datasets. As will be explained in Section 3.2, in order to compare two seriation sequences we need to take into account the relative object positions within the sequences and not their precedence, as the latter does not directly reflect the character and goals of seriation. Existing permutation measures are not necessarily designed to take this into account. Examples include the popular Kendall coefficient that enumerates discordant object pairs between two rankings, the Spearman coefficient based on the sum of squared rank differences, or the Hamming distance that counts matching object locations (see Section 3.5).

With regard to consensus analysis whereby the results of multiple algorithms are combined to a single unifying outcome τ , a lot has been done. One instance of such work

is *consensus ranking* [17]–[21]. The procedure relies on the minimization of a cost typically defined as

$$\sum_{k=1}^b w_k d(\tau, \pi_k), \quad (1)$$

where a set of b permutations or relations π_k with $k = 1, \dots, b$, possibly weighted by some positive w_k , are given. The function $d(\cdot, \cdot)$ is used to measure the distance between two given permutations or relations. As it is to be expected, the specific choice of this function defines the characteristics of the resulting outcome τ as well as the possible options for optimizing Eq.(1). A popular way of solving this problem is to represent each π_k as a binary asymmetric relation \mathbf{R}^k , where \mathbf{R}_{ij}^k is equal to 1 when object i is given precedence over object j (disregarding ties). Then, the consensus or median relation \mathbf{R} can be recovered by solving

$$\begin{aligned} \min_{\mathbf{R}_{ij} \in \{0,1\}:} & \sum_{1 \leq i < j \leq n} \mathbf{Q}_{ij} \mathbf{R}_{ij} \\ \text{s.t.} & \quad 0 \leq \mathbf{R}_{ij} + \mathbf{R}_{jk} - \mathbf{R}_{ik} \leq 1, \quad \forall 1 \leq i < j < k \leq n, \end{aligned} \quad (2)$$

where $\mathbf{Q}_{ij} = \sum_{k=1}^b (2w_k \mathbf{R}_{ij}^k - 1)$. This is an NP-hard, computationally prohibitive for large dimensionalities problem, but various procedures have been proposed to tackle it [18]–[20]. Although, in this work we target the finding of a consensus permutation τ based on a formulation similar to Eq.(1), as we are not concerned with object precedence, we do not use existing permutation metrics. Instead, we propose a new measure $d(\cdot, \cdot)$ to fit the current problem and we also show how the resulting consensus formulation can lend itself to an efficient approximation (see Section 3.4).

Another example of consensus analysis is that of *consensus* (or *ensemble*) *clustering* [22], [23], where the objective is to fuse a set of data partitions resulting from different clustering scenarios. A typical formulation of this problem for finding its median partition τ , is expressed as in Eq.(1), but with each π_k representing the dataset partition generated by the k th clustering algorithm. Again, various functions $d(\cdot, \cdot)$ have been proposed to measure the distance between two partitions (such as the Mirkin distance or the Rand index) and various procedures to solve the consequent optimization problem [23]. Of relevance to our work is the rationale for why a consensus solution would be of any value or need. One reason is that when different algorithms produce different solutions and there is no reason to prefer any particular one over the others, then a solution that combines them all may be the most reasonable to adopt. Similar to consensus clustering, such a solution could exhibit better average performance across the datasets and reduce sensitivity to noise, outliers and sampling variations [22]. Additionally, when a single algorithm is executed multiple times and produces a different outcome in each run (because it is non-deterministic and sensitive to initial conditions, or different experimental conditions have been repeated, e.g., in gene expression analysis) then it often becomes necessary to combine its multiple outcomes.

In terms of *consensus seriation* very little has been done so far. A recent method [6] proposes to combine two specific seriation algorithms by taking advantage of their operating characteristics. An ordering sequence based on hierarchical

clustering is initially obtained and then the nodes of the associated dendrogram are suitably flipped to bias the order towards the one generated by a second algorithm. In this way, the fused sequence combines the benefits from both algorithms, such as the local object behavior from the clustering and the global pattern grouping from the latter method. Another type of consensus seriation [4] relies on the adjustment of a specific seriation algorithm to accommodate an external permutation vector. This vector could either be the output of another algorithm or some ordering based on user domain knowledge. The method works by modifying the Laplacian matrix of a spectral seriation method to bias it towards the external order. Although both of the above approaches are achieving in a sense consensus seriation, they operate upon very specific seriation algorithms and they cannot be generalized to fuse multiple arbitrary sequences. In a recent major seriation review [15], consensus seriation is stated to be a natural extension of the current status, when sequences from multiple algorithms are used or when varying control parameters lead to more robust solutions. Although it is advocated that this can be achieved through formulations similar to Eq.(2), as previously explained such solutions may not be appropriate because of the underlying distance function $d(\cdot, \cdot)$ relying on object precedences.

3 PROPOSED METHODOLOGY

3.1 Preliminaries

A permutation π of n objects is an one-to-one mapping from $\{1, 2, \dots, n\}$ onto itself. Frequently, a permutation π is written as the vector listing the objects (or indices) in their permuted order, that is object $\pi(i)$ is assigned the position i , or conversely, $\pi^{-1}(i)$ is the position of object i (although the reverse convention can be used where $\pi(i)$ denotes the rank of the i th object). There are a total of $n!$ distinct permutations whose container set is denoted by S_n .

For every permutation $\pi \in S_n$ we can directly define an $n \times n$ permutation matrix \mathbf{P}_π as

$$[\mathbf{P}_\pi]_{ij} = \begin{cases} 1 & \text{if } \pi(i) = j, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Multiplying any arbitrary conformable matrix \mathbf{X} from the left (right) with a permutation matrix \mathbf{P}_π , will permute accordingly the rows (columns) of \mathbf{X} . The operation $\mathbf{P}_\pi \mathbf{X} \mathbf{P}_\pi^T$ will permute both rows and columns simultaneously according to the order dictated by π .

Furthermore, two permutations π and τ can be combined to produce a new permutation. For example, given a vector $\mathbf{x} \in \mathcal{R}^n$, the sequence $\mathbf{P}_\tau \mathbf{P}_\pi \mathbf{x}$, applies \mathbf{P}_π first to permute \mathbf{x} and then \mathbf{P}_τ . This is equivalent to applying $\pi \circ \tau$ to \mathbf{x} (where $\pi \circ \tau$ is defined as the standard function composition that maps i to $\pi(\tau(i))$, and is often denoted as the product $\pi\tau$). This is because $[\mathbf{P}_\tau \mathbf{P}_\pi \mathbf{x}]_i = [\mathbf{P}_\pi \mathbf{x}]_{\tau(i)} = \mathbf{x}_{\pi(\tau(i))} = \mathbf{x}_{(\pi\tau)(i)} = [\mathbf{P}_{\pi\tau} \mathbf{x}]_i$.

Of relevance to this work, are two popular combinatorial problems whose solutions rely on finding permutations that optimize some performance index. One is the linear assignment problem (LAP) [24], [25] with the objective of finding an optimal allocation of n objects (e.g., facilities) to n different objects (e.g., locations), given costs \mathbf{X}_{ij} of placing

the i th facility to the j th location. Two possible expressions of the overall minimizing cost are given by

$$\text{LAP}(\mathbf{X}) \stackrel{\text{def}}{=} \sum_{i=1}^n \mathbf{X}_{i, \pi(i)} = \text{tr} \left[\mathbf{X} \mathbf{P}_\pi^T \right], \quad (4)$$

where the sought permutation π assigns the i th facility to the $\pi(i)$ th location. The optimal solution of LAP can be found using the Hungarian method of complexity $O(n^3)$.

A related but NP-hard variation is the quadratic assignment problem (QAP) [25]–[29]. Its objective is also to minimize the overall cost associated with the one-to-one assignment π between two different types of n objects (e.g., facilities and locations), but it is based on two $n \times n$ coefficient matrices. For example, \mathbf{X}_{ij} can correspond to the distance between locations i and j and \mathbf{Y}_{ij} to the flow between facilities i and j . Two of the many possible expressions of the QAP are given by

$$\text{QAP}(\mathbf{X}, \mathbf{Y}) \stackrel{\text{def}}{=} \sum_{i,j=1}^n \mathbf{X}_{\pi(i), \pi(j)} \mathbf{Y}_{ij} = \text{tr} \left[\mathbf{P}_\pi \mathbf{X} \mathbf{P}_\pi^T \mathbf{Y}^T \right]. \quad (5)$$

3.2 Positional Proximity Measure

We firstly introduce here a *positional proximity measure* $d_P : S_n \times S_n \rightarrow \mathcal{R}^+$ that calculates the similarity between any two given permutations $\tau, \pi \in S_n$. We define this as

$$d_P(\tau, \pi) \stackrel{\text{def}}{=} \sum_{j=2}^n \sum_{i=1}^{j-1} (\tau(i) - \tau(j))^2 \cdot (\pi(i) - \pi(j))^2. \quad (6)$$

This measure aggregates information from all individual pairs (i, j) with $1 \leq i < j \leq n$ within τ and π , on how proximate the positions of objects i and j are in relation to each other¹. The individual positional distances in Eq.(6) are squared, so that the actual orders in which the two objects occur are ignored². Therefore, if the relative distance between $\tau(i)$ and $\tau(j)$ is the same as the one between $\pi(i)$ and $\pi(j)$, then the contribution of the pair (i, j) will be the same regardless of where i and j are placed in each permutation. As an example, we can consider the extreme case where $\tilde{\pi}$ is a flipped version of π , i.e., $\tilde{\pi}(i) = \pi(n - i + 1)$. Using the above measure, we have $d_P(\tau, \pi) = d_P(\tau, \tilde{\pi})$ for any τ .

This precedence invariance is not the case, however, for other similarity measures or distance metrics between permutations [16]. The reason is that they take into account the occurrence order of elements, as they conceptually assume the permutations to represent rankings or priorities between the n objects. This is likely to be the case for many applications, and we discuss such measures as well as their links to the proposed measure later in Section 3.5. The precedence

1. Strictly speaking, following the convention of Section 3.1, the two squared distances should be written as $(\tau^{-1}(i) - \tau^{-1}(j))^2$ and $(\pi^{-1}(i) - \pi^{-1}(j))^2$, since $\tau^{-1}(i)$ and $\pi^{-1}(i)$ denote the positions of the object i . For example, let us consider $\tau = [2, 5, 4, 1, 3]^T$ to represent a listing of permuted objects. Then, we have $\tau^{-1} = [4, 1, 5, 3, 2]^T$ and so the positional distance within τ between objects $i = 2$ and $j = 1$ equals $\tau^{-1}(2) - \tau^{-1}(1) = 1 - 4 = -3$. Henceforth, for notational simplicity we drop the explicit use of mapping inversion and assume it implicit unless otherwise stated.

2. The measure could also be alternatively defined as $d_P(\tau, \pi) = \sum_{1 \leq i < j \leq n} |\tau(i) - \tau(j)| \cdot |\pi(i) - \pi(j)|$. However, the former formulation allows a direct approximation of the consensus seriation model presented in Section 3.4.

invariance of the proposed d_P is much more suitable to the underlying character of seriation that requires more similar objects to be positioned closer together, without paying attention to their actual precedence.

It can be seen from Eq.(6), that because d_P is a sum of products of terms (with these terms being the squared distances from each of the τ and π), for each pair (i, j) , the more similar in magnitude the two terms are, the higher the summation becomes. This can be justified more formally by defining the $\binom{n}{2}$ -length vector $\omega_\tau = [(\tau(1) - \tau(2))^2, (\tau(1) - \tau(3))^2, (\tau(1) - \tau(4))^2, \dots, (\tau(i) - \tau(j))^2, \dots, (\tau(n-1) - \tau(n))^2]^T$ to contain in some predetermined order all squared distances enumerated by the summation in Eq.(6). Then, for a similarly defined accompanying ω_π , d_P can be simply expressed as the inner product $\omega_\tau^T \omega_\pi$. Based on that, we can employ a rearrangement theorem from [30] which states that given any two vectors $\mathbf{x}, \mathbf{y} \in \mathcal{R}^n$ and their corresponding versions \mathbf{x}^+ and \mathbf{y}^+ sorted in ascending order, we have

$$\sum_{i=1}^n \mathbf{x}_i^+ \mathbf{y}_{n-i+1}^+ \leq \sum_{i=1}^n \mathbf{x}_i \mathbf{y}_i \leq \sum_{i=1}^n \mathbf{x}_i^+ \mathbf{y}_i^+. \quad (7)$$

From the rightmost inequality, we can see that $\omega_\tau^T \omega_\pi$ obtains its maximum when the vectors ω_τ and ω_π are ordered similarly, which occurs when τ and π have identical positional distance terms $(\tau(i) - \tau(j))^2$ and $(\pi(i) - \pi(j))^2$. The more similar the two terms are on average for each (i, j) , the higher $\omega_\tau^T \omega_\pi$ and accordingly d_P become. Note, that for any $\tau \in S_n$, its corresponding ω_τ is always composed of exactly the same elements (e.g., $n-1$ ones, $n-2$ fours, $n-3$ nines, \dots , one occurrence of $(n-1)^2$) but these occur in different restricted order within ω_τ for each different τ .

A desirable property when working with permutations is the right-invariance one which is shared by most measures [16]. This property requires the comparison between two permutations to be independent to any arbitrary relabeling of the n objects. This is also a reasonable expectation for the object sequencing applications we consider here. The proposed d_P is right-invariant because

$$d_P(\tau, \pi) = d_P(\tau\sigma, \pi\sigma), \forall \tau, \pi, \sigma \in S_n, \quad (8)$$

as a result of Eq.(6) not changing when σ simultaneously permutes the ranks in τ and π to realize the object relabeling. Because of this, we can simplify the similarity estimation as $d_P(\tau, \pi) = d_P(\tau\pi^{-1}, \mathbf{e})$, where \mathbf{e} denotes the identity permutation defined by $e(i) = i$.

Some further useful observations can be obtained if we expand and then simplify Eq.(6) using some basic properties of permutations. We can finally reach the equivalent expression

$$d_P(\tau, \pi) = n \sum_{i=1}^n \tau(i)^2 \pi(i)^2 + 2 \left(\sum_{i=1}^n \tau(i) \pi(i) \right)^2 - 2k_1 \sum_{i=1}^n (\tau(i)^2 \pi(i) + \tau(i) \pi(i)^2) + k_2^2, \quad (9)$$

which depends on sums of products of components from the two vectors, and involves constants $k_1 = \tau^T \mathbf{1}_n = \frac{n(n+1)}{2}$ and $k_2 = \tau^T \tau = \frac{n(n+1)(2n+1)}{6}$ ($\mathbf{1}_n$ is the column vector containing n ones, and the notation τ is used interchangeably to denote both the underlying mapping and

the column vector corresponding to the permutation). It is noteworthy, that Eq.(9) is equivalent to a nonlinear mixture of the central cross-moments of order two and three between τ and π . An m th order central cross-moment between two stochastic variables X and Y is defined as $\mu_{XY}^{(m_X, m_Y)} = E[(X - E[X])^{m_X} \cdot (Y - E[Y])^{m_Y}]$, for any $m_X, m_Y \in \mathbb{Z}^+$ with $m = m_X + m_Y$. Specifically, we can rewrite Eq.(9) as

$$d_P(\tau, \pi) = n^2 \mu_{\tau\pi}^{(2,2)} + 2n^2 \left(\mu_{\tau\pi}^{(1,1)} \right)^2 + \frac{(n^3 - n)^2}{12^2}. \quad (10)$$

This can be easily proved by re-expressing the covariance as

$$\mu_{\tau\pi}^{(1,1)} = \frac{1}{n} \tau^T \pi - \frac{k_1^2}{n^2}, \quad (11)$$

and one of the three possible cokurtosis measures as

$$\begin{aligned} \mu_{\tau\pi}^{(2,2)} &= \frac{1}{n} \sum_{i=1}^n \tau(i)^2 \pi(i)^2 + \frac{4k_1^2}{n^3} \sum_{i=1}^n \tau(i) \pi(i) \\ &\quad - \frac{2k_1}{n^2} \sum_{i=1}^n (\tau(i)^2 \pi(i) + \tau(i) \pi(i)^2) - \frac{3k_1^4}{n^4} + \frac{2k_1^2 k_2}{n^3}. \end{aligned} \quad (12)$$

The simplifications in Eqs.(11,12) rely on the fact that all permutations have the same mean $\frac{k_1}{n}$ and the same sum of squares k_2 . From the latest formulation of the measure in Eq.(10), it can be seen that d_P increases when the squared covariance increases. This corresponds to both permutations τ and π ordering the objects with their positions correlated or anti-correlated. Additionally, d_P increases when the cokurtosis increases. In general, this statistic measures the degree of peakedness of one variable with respect to the other, but in the current context the increase corresponds to how the squared distances from the mean for each individual variable match those in the other variable. In either case, the actual precedence between objects is ignored so that, as previously mentioned, even a mirrored version $\tilde{\pi}$ yields squared covariance, cokurtosis and consequently d_P values equal to those as when comparing π with a reference τ . Other measures, such as Spearman correlation (discussed later in Section 3.5) can also be expressed directly using the covariance, but it is sensitive to the actual precedence between objects.

3.3 Positional Proximity Coefficient

We now define two matrix functions $\mathbf{A}, \mathbf{B} : S_n \rightarrow \mathcal{R}^{n \times n}$ as

$$\mathbf{A}(\tau) = \mathbf{B}(\tau) \odot \mathbf{B}(\tau), \quad (13)$$

$$\mathbf{B}(\tau) = \tau \mathbf{1}_n^T - \mathbf{1}_n \tau^T, \quad (14)$$

where the symbol \odot denotes the Hadamard matrix product. These two compact notations provide us with matrix elements of the form $[\mathbf{A}(\tau)]_{ij} = (\tau(i) - \tau(j))^2$ and $[\mathbf{B}(\tau)]_{ij} = \tau(i) - \tau(j)$.

Using the above, we can obtain a very convenient expression for d_P that can be used to establish a more direct link with the combinatorial problems considered here (see Section 3.4) and also with other existing permutation measures (see Section 3.5). Specifically, the positional proximity

measure in Eq.(6) can be equivalently re-expressed using matrix notation as

$$\begin{aligned} d_P(\tau, \pi) &= \frac{1}{2} \sum_{i,j=1}^n [\mathbf{A}(\tau)]_{ij} [\mathbf{A}(\pi)]_{ij} \\ &= \frac{1}{2} \text{tr} [\mathbf{A}(\tau) \mathbf{A}(\pi)^T]. \end{aligned} \quad (15)$$

The latter expression can be used to provide an instance of the generalized correlation coefficient [31], [32]

$$\gamma(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} \frac{\sum_{i,j} \mathbf{X}_{ij} \mathbf{Y}_{ij}}{\sqrt{\sum_{i,j} \mathbf{X}_{ij}^2 \sum_{i,j} \mathbf{Y}_{ij}^2}} \quad (16)$$

between two vectors of observations \mathbf{x} and \mathbf{y} . The elements \mathbf{X}_{ij} and \mathbf{Y}_{ij} of the two auxiliary matrices \mathbf{X} and \mathbf{Y} , are some type of scores associated with the sample pairs $(\mathbf{x}_i, \mathbf{x}_j)$ and $(\mathbf{y}_i, \mathbf{y}_j)$, respectively. By setting the auxiliary score matrices in Eq.(16) to be $\mathbf{A}(\tau)$ and $\mathbf{A}(\pi)$, we can define a *positional proximity coefficient* given by

$$\begin{aligned} \gamma_P(\tau, \pi) &= \frac{\text{tr} [\mathbf{A}(\tau) \mathbf{A}(\pi)^T]}{\sqrt{\text{tr} [\mathbf{A}(\tau) \mathbf{A}(\tau)^T] \text{tr} [\mathbf{A}(\pi) \mathbf{A}(\pi)^T]}} \\ &= \frac{\text{tr} [\mathbf{A}(\tau) \mathbf{A}(\pi)^T]}{\text{tr} [\mathbf{A}(\tau) \mathbf{A}(\tau)^T]}. \end{aligned} \quad (17)$$

This is equivalent to normalizing d_P in Eq.(15) with the maximum possible value it can assume. It further simplifies to

$$\gamma_P(\tau, \pi) = \frac{\text{tr} [\mathbf{A}(\tau) \mathbf{A}(\pi)]}{\frac{n^6}{15} - \frac{n^4}{6} + \frac{n^2}{10}}, \quad (18)$$

because from Eqs.(13,14) $\mathbf{A}(\pi)$ is symmetric and also the denominator is independent of the chosen permutation and therefore constant (see Supplementary material, Appendix A). γ_P is positive because of the trace quantities being positive and it cannot exceed one because of the Cauchy-Bunyakovskii-Schwarz inequality. The more similar the positional information two given permutations τ and π contain, the higher the value of γ_P will be.

The proposed coefficient can be used, for example, to measure the degree by which two different sequences agree. These sequences could be the output of two different seriation algorithms or the output from one algorithm executed with two different conditions (e.g., different initial state or randomization, different sample (dis)similarity measures applied on the raw dataset, etc.). Coefficient values could also be combined over multiple datasets (from different or a specific problem domain) to examine the gross agreement between two or more algorithms (Section 4 exemplifies its use in similar setups).

It is of interest to derive some basic statistics of $\gamma_P(\tau, \pi)$ that could contribute to such comparisons. Using the right-invariance property, discussed in Section 3.2, this is equivalent to deriving statistics for a unary reformulation of γ_P , expressed as $\hat{\gamma}_P(\rho) \equiv \gamma_P(\rho, e)$ for a random permutation $\rho \in S_n$ and the identity permutation e . Assuming a uniform distribution imposed over all $n!$ permutations ρ , the mean and variance of $\hat{\gamma}_P$ and thus γ_P can be derived as functions of n . This can be achieved by employing certain formulae

for more general QAP matrices [32], which (as shown in Supplementary material, Appendix B) eventually lead to

$$E[\hat{\gamma}_P] = \frac{5n(n+1)}{6(2n^2-3)}, \quad (19)$$

$$\sigma^2(\hat{\gamma}_P) = \frac{(n-2)(2n^4+37n^3+42n^2-45n-54)}{18n(n-1)(2n^2-3)^2}. \quad (20)$$

The above can be useful for statistically analyzing observations using, for instance, Chebyshev's or Cantelli's inequalities. Also, z-scores $\frac{\hat{\gamma}_P(\rho) - E[\hat{\gamma}_P]}{\sqrt{\sigma^2(\hat{\gamma}_P)}}$ can be used for more useful interpretation of comparisons; for example, when comparing the performance of algorithms using multiple datasets of different sizes in order to make the evaluation independent of n . Also, although the distribution of $\hat{\gamma}_P$ appears to be heavy right-tailed, it has been demonstrated in [32] for more general QAP expressions that it can be well approximated by a normal distribution, and in this case standard normal tables can also be utilized.

The minimum value attainable by $\gamma_P(\tau, \pi)$ corresponds to two sequences τ and π containing object positions in total disagreement. Because of the way the proximity measure is defined in Eq.(6), however, this minimum value of γ_P can never reach zero. We can calculate its minimum by finding the constant permutation ξ (we refer to it as the *bounce* permutation) that minimizes $\hat{\gamma}_P(\xi)$. This permutation, which resembles but is not equal to the zig-zag permutation defined in [25], [26] for a certain class of QAP problems, is given by

$$\xi(i) \stackrel{\text{def}}{=} \begin{cases} m-i+1, & \text{if } i \leq m \text{ and } i \text{ is odd,} \\ m+i, & \text{if } i \leq m \text{ and } i \text{ is even,} \\ i-m, & \text{if } i > m \text{ and } i \text{ is odd,} \\ n+m-i+1, & \text{if } i > m \text{ and } i \text{ is even,} \end{cases} \quad (21)$$

for $i = 1, \dots, n$ with $m = \frac{n}{2}$. We only investigate the case for even n here (see Supplementary material, Appendix D). Eq.(21) can also be expressed more concisely as

$$\xi(i) = m + (-1)^i (m - |i - m| - \text{mod}(i, 2) + [i > m]), \quad (22)$$

where $[\cdot]$ is the Iverson bracket. For example, for $n = 12$ we have $\xi = [6, 8, 4, 10, 2, 12, 1, 11, 3, 9, 5, 7]^T$. This vector relocates all the objects to be maximally apart (in terms of their squared pairwise positional distances evaluated by d_P) from $e = [1, 2, 3, \dots, 12]^T$. For instance, objects 1 and 12 which appear furthest within e are moved to be the closest in ξ ; then objects 2 and 11 from the next largest distance within e are moved on immediately next to 1 and 12, and the pattern carries on by alternating the placement of each subsequent pair.

To find the minimum value of d_P , we need to compute $d_P(\xi, e) = \frac{1}{2} \text{tr} [\mathbf{A}(\xi) \mathbf{A}(e)^T] = \sum_{1 \leq i < j \leq n} (\xi(i) - \xi(j))^2 (i - j)^2$. We can express this value in terms of n (see Supplementary material, Appendix C) as

$$\min_{\tau, \pi \in S_n} d_P(\tau, \pi) = \frac{13n^6 - 20n^4 + 52n^2}{1440} + \frac{\text{mod}(m, 2)}{2}, \quad (23)$$

which is valid for both cases of m being even or odd. We can further normalize this quantity as in Eq.(18), in order to obtain the minimum value of γ_P

$$L_{\gamma_P}(n) \stackrel{\text{def}}{=} \min_{\tau, \pi \in S_n} \gamma_P(\tau, \pi) = \frac{13n^4 - 20n^2 + 52 + \frac{720}{n^2} \bmod(m, 2)}{24(n-1)(n+1)(2n^2-3)}. \quad (24)$$

It can be seen that $L_{\gamma_P} \rightarrow \frac{13}{48} = 0.2708\bar{3}$ as $n \rightarrow \text{inf}$. This value can be useful as an alternative way of facilitating comparisons between sets of sequences with different lengths, by scaling the coefficient as $\frac{\gamma_P(\tau, \pi) - L_{\gamma_P}(n)}{1 - L_{\gamma_P}(n)}$ so that it spreads across the entire $[0, 1]$ range.

A final observation is that the proposed proximity coefficient can be used to derive a metric distance. The properties of non-negativity, definiteness and reflexivity, as well as symmetry can be satisfied with the simple transformation $1 - \gamma_P(\tau, \pi)$. The triangle inequality can also be satisfied using transformations based on concave metric preserving functions [33].

3.4 Consensus Seriation and Sequence Fusion

We now turn our attention to achieving consensus seriation using the proposed positional proximity measure. That is, as discussed in Section 2, combining multiple sequences by fusing their constituent positional information, so that objects that frequently appear closer in different sequences are more likely to appear closer in the unifying sequence. Given a set of b positive weights w_k which sum to unity and b associated permutations $\pi_k \in S_n$, with $k = 1, \dots, b$, the objective is to find the optimal permutation

$$\tau^* = \operatorname{argmax}_{\tau \in S_n} \sum_{k=1}^b w_k d_P(\tau, \pi_k). \quad (25)$$

The weights w_k are optionally used to alter the importance of the different permutations. The above optimization employs a weighted summation similar to Eq.(1) for consensus ranking and consensus clustering models, but it is based on the proposed positional similarity measure d_P of Eq.(6) to maximize the average pairwise object similarities between the sought τ^* and each π_k . In the following, we will elaborate on the formulation of Eq.(25), establish links with combinatorial optimization problems and suggest an efficient approximation to solve it.

Using Eq.(15), the above maximand can be written (adjusting for constant terms) as

$$\begin{aligned} 2 \sum_{k=1}^b w_k d_P(\tau, \pi_k) &= \sum_{k=1}^b w_k \operatorname{tr} [\mathbf{A}(\tau) \mathbf{A}(\pi_k)^T] \\ &= \operatorname{tr} \left[\mathbf{A}(\tau) \sum_{k=1}^b w_k \mathbf{A}(\pi_k)^T \right] \\ &= \operatorname{tr} [\mathbf{A}(\tau) \mathbf{K}^T]. \end{aligned} \quad (26)$$

The symmetric *positional distance matrix* $\mathbf{K} = \sum_{k=1}^b w_k \mathbf{A}(\pi_k)$ contains the combined positional information for the problem at hand. Each ij th element

$$\mathbf{K}_{ij} = \sum_{k=1}^b w_k (\pi_k(i) - \pi_k(j))^2 \quad (27)$$

corresponds to the positional distances between objects i and j , weighted and aggregated over all permutations π_k . The intuitive character of the optimization can be easily observed if we express Eq.(26) as $\sum_{i,j=1}^n (\tau(i) - \tau(j))^2 \mathbf{K}_{ij}$. Each positional distance term \mathbf{K}_{ij} can be thought of as a reference weight. The larger this weight is, the stronger the maximization for that ij th term will be, and therefore the further apart the positions $\tau(i)$ and $\tau(j)$ are pushed.

The optimization problem of Eq.(25) is in fact a QAP one (see Section 3.1). We can show this, by observing that the maximand can be rewritten from Eq.(26) as

$$\operatorname{tr} [\mathbf{A}(\tau) \mathbf{K}^T] = \operatorname{tr} [\mathbf{A}(\mathbf{P}_\tau \mathbf{e}) \mathbf{K}^T] = \operatorname{tr} [\mathbf{P}_\tau \mathbf{A}(\mathbf{e}) \mathbf{P}_\tau^T \mathbf{K}^T], \quad (28)$$

since for any τ and its corresponding permutation matrix \mathbf{P}_τ , we have $[\mathbf{P}_\tau \mathbf{A}(\mathbf{e}) \mathbf{P}_\tau^T]_{ij} = (\tau(i) - \tau(j))^2 = [\mathbf{A}(\tau)]_{ij}$. Eq.(28) matches the definition of Eq.(5) and it therefore corresponds to QAP $(\mathbf{A}(\mathbf{e}), -\mathbf{K})$. The two defining QAP coefficient matrices are the distance $\mathbf{A}(\mathbf{e})$ (a constant symmetric Toeplitz matrix with ij th elements of the form $(i - j)^2$) and the flow matrix $-\mathbf{K}$ (negated to indicate maximization).

The maximizing quantity of Eq.(26) can also be expressed more conveniently as follows

$$\sum_{i,j=1}^n (\tau(i) - \tau(j))^2 \mathbf{K}_{ij} \quad (29)$$

$$\begin{aligned} &= 2 \sum_{i=1}^n \tau(i)^2 \sum_{j=1}^n \mathbf{K}_{ij} - 2 \sum_{i,j=1}^n \tau(i) \tau(j) \mathbf{K}_{ij} \\ &= 2 \sum_{i=1}^n \tau(i)^2 \mathbf{K}_{i\bullet} \mathbf{1}_n - 2 \tau^T \mathbf{K} \tau = 2 \tau^T \mathbf{L} \tau, \end{aligned} \quad (30)$$

where $\mathbf{K}_{i\bullet}$ is the i th row vector of \mathbf{K} . The last expression makes use of a symmetric Laplacian matrix defined as $\mathbf{L} = \operatorname{diag}(\mathbf{K} \mathbf{1}_n) - \mathbf{K}$, where $\operatorname{diag}(\mathbf{K} \mathbf{1}_n)$ is the diagonal matrix whose i th diagonal element is formed by the sum of $\mathbf{K}_{i\bullet}$. The expression in Eq.(30), for continuous τ and similarities \mathbf{K}_{ij} , rather than distances, between vertices/objects i and j , is similar to the quadratic quantity commonly seen in spectral embedding methods [34], [35].

It can be easily seen that the quadratic form in Eq.(30) can give rise to an alternative but equivalent QAP formulation because

$$\tau^T \mathbf{L} \tau = (\mathbf{P}_\tau \mathbf{e})^T \mathbf{L} \mathbf{P}_\tau \mathbf{e} = \operatorname{tr} [\mathbf{P}_\tau \mathbf{e} \mathbf{e}^T \mathbf{P}_\tau^T \mathbf{L}]. \quad (31)$$

The expression in Eq.(31) also matches the definition of Eq.(5) and corresponds to QAP $(\mathbf{e} \mathbf{e}^T, -\mathbf{L})$, with $\mathbf{e} \mathbf{e}^T$ being the distance matrix and $-\mathbf{L}$ the flow. A similar QAP formulation has also been used in [36] for a different problem, the 2-sum one, to model matrix envelope reduction.

Any QAP solver [27]–[29] can be used to optimize either of the two QAP formulations. For example, one could consider an exact solver based on dynamic programming, cutting planes or branch and bound methods. However, for the sequencing problems considered here with dimensionalities of at least hundreds of samples, such an approach would be impractical. An alternative would be to use a heuristic method, such as local search, simulated annealing or evolutionary optimization. However, such methods can also be slow due to the large number of function evaluations they require and can be sensitive to different parameterizations.

In this work, we approximate the maximization of $\sum_{k=1}^b w_k d_P(\tau, \pi_k) = \tau^T \mathbf{L} \tau$ with $\tau \in S_n$, using for simplicity a spectral seriation approach [4], [8], [37], [38]. This relies on the relaxation of τ to a continuous vector accompanied by scale constraints, which according to the Rayleigh-Ritz theorem [39], finds the optimal solution by simply eigen-decomposing the Laplacian. As has been shown in [38], spectral ordering can provide the means for optimally recovering the ordering of a symmetric similarity matrix, if there actually exists a permutation that can bring this matrix into a Robinson form (that is, a matrix with entries non-increasing while moving away from the diagonal). This substantiates the use of the method despite its approximate character owed to the relaxation. The solving permutation in that case is found by examining the order of the elements of the Fiedler eigenvector (the one corresponding to the second smallest eigenvalue). Although in most practical cases a similarity matrix contains measurement noise and cannot be brought into a perfect Robinson form, as demonstrated in the aforementioned works, the solving eigenvector generates an order that keeps strongly similar objects mostly together.

In our case, since \mathbf{K} contains dissimilarities and we maximize $\tau^T \mathbf{L} \tau$, we are only interested in the dominant eigenvector \mathbf{v} of \mathbf{L} . Adapting the method of [38], if the b permutations π_k are in full agreement, then matrix \mathbf{K} can be reordered into an anti-Robinson form (that is, one with non-decreasing entries while moving away from the diagonal). Although this is an ideal situation, in practical cases where the π_k likely have discrepant information, object pairs i and j with large average positional distance \mathbf{K}_{ij} will be mostly positioned further apart. Here, we simply take the solution τ^* to be the permutation that renders $\mathbf{P}_{\tau^*} \mathbf{v}$ monotonically ordered in either direction. This procedure is very fast and produces approximate but practical solutions; other spectral variations from [4], [8], [38] can also be adapted. The cost of the optimization does not depend on the number b of available orderings to fuse, as the matrix \mathbf{K} summarizes all given π_k (the inverse π_k^{-1} are actually used in Eq.(27) to get the object positions¹) and their weights w_k . The cost of eigen-decomposition is in general of $O(n^3)$, but since only the dominant eigenvector of the Laplacian \mathbf{L} is needed, a simple method such as the power iteration with converges linearly with a cost of $O(n^2)$ operations per step, or the Rayleigh quotient iteration with converges cubically but with $O(n^3)$, or a more efficient Krylov subspace method can be used [40].

3.5 Links To Other Methods

In this section, we present some alternative ways of optimizing the quantity $\sum_{k=1}^b w_k d(\tau, \pi_k)$ from Eqs.(1,25) using different existing measures $d(\cdot, \cdot)$ between permutations [16]. Although, as previously explained such measures are not very suitable for seriation, here we focus on the type of combinatorial problems their consensus formulations generate for comparative purposes.

3.5.1 Hamming measure

The Hamming distance, which is commonly used in information theory and communications, is perhaps the simplest

way for comparing two permutations. It is defined as

$$d_H(\tau, \pi) = \sum_{i=1}^n [\tau(i) \neq \pi(i)]. \quad (32)$$

It is easy to see the relationship between the Hamming distance of two permutations and the Frobenius norm of their corresponding permutation matrices. Specifically, using the definition in Eq.(3), we obtain

$$\|\mathbf{P}_\tau - \mathbf{P}_\pi\|_F^2 = \sum_{i,j=1}^n ([\mathbf{P}_\tau]_{ij} - [\mathbf{P}_\pi]_{ij})^2 = 2d_H(\tau, \pi). \quad (33)$$

Based on this and the fact that $\text{tr}[\mathbf{P}_\tau \mathbf{P}_\tau^T] = n$ for any $\tau \in S_n$, we can rewrite the relevant minimand as

$$2 \sum_{k=1}^b w_k d_H(\tau, \pi_k) = \sum_{k=1}^b w_k \|\mathbf{P}_\tau - \mathbf{P}_{\pi_k}\|_F^2 \quad (34)$$

$$\begin{aligned} &= \sum_{k=1}^b w_k (2n - 2\text{tr}[\mathbf{P}_\tau \mathbf{P}_{\pi_k}^T]) \\ &= 2n - 2\text{tr}[\mathbf{P}_\tau \mathbf{S}^T]. \end{aligned} \quad (35)$$

In the above, the real valued matrix $\mathbf{S} = \sum_{k=1}^b w_k \mathbf{P}_{\pi_k}$ is the weighted sum of all permutations matrices corresponding to the b given π_k . Therefore, minimizing the original quantity in Eq.(34), is equivalent to maximizing $\text{tr}[\mathbf{P}_\tau \mathbf{S}^T]$ in Eq.(35). This latter syntax can be seen to match the LAP definition in Eq.(4) as it corresponds to the form LAP $(-\mathbf{S})$, with $-\mathbf{S}$ being the cost matrix of the assignment.

It is interesting to additionally observe that since \mathbf{S} is a convex combination of permutation matrices, it is a doubly stochastic matrix and according to Birkhoff's theorem [25], it belongs to the convex polytope in \mathcal{R}^{n^2} defined by vertices corresponding to all $n \times n$ permutation matrices. If we further express the trace quantity in Eq.(35) as

$$\text{tr}[\mathbf{P}_\tau \mathbf{S}^T] = \frac{1}{2} (n + \text{tr}[\mathbf{S} \mathbf{S}^T] - \|\mathbf{P}_\tau - \mathbf{S}\|_F^2), \quad (36)$$

we can see that the above problem is also equivalent to minimizing $\|\mathbf{P}_\tau - \mathbf{S}\|_F^2$. This shows that a consensus ranking search based on d_H , is equivalent to searching for that vertex \mathbf{P}_τ that is closest (in the Frobenius norm sense) to the interior polytope point \mathbf{S} , formed by the convex combination of the b selected vertices \mathbf{P}_{π_k} .

3.5.2 Spearman measure

The unnormalized Spearman distance between two permutations is defined [16] as

$$d_S(\tau, \pi) = \|\tau - \pi\|_2^2. \quad (37)$$

The commonly used Spearman correlation coefficient is a normalized version of the above, designed to range within $[-1, +1]$. It is given by $\gamma_S(\tau, \pi) = 1 - \frac{d_S(\tau, \pi)}{k_3}$ using the constant $k_3 = \frac{n(n^2-1)}{6}$.

It is noteworthy that, as it has been observed in [31], [32], the Spearman coefficient can be expressed as a special case of the generalized correlation coefficient. This is possible through the use of Eq.(16) while setting the scores \mathbf{X}_{ij} and \mathbf{Y}_{ij} to be $\tau(i) - \tau(j)$ and $\pi(i) - \pi(j)$, respectively.

TABLE 1

Summary of the main formulations of the raw and the coefficient forms of the proposed permutation measure and other existing ones. The last column indicates the underlying combinatorial problem for the consensus task.

Measure	Unnormalized $d(\tau, \pi)$	Coefficient $\gamma(\tau, \pi)$	Assignment problem
Hamming	$d_H = \sum_{i=1}^n [\tau(i) \neq \pi(i)]$	$\gamma_H = \frac{\text{tr}[\mathbf{P}_\tau \mathbf{P}_\pi^T]}{n}$	LAP $(-\mathbf{S})$
Spearman	$d_S = \ \tau - \pi\ _2^2$	$\gamma_S = \frac{\text{tr}[\mathbf{B}(\tau)\mathbf{B}(\pi)^T]}{\frac{n^2(n^2-1)}{6}}$	LAP $(-\mathbf{se}^T)$, or LAP $(-\mathbf{See}^T)$
Kendall	$d_K = \sum_{i,j=1}^n [\tau(i) < \tau(j) \wedge \pi(i) > \pi(j)]$	$\gamma_K = \frac{\text{tr}[\mathbf{C}(\tau)\mathbf{C}(\pi)^T]}{n(n-1)}$	QAP $(\mathbf{C}(e), -\mathbf{N})$
Positional proximity	$d_P = \sum_{1 \leq i < j \leq n} (\tau(i) - \tau(j))^2 (\pi(i) - \pi(j))^2$	$\gamma_P = \frac{\text{tr}[\mathbf{A}(\tau)\mathbf{A}(\pi)^T]}{\frac{n^6}{15} - \frac{n^4}{6} + \frac{n^2}{10}}$	QAP $(\mathbf{A}(e), -\mathbf{K})$, or QAP $(\mathbf{ee}^T, -\mathbf{L})$
where: $\mathbf{A}(\tau) = \mathbf{B}(\tau) \odot \mathbf{B}(\tau)$ $\mathbf{B}(\tau) = \tau \mathbf{1}_n^T - \mathbf{1}_n \tau^T$ $\mathbf{C}(\tau) = \text{sign}(\mathbf{B}(\tau))$		$\mathbf{S} = \sum_{k=1}^b w_k \mathbf{P}_{\pi_k}$ $\mathbf{s} = \sum_{k=1}^b w_k \pi_k$ $\mathbf{e} = [1, 2, \dots, n]^T$	$\mathbf{K} = \sum_{k=1}^b w_k \mathbf{A}(\pi_k)$ $\mathbf{L} = \text{diag}(\mathbf{K} \mathbf{1}_n) - \mathbf{K}$ $\mathbf{N} = \sum_{k=1}^b w_k \mathbf{C}(\pi_k)$

Further, using Eq.(14) leads to the coefficient rewritten more conveniently as

$$\gamma_S(\tau, \pi) = \frac{\text{tr}[\mathbf{B}(\tau)\mathbf{B}(\pi)^T]}{\text{tr}[\mathbf{B}(\tau)\mathbf{B}(\tau)^T]} = \frac{\text{tr}[\mathbf{B}(\tau)\mathbf{B}(\pi)^T]}{nk_3}, \quad (38)$$

since $\text{tr}[\mathbf{B}(\tau)\mathbf{B}(\pi)^T] = \sum_{i,j=1}^n (\tau(i) - \tau(j))(\pi(i) - \pi(j))$. Note, that the measure d_S and the numerator above are affine versions of each other, since $\text{tr}[\mathbf{B}(\tau)\mathbf{B}(\pi)^T] + nd_S(\tau, \pi) = nk_3$.

Contrasting γ_S in Eq.(38) to the proposed positional proximity coefficient γ_P in Eqs.(17, 18), shows that they both combine aggregate scores over all possible object pairs, but differ in the way these individual scores are calculated. Moreover, using the expression of the covariance $\mu_{\tau\pi}^{(1,1)}$ from Eq.(11), we can rewrite Eq.(37) as

$$d_S(\tau, \pi) = k_3 - 2n\mu_{\tau\pi}^{(1,1)}. \quad (39)$$

This can be compared with the more complex expression of d_P in Eq.(10) that involves squared covariance and cokurtosis that enable it to ignore object precedences.

We focus now on the combination of multiple permutations using the Spearman measure. Again, based on Eqs.(1,25), this leads to minimizing $\sum_{k=1}^b w_k d_S(\tau, \pi_k)$. Substituting with Eq.(37) we have

$$\begin{aligned} \sum_{k=1}^b w_k \|\tau - \pi_k\|_2^2 &= \sum_{k=1}^b w_k (\tau^T \tau + \pi_k^T \pi_k - 2\tau^T \pi_k) \\ &= 2k_2 - 2 \sum_{k=1}^b w_k \tau^T \pi_k \\ &= 2k_2 - 2\tau^T \mathbf{s}, \end{aligned} \quad (40)$$

where $\mathbf{s} = \sum_{k=1}^b w_k \pi_k$ is the convex combination of all permutations and k_2 the constant defined earlier in Section 3.2. Thus, the optimization is equivalent to maximizing the inner product $\tau^T \mathbf{s}$. However, this can be rewritten as

$$\tau^T \mathbf{s} = (\mathbf{P}_\tau \mathbf{e})^T \mathbf{s} = \mathbf{e}^T \mathbf{P}_\tau^T \mathbf{s} = \text{tr}[\mathbf{se}^T \mathbf{P}_\tau^T], \quad (41)$$

which matches the problem LAP $(-\mathbf{se}^T)$, with $-\mathbf{se}^T$ being the underlying cost matrix. Note, that from the definition of \mathbf{S} in Section 3.5.1, we have $\mathbf{S}\mathbf{e} = \sum_{k=1}^b w_k \pi_k = \mathbf{s}$, which shows that the LAP cost matrix can be equivalently considered to be $-\mathbf{See}^T$. In this particular case, however, the consensus ranking solution based on the Spearman measure can be obtained without solving the assignment problem, as by relying on Eq.(7), maximizing $\tau^T \mathbf{s}$ is achieved by simply finding a permutation τ^* that orders \mathbf{s} monotonically.

3.5.3 Kendall measure

The unnormalized Kendall distance between two permutations is defined [16] as

$$d_K(\tau, \pi) = \sum_{i,j=1}^n [\tau(i) < \tau(j) \wedge \pi(i) > \pi(j)], \quad (42)$$

which counts the number of discordant object pairs (i, j) between τ and π . The commonly used Kendall correlation coefficient is a normalized version of the above and is given by $\gamma_K(\tau, \pi) = 1 - \frac{d_K(\tau, \pi)}{k_4} \in [-1, +1]$, using the constant $k_4 = \frac{n(n-1)}{4}$.

As with Spearman, the Kendall coefficient can be deemed to be a special case of the generalized correlation coefficient [31], [32]. This is possible by setting the scores \mathbf{X}_{ij} and \mathbf{Y}_{ij} in Eq.(16) as $\text{sign}(\tau(i) - \tau(j))$ and $\text{sign}(\pi(i) - \pi(j))$, respectively. Based on this, we can define

$$\mathbf{C}(\tau) = \text{sign}(\tau \mathbf{1}_n^T - \mathbf{1}_n \tau^T) = \text{sign}(\mathbf{B}(\tau)), \quad (43)$$

where the function $\text{sign}(\cdot)$ is applied to a matrix element-wise. Then, the coefficient can be conveniently expressed as

$$\gamma_K(\tau, \pi) = \frac{\text{tr}[\mathbf{C}(\tau)\mathbf{C}(\pi)^T]}{\text{tr}[\mathbf{C}(\tau)\mathbf{C}(\tau)^T]} = \frac{\text{tr}[\mathbf{C}(\tau)\mathbf{C}(\pi)^T]}{4k_4}. \quad (44)$$

The measure d_K and the numerator above are affine versions of each other, since $\text{tr}[\mathbf{C}(\tau)\mathbf{C}(\pi)^T] + 4d_K(\tau, \pi) = 4k_4$. This expression of γ_K in Eq.(44) can be compared with the Spearman coefficient γ_S in Eq.(38), and the positional

proximity coefficient γ_P in Eqs.(17, 18). All are special cases of the generalized correlation coefficient by using different types of scores over object pairs (i, j) . Table 1 summarizes all the relevant formulations for the existing and the proposed measures that act on permutation vectors.

As discussed in Section 2, consensus ranking methods [17]–[21] can rely on the minimization of Eq.(1), which typically leads to the problem of Eq.(2) when d_K is used to measure permutation distances. Here, we provide an alternative formulation as a QAP that can be contrasted to the previous analysis. Similar to manipulations used in Section 3.4, we obtain the equivalent maximization of

$$\sum_{k=1}^b w_k \text{tr}[\mathbf{C}(\tau)\mathbf{C}(\pi_k)^T] = \text{tr}[\mathbf{C}(\tau)\mathbf{N}^T] \\ = \text{tr}[\mathbf{P}_\tau \mathbf{C}(e) \mathbf{P}_\tau^T \mathbf{N}^T]. \quad (45)$$

The skew-symmetric matrix $\mathbf{N} = \sum_{k=1}^b w_k \mathbf{C}(\pi_k)$ collects the individual score matrices from the given permutations. $\mathbf{C}(e)$ is a constant matrix with 0, -1 and $+1$ on, above and below the main diagonal, respectively. It can be seen, that maximizing Eq.(45) corresponds to QAP $(\mathbf{C}(e), -\mathbf{N})$, with the distance and flow matrices being correspondingly $\mathbf{C}(e)$ and $-\mathbf{N}$. QAP formulations for consensus ranking using other distance measure variants and based on different formulations from the ones employed here can also be found in [17]. Another observation is that both formulations of Eq.(2) and Eq.(45) are making use of weighted sums of the problem data through the matrices \mathbf{Q} and \mathbf{N} . However, the former is not processing permutation vectors π_k or matrices \mathbf{P}_{π_k} , but rather relations \mathbf{R}^k and seeks the median relation matrix \mathbf{R} using a suitable set of transitivity constraints.

4 EXPERIMENTATION AND RESULTS

We now present various experiments to demonstrate the utility of the proposed contributions. Specifically, Section 4.1 shows how the proposed positional proximity coefficient γ_P can quantify the similarity between the outputs of different seriation algorithms. Also, Section 4.2 exemplifies the concept of consensus seriation through the merging of sequences from multiple algorithms using the proposed positional similarity d_P and the spectral approximation. The applicability of such tasks has been previously discussed in Sections 1, 2 and 3.3.

4.1 Comparing Different Seriation Algorithms

For the comparative evaluation, we have employed a set of sixteen seriation algorithms³ with varying characteristics. These, together with short descriptions of their inner workings, are given below:

- *Bond energy algorithm* (BEA): maximizes the measure of effectiveness defined using local similarities within the reordered distance matrix [15], [41].
- *Correlation order* (CO): relies on the angular position of the samples from the two main eigenvectors of the correlation matrix [42].

3. We have implemented a Matlab software toolbox that supports many seriation algorithms, different assessment and error measures, and various sequence manipulation routines; it is publicly available at <http://pcwww.liv.ac.uk/~goulerma/software/seriation.zip>.

- *Hierarchical clustering* (HC): returns the leaf order of the dendrogram constructed using an average linkage agglomerative clustering method [6]. The second variation improves this output by further applying optimal leaf ordering (OLO) [43].
- *Hamiltonian path* (HP): minimizes the Hamiltonian path through the graph of the object distances using a traveling salesman problem solver [15].
- *Linear seriation* (LS): optimizes the QAP of Eq.(5) with a distance matrix containing the distances between objects, and a flow matrix with ij th elements defined as $-|i - j|$ [32], [44].
- *Multidimensional scaling* (MDS1): returns the order of the objects along the first component generated using classical scaling. The second variation (MDS2) is based on non-metric scaling [45].
- *Principal component analysis* (PCA): a simple seriation method that returns the order of objects along the first principal component [15].
- *Rank-2 elliptic seriation* (R2E): locates the order of objects on the elliptic arrangement generated from repetitive application of correlation coefficient [10].
- *Simulated annealing* (SA): heuristically minimizes the anti-Robinson events of the distance matrix [2].
- *Sorting points into neighborhoods* (SPIN1): based on a fixed-point heuristic and a seriation template for the side-to-side version [7]. The second variation (SPIN2) is based on the neighborhood version [7].
- *Spectral seriation* (SS1): based on the eigen-decomposition of the unnormalized Laplacian matrix of the corresponding dissimilarity graph [46]. The second variation (SS2) is based on a symmetric normalized Laplacian [8].
- *Visual assessment of cluster tendency* (VAT): based on single linkage clustering where the order is returned via Prim's minimal spanning tree algorithm [5].

For LS, due to the large size of the datasets, we use a genetic algorithm to obtain an approximate solution in reasonable computation time. For HP, we use a heuristic solver based on pairwise node exchanges. The results with the SA method are entirely based on its implementation within the R seriation package [15].

To allow for a consistent comparison we have focused on a particular domain, namely bioinformatics, and we use the following seven gene expression datasets (additional results on a different domain can be found in the Supplementary material, Appendix E). Alpha CI [47] that contains 4,579 genes and 22 arrays representing genome-wide mRNA levels for yeast. Elutriation [47] that contains measured ratios of gene expression levels for yeast with 5,981 genes and 14 arrays. Colon tissues [48] with data from 2,000 genes across 62 tissues. Grr [49] that contains the steady-state responses to changes in growth rate of yeast from 5,454 genes and 45 arrays. Carcinoma [50] with 7,457 genes across 36 tissues. Adenoma [50] with 7,086 genes from 8 tissues. Finally, SRBCT [51] containing gene expression data for 2,308 genes across 63 tissues. To enable all algorithms to produce results in reasonable execution times, we have randomly selected $n=500$ samples from each original dataset. Subsequently, each similarity matrix between all samples is estimated using Pearson correlation, which is typical in this application

TABLE 2

Comparisons between the 16 evaluated algorithms based on the proximity coefficient γ_P . The upper triangle of the table contains the mean γ_P taken across all 7 datasets (values ≥ 0.8 are boldfaced), whereas the lower triangle displays the corresponding standard deviations.

	BEA	CO	HC	OLO	HP	LS	MDS1	MDS2	PCA	R2E	SA	SPIN1	SPIN2	SS1	SS2	VAT
BEA	•	0.493	0.543	0.508	0.429	0.576	0.580	0.506	0.567	0.540	0.534	0.582	0.456	0.539	0.584	0.452
CO	0.096	•	0.572	0.552	0.416	0.586	0.585	0.557	0.556	0.471	0.630	0.579	0.417	0.556	0.575	0.450
HC	0.079	0.107	•	0.603	0.419	0.626	0.626	0.584	0.625	0.526	0.611	0.625	0.458	0.615	0.631	0.447
OLO	0.092	0.130	0.196	•	0.429	0.667	0.663	0.644	0.650	0.565	0.679	0.666	0.416	0.612	0.662	0.477
HP	0.021	0.021	0.022	0.019	•	0.434	0.435	0.428	0.430	0.426	0.427	0.436	0.412	0.425	0.435	0.416
LS	0.108	0.100	0.089	0.105	0.018	•	0.990	0.855	0.957	0.635	0.866	0.986	0.463	0.859	0.982	0.468
MDS1	0.112	0.097	0.093	0.106	0.017	0.008	•	0.848	0.965	0.635	0.857	0.997	0.465	0.860	0.994	0.466
MDS2	0.087	0.081	0.074	0.120	0.024	0.158	0.159	•	0.835	0.566	0.886	0.839	0.454	0.813	0.831	0.448
PCA	0.104	0.064	0.091	0.087	0.008	0.076	0.076	0.164	•	0.615	0.816	0.969	0.468	0.844	0.975	0.479
R2E	0.126	0.188	0.084	0.131	0.031	0.087	0.082	0.118	0.028	•	0.614	0.640	0.395	0.581	0.639	0.466
SA	0.096	0.129	0.058	0.121	0.028	0.053	0.055	0.145	0.080	0.180	•	0.846	0.447	0.799	0.834	0.465
SPIN1	0.122	0.098	0.098	0.104	0.016	0.013	0.003	0.164	0.070	0.079	0.059	•	0.465	0.855	0.998	0.468
SPIN2	0.049	0.063	0.034	0.036	0.025	0.064	0.065	0.071	0.060	0.053	0.068	0.066	•	0.447	0.464	0.427
SS1	0.074	0.063	0.068	0.114	0.013	0.204	0.205	0.203	0.226	0.073	0.140	0.210	0.060	•	0.850	0.464
SS2	0.126	0.097	0.097	0.101	0.014	0.017	0.007	0.166	0.060	0.073	0.063	0.002	0.064	0.213	•	0.469
VAT	0.070	0.076	0.092	0.087	0.020	0.072	0.072	0.027	0.095	0.062	0.065	0.073	0.042	0.037	0.077	•

domain [6], [13], [48]. As the seriation algorithms rely on dissimilarities, the correlation values are subtracted from the unity according to [52].

Each of the sixteen algorithms is applied independently to the dissimilarity matrix of each of the seven datasets and the generated ordering sequences are recorded. Then, the similarities between all 120 possible algorithm pairs are evaluated using the unnormalized version of the proposed coefficient γ_P . For each pair, the mean γ_P across all seven datasets is calculated together with the standard deviation. All these values are collected in Table 2 with the most similar algorithms shown boldfaced. As can be seen, there are algorithms with highly similar output, such as the pairs {LS, SS2}, {MDS1, SS2} and {PCA, SPIN1}, with corresponding average γ_P values of 0.982, 0.994 and 0.969. Other pairs have very low similarity; examples include {CO, HP} and {R2E, SPIN2} with average γ_P of 0.416 and 0.395, respectively. The low overall standard deviations show consistency of these findings across the datasets. This is also supported by the fact that the datasets are from a particular domain and are likely to exhibit similar characteristics.

The information from Table 2 can also be used to group the algorithms according to their pairwise similarities. A straightforward way of doing so, is to hierarchically cluster the algorithms using the upper part of the table. Fig.2 shows the resulting dendrogram representing the different group hierarchies together with the corresponding linkage values. It can be seen that at the bottom of the figure, the algorithms {PCA, LS, MDS1, SS2, SPIN1} are all grouped together in a tight cluster, as all average pairwise coefficients between them are greater than 0.95. Other algorithms such as {SA, MDS2} form a moderately less tight group with similarity of 0.886. Remaining algorithms show to produce less similar seriation output; for example SPIN2 and VAT have less than 0.48 similarity with any other method.

Fig.3 presents a qualitative comparison between similar and dissimilar generated sequences. The figure displays the ordered distance maps for some selected algorithms applied to a dataset. Specifically, Figs.3(a,b) which correspond to LS and SS2 show to be almost identical, and this is corroborated by the fact that their averaged similarity in Table 2 registers very high ($\gamma_P = 0.982$). CO and HP, on the other hand, have

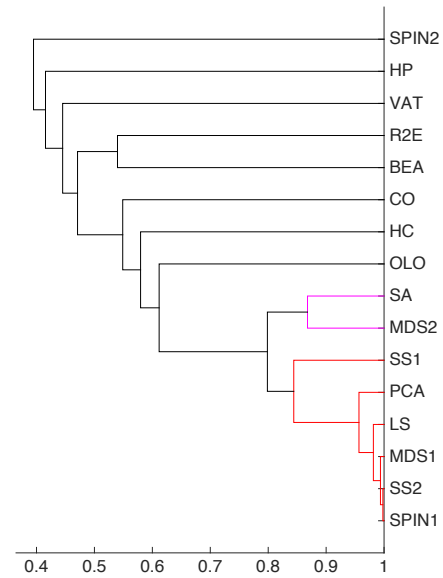


Fig. 2. Dendrogram (of cophenetic correlation coefficient = 0.98) computed using complete linkage and the averaged γ_P values from Table 2. Groups with linkage values (shown on the horizontal axis) of 0.8 and over are drawn in different shades.

a much lower similarity ($\gamma_P = 0.416$) which can be visually verified by the distinctly contrasting structures within the distance maps of Figs.3(c,d). Comparably, differing visual patterns in the maps accompanied by low coefficient values can also be observed between the pairs SS2 and HP ($\gamma_P = 0.435$), LS and CO ($\gamma_P = 0.586$), and SS2 and CO ($\gamma_P = 0.575$).

Using the proposed coefficient to compare the ordering sequences generated by different algorithms can be very useful to the practitioners of a specific field in a variety of ways. For example, Table 2 shows that LS and MDS1 give almost identical output ($\gamma_P = 0.99$). But, since MDS1 can be executed much faster than LS as it relies on an eigen-decomposition, whereas LS needs to solve approximately a QAP problem, the use of MDS1 in terms of computational efficiency may stand as a better option. Another beneficial application would be to select a subset of algorithms with mostly dissimilar results. Dissimilarly performing algorithms could be used to capture different

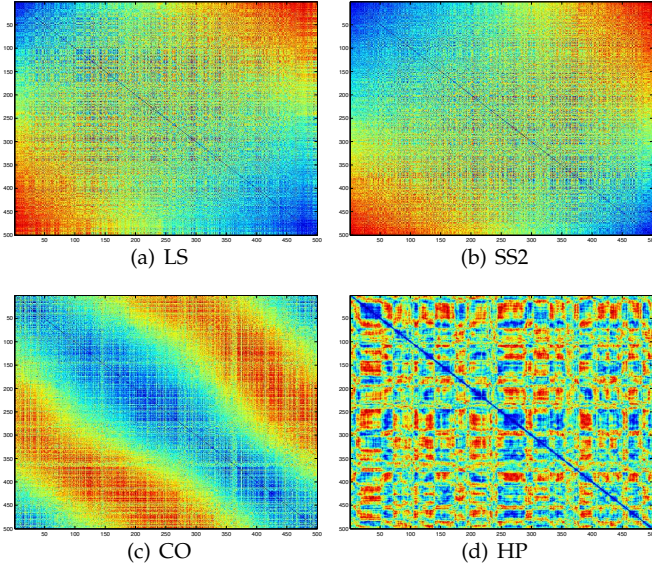


Fig. 3. Dissimilarity matrices of the Elutriation dataset, ordered using the output from four different algorithms.

views and structural characteristics of a given dataset. Their output could also be used as sequences for fusion by a consensus seriation approach (see Section 4.2), or even as multiple seed solutions in heuristic seriation algorithms, such as the population-based optimizer for LS used here. It is also possible, instead of comparing two different algorithms across multiple datasets, to compare the output of a single algorithm on measurements obtained by varying the experimental conditions, or a single dataset where the sample (dis)similarities are obtained in different ways (e.g., using L_p distances, kernels, correlations, divergences, etc.).

4.2 Consensus Seriation Experiments

In this section, we demonstrate the use of the consensus seriation method proposed in Section 3.4. For the purpose of presenting different examples, from the total of sixteen algorithms we choose five subsets T_i that contain between three and six algorithms whose outputs are fused independently. These subsets are: $T_1 = \{\text{MDS1, R2E, VAT}\}$, $T_2 = \{\text{MDS1, SS1, VAT}\}$, $T_3 = \{\text{MDS1, OLO, SS1, VAT}\}$, $T_4 = \{\text{BEA, LS, MDS2, SS1, VAT}\}$, and $T_5 = \{\text{HP, LS, MDS1, MDS2, SS1, VAT}\}$. The members of these subsets are mostly chosen randomly, but as it is not very useful to combine sequences that are too similar, care was taken to avoid having many similar ones for the shorter subsets (relying on Table 2). For example, the most similar pair in T_1 is $\{\text{MDS1, R2E}\}$ with $\gamma_P = 0.635$, and in T_2 and T_3 only the pair $\{\text{MDS1, SS1}\}$ has relatively high $\gamma_P = 0.86$ whereas the other pairs have $\gamma_P < 0.665$. For the remaining two larger subsets more similar pairs are allowed. We also set varying degrees of overlap between the subsets to examine the corresponding differences between the results. For example, T_2 and T_3 have three common members, and T_4 and T_5 have five. The ordering sequences used for fusion are the exact same ones that resulted from the experiments in Section 4.1 using $n=500$ samples per dataset. The Hamming based consensus method is implemented using a LAP solver (Section 3.5.1), the Spearman using sorting (Section 3.5.2) and the Kendall

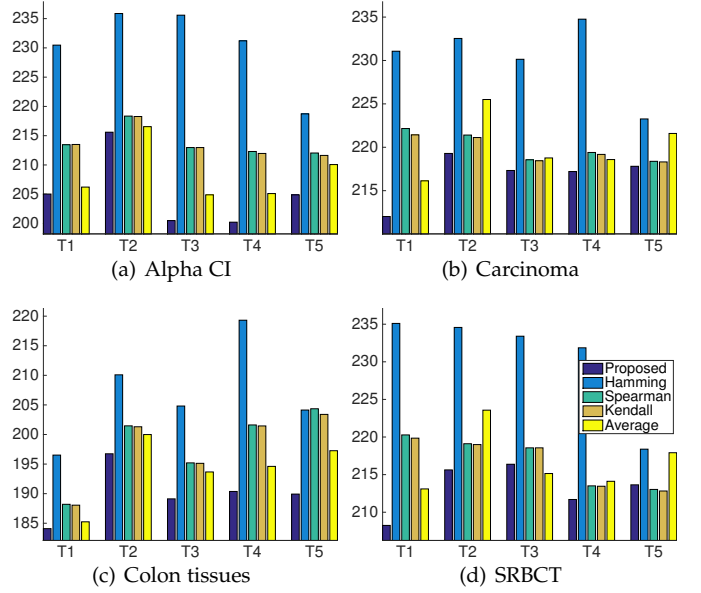


Fig. 4. ARGAR errors for consensus seriation results obtained with the proposed, Hamming, Spearman and Kendall measures, showing also the average of the original algorithms comprising each T_i .

using a genetic algorithm to approximate the underlying QAP (Section 3.5.3). In all experiments, as Hamming, Spearman and Kendall are not insensitive to the orientation of sequences, to make the comparison more objective for those three methods, we repeat the fusion procedure with all possible combinations of having each sequence flipped and keep the best outcome. Also, for simplicity we set all weights w_k to be equal; this renders equal significance to all fused sequences.

To quantitatively compare the seriation results, we use the relative generalized anti-Robinson events (RGAR) error measure [6]. Given a symmetric dissimilarity matrix $\mathbf{D} \in \mathcal{R}^{n \times n}$, the measure counts the number of negative gradients horizontally and vertically away from the main diagonal. In essence, it counts the number of object triples that make \mathbf{D} depart from an ideal anti-Robinson form (see Section 3.4), which is a natural expectation in seriation [15]. This measure is further restricted by a window of size δ and is normalized by the maximum number of negative gradients. In a simplified form, it can be expressed as

$$\text{RGAR}(\mathbf{D}, \delta) = \frac{\sum_{j-\delta \leq i < k < j} [\mathbf{D}_{ik} > \mathbf{D}_{ij}] + \sum_{j-\delta \leq i < k < j} [\mathbf{D}_{kj} > \mathbf{D}_{ij}]}{(\frac{2}{3} - n)\delta + n\delta^2 - \frac{2}{3}\delta^3}.$$

The advantage of RGAR over other measures (e.g., ones reviewed in [15]) is that by varying δ , we can assess the quality of ordering at multiple structural levels. Specifically, smaller/larger values of δ enable RGAR to inspect more the local/global aspects of the seriated map \mathbf{D} . In addition to this measure, we also make use of its accumulated version $\text{ARGAR}(\mathbf{D}) = \sum_{\delta=2}^{n-1} \text{RGAR}(\mathbf{D}, \delta)$ across all window sizes. This provides an overall image for the ordering quality.

Fig.4 displays the ARGAR errors of the proposed consensus method and the ones derived from the existing measures. For each subset T_i , each consensus method is applied to combine the sequences generated from all member

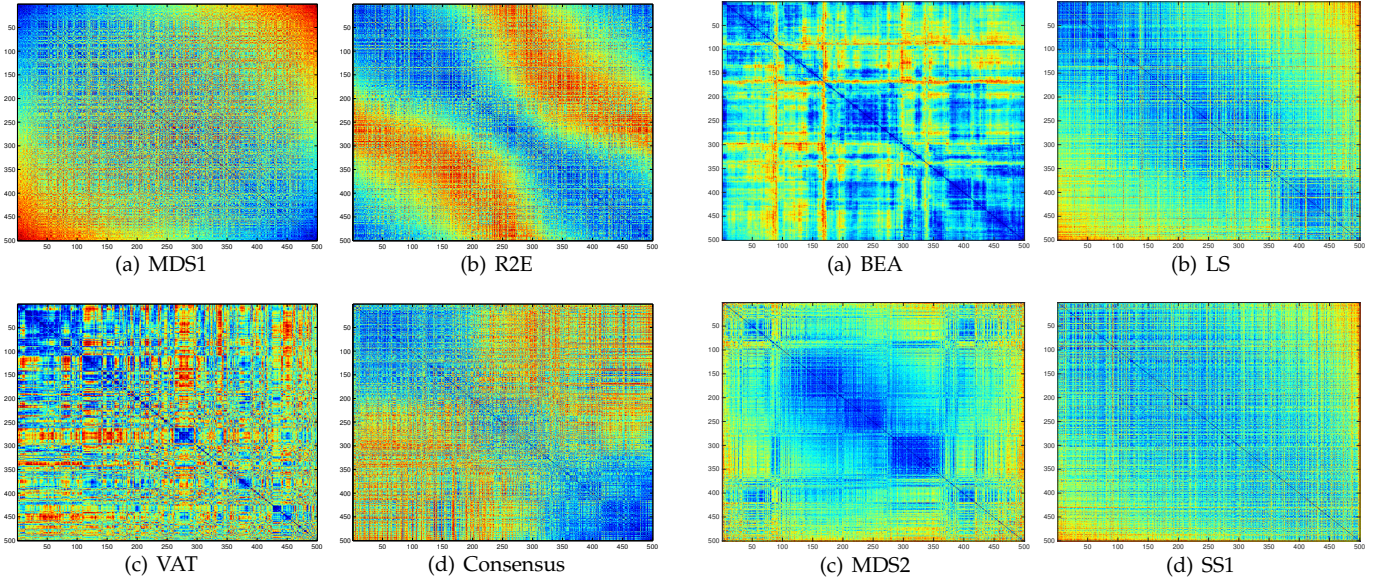


Fig. 5. Seriated maps for the individual algorithms of subset T_1 as well as the proposed consensus method, for the dataset Alpha CI.

algorithms of that particular subset. Each experiment is also repeated independently to four selected datasets. It can be seen that for all the cases but one, the proposed method performs better than Hamming, Spearman and Kendall based consensus task. Only for the SRBCT dataset and subset T_5 , Spearman and Kendall based fusions show to be better with ARGAR errors of 213.0 and 212.8, respectively, compared to 213.6 of the proposed method (both Kendall and Spearman based methods have very similar errors in most cases). The plots in Fig.4 also include the ARGAR errors averaged over all the individual algorithms comprising each subset T_i . It can be seen, that only in subset T_3 for the SRBCT dataset the proposed method is worse than the average with an ARGAR difference of 1.2. However, in all other cases the former is consistently below the error of the average with a mean ARGAR difference of 3.9. These observations show that the proposed consensus fusion outperforms Hamming, Spearman and Kendall, and is mostly better than the average error of the fused sequences. In general though, it cannot be expected to be better than all fused sequences as the optimization does not take into account RGAR errors. The figure plots also show that even for subsets with large overlap, the results can vary distinctly. For instance, T_2 and T_3 in Figs.4(a,c), and T_4 and T_5 in Figs.4(a,d).

We use Fig.5 to qualitatively compare the seriated distance maps for the proposed method and each of the original algorithms for the selected subset $T_1 = \{MDS1, R2E, VAT\}$ and dataset Alpha CI. The map of MDS1 shows two main concentrations of samples: those with low (≤ 200) and those with high (≥ 300) sequence index positions. The concentrations are indicated by the small dissimilarity values in the top-left and bottom-right corners of the map and the high values in the other two corners. This map describes the more global aspects of the dataset as it reveals the sample separation at a larger scale. The blocky structure of the VAT map, on the other hand, captures more the local dissimilarity characteristics of the samples as it reveals multiple local cluster fragments. The R2E map captures mostly the mid-

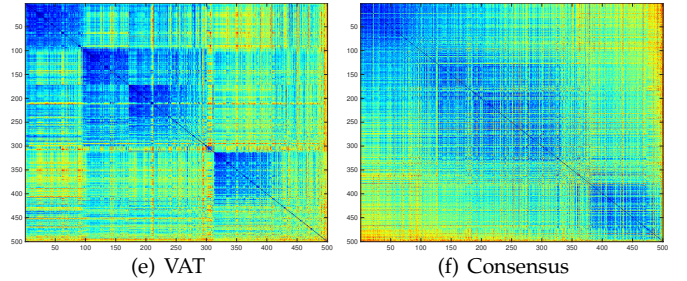


Fig. 6. Seriated maps for the individual algorithms of subset T_4 as well as the proposed consensus method, for the dataset Colon tissues.

level structure of the samples, as most small dissimilarity values are lying along the main diagonal. The map of the fused sequence in Fig.5(d) is derived by combining all three previous sequences. Overall, this map seems to have the global concentration characteristics of MDS1, some of the linear variations of R2E along the diagonal with increasing dissimilarities moving off the diagonal, and some of the sporadic blocky structures from VAT. This demonstrates that consensus seriation can generate novel solutions unattainable by any of the original methods; this principal advantage is also the case in consensus clustering approaches [22]. Similar observations can also be drawn from Fig.6, which displays the maps for each of the individual algorithms within $T_4 = \{BEA, LS, MDS2, SS1, VAT\}$ for another dataset. BEA, MDS2 and VAT describe more the local aspects of the data, while LS and SS1 the global structure. The consensus map in Fig.6(f) resembles the LS one, but exhibits sharper separations between the four main concentrations which is likely owed to the three sequences emphasizing the local structure.

To quantitatively compare the methods at different scales, we present Fig.7 which records the RGAR errors for the entire range $\delta \in [2, n - 1]$ of windows. The errors are measured for the sequences of MDS1, R2E and VAT in T_1 , as well as the Hamming, Spearman, Kendall based fusion methods and the proposed one for the Alpha CI dataset. It can be seen, that VAT performs very well at smaller window sizes by starting off with its minimum RGAR value of 0.31 at

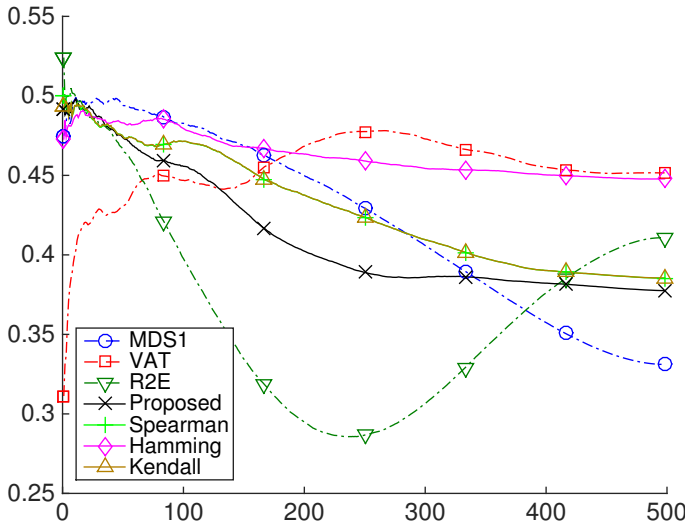


Fig. 7. RGAR errors for the sequence generating algorithms in T_1 and all consensus methods plotted against all window sizes δ .

$\delta=2$ and then steeply increasing to an error of 0.45 at $\delta=70$. This confirms the local capabilities of the algorithm owed to its linkage-based search. Conversely, MDS1 works better on the global structure due to its projection-based ordering, and outperforms all other methods after window size $\delta=385$, with finally reaching a minimum RGAR of 0.33. R2E is better at the mid-level structure (similarly observed in [6]), as it shows lower errors at middle window sizes and reaches a minimum RGAR of 0.29 at $\delta=237$. This is likely owed to its gradual convergence to a low-dimensional elliptical structure via successively iterating correlation matrices. The proposed consensus method seems to have lower RGAR than R2E within the first 20 and the last 80 window sizes. Also, after $\delta=130$ with RGAR of 0.44, it performs much better than VAT. For δ values between 130 to 340, that define mostly the mid-level structure, it outperforms both VAT and MDS1. Only for a small window range between 340 and 410 it is worse than both R2E and MDS1. These observations, and similarly for experiments with other algorithms and datasets, show that consensus seriation can capture different characteristics from the fused sequences expressed at diverse scales. Another observation is that the fused sequence exhibits relatively less abrupt error variations across the different scales. This can be seen by the RGAR ranges (standard deviations), which for the MDS1, R2E, VAT and the proposed consensus are 0.168 (0.056), 0.238 (0.062), 0.167 (0.020) and 0.121 (0.037), respectively; that is, the consensus has the narrowest range and the second smallest deviation. Finally, with regard to Hamming, Spearman and Kendall based consensus, the proposed one performs similarly for δ values up to 35, and thereafter maintains consistently lower RGAR errors.

5 CONCLUSION

We have introduced a novel measure for comparing arbitrary object sequences generated by seriation algorithms and an instance of the generalized correlation coefficient based on this measure. Its principal advantage, compared to other existing measures evaluating permutation vectors,

is that it takes explicitly into account positional distances between object pairs and it is invariant to object ordering. Various statistical properties to aid the understanding and applicability of the measure and the coefficient have been presented. Furthermore, we have introduced a consensus seriation method formulated as a quadratic assignment problem that can be approximated by a fast spectral optimization procedure. We have also derived the combinatorial consensus formulations of the Hamming, Spearman and Kendall measures for reasons of comparison with the proposed model. Finally, thorough experimentations demonstrated the utility of the introduced coefficient by comparing multiple seriation algorithms applied to various datasets. These also supported the qualitative and quantitative analyses of the different consensus formulations.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive and helpful comments.

REFERENCES

- [1] L. Hubert, P. Arabie, and J. Meulman, *Combinatorial Data Analysis: Optimization by Dynamic Programming*. Society for Industrial and Applied Mathematics, 2001.
- [2] M. J. Brusco and S. Stahl, *Branch-and-Bound Applications in Combinatorial Data Analysis*, ser. Statistics & Computing. Springer, 2005.
- [3] W. M. F. Petrie, "Sequences in prehistoric remains," *The Journal of the Anthropological Institute of Great Britain and Ireland*, vol. 29, no. 3/4, pp. 295–301, 1899.
- [4] D. Mavroudis and E. Bingham, "Enhancing the stability and efficiency of spectral ordering with partial supervision and feature selection," *Knowledge and Information Systems*, vol. 23, no. 2, pp. 243–265, 2010.
- [5] T. Havens and J. Bezdek, "An efficient formulation of the improved visual assessment of cluster tendency (iVAT) algorithm," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 813–822, 2012.
- [6] Y.-J. Tien, Y.-S. Lee, H.-M. Wu, and C.-H. Chen, "Methods for simultaneously identifying coherent local clusters with smooth global patterns in gene expression profiles," *Bioinformatics*, vol. 9, no. 155, 2008.
- [7] D. Tsafir, I. Tsafir, L. Ein-Dor, O. Zuk, D. Notterman, and E. Domany, "Sorting points into neighborhoods (SPIN): data analysis and visualization by ordering distance matrices," *Bioinformatics*, vol. 21, no. 10, pp. 2301–2308, 2005.
- [8] C. Ding and X. He, "Linearized cluster assignment via spectral ordering," in *Proceedings of the 21st International Conference on Machine Learning, ICML, New York, NY, USA, 2004*, pp. 30–37.
- [9] I. Liiv, "Seriation and matrix reordering methods: An historical overview," *Statistical Analysis and Data Mining*, vol. 3, no. 2, pp. 70–91, 2010.
- [10] C.-H. Chen, "Generalized association plots: information visualization via iteratively generated correlation matrices," *Statistica Sinica*, vol. 12, pp. 7–29, 2002.
- [11] A. Abbott, "Sequence analysis: New methods for old ideas," *Annual Review of Sociology*, vol. 21, no. 1, pp. 93–113, 1995.
- [12] F. Marcotorchino, "Seriation problems: An overview," *Applied Stochastic Models and Data Analysis*, vol. 7, no. 2, pp. 139–151, 1991.
- [13] H.-M. Wu, Y.-J. Tien, and C.-H. Chen, "GAP: A graphical environment for matrix visualization and cluster analysis," *Computational Statistics and Data Analysis*, vol. 54, no. 3, pp. 767–778, 2010.
- [14] S. Climer and W. Zhang, "Rearrangement clustering: Pitfalls, remedies, and applications," *Journal of Machine Learning Research*, vol. 7, pp. 919–943, 2006.
- [15] M. Hahsler, K. Hornik, and C. Buchta, "Getting things in order: an introduction to the R package seriation," *Journal of Statistical Software*, vol. 25, no. 3, pp. 1–34, 2008.
- [16] D. E. Critchlow, *Metric Methods for Analyzing Partially Ranked Data*, ser. Lecture Notes in Statistics. Springer, 1985, vol. 34.

- [17] W. D. Cook, "Distance-based and ad hoc consensus models in ordinal preference ranking," *European Journal of Operational Research*, vol. 172, no. 2, pp. 369–385, 2006.
- [18] K. Hornik and D. Meyer, "Deriving consensus rankings from benchmarking experiments," in *Advances in Data Analysis*, R. Decker and H.-J. Lenz, Eds. Springer, 2007, pp. 163–170.
- [19] M. Meila, K. Phadnis, A. Patterson, and J. A. Biles, "Consensus ranking under the exponential model," in *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, UAI, R. Parr and L. C. van der Gaag, Eds., 2007, pp. 285–294.
- [20] F. Marcotorchino and P. Michaud, "Agrégation de similarités en classification automatique," *Revue de Statistique Appliquée*, vol. 30, no. 2, pp. 21–44, 1982.
- [21] Y. Wakabayashi, "The complexity of computing medians of relations," *Resenhas IME-USP*, vol. 3, no. 3, pp. 323–349, 1998.
- [22] A. Topchy, A. K. Jain, and W. Punch, "Clustering ensembles: Models of consensus and weak partitions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1866–1881, 2005.
- [23] S. Vega-Pons and J. Ruiz-Shulcloper, "A survey of clustering ensemble algorithms," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 3, pp. 337–372, 2011.
- [24] R. Burkard and E. Çela, "Linear assignment problems and extensions," in *Handbook of Combinatorial Optimization*, D.-Z. Du and P. M. Pardalos, Eds. Kluwer, 1999, pp. 75–149.
- [25] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2009.
- [26] E. Çela, *The Quadratic Assignment Problem: Theory and Algorithms*, ser. Combinatorial Optimization. Springer, 1997.
- [27] R. E. Burkard, E. Çela, P. M. Pardalos, and L. S. Pitsoulis, "The quadratic assignment problem," in *Handbook of Combinatorial Optimization*, D.-Z. Du and P. M. Pardalos, Eds. Kluwer, 1998, pp. 241–338.
- [28] K. M. Anstreicher, "Recent advances in the solution of quadratic assignment problems," *Mathematical Programming*, vol. 97, no. 1-2, pp. 27–42, 2003.
- [29] E. M. Loiola, N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido, "A survey for the quadratic assignment problem," *European Journal of Operational Research*, vol. 176, no. 2, pp. 657–690, 2007.
- [30] G. Hardy, J. Littlewood, and G. Pólya, *Inequalities*, ser. Cambridge Mathematical Library. Cambridge University Press, 1952.
- [31] H. E. Daniels, "The relation between measures of correlation in the universe of sample permutations," *Biometrika*, vol. 33, no. 2, pp. 129–135, 1944.
- [32] L. Hubert and J. Schultz, "Quadratic assignment as a general data analysis strategy," *British Journal of Mathematical and Statistical Psychology*, vol. 29, no. 2, pp. 190–241, 1976.
- [33] S. van Dongen and A. J. Enright, "Metric distances derived from cosine similarity and pearson and spearman correlations," *CoRR*, vol. abs/1208.3145, 2012.
- [34] U. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [35] T. Mu, J. Y. Goulermas, S. Ananiadou, and J. Tsujii, "Proximity-based frameworks for generating embeddings from multi-output data," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2216–2232, 2012.
- [36] A. George and A. Pothen, "An analysis of spectral envelope reduction via quadratic assignment problems," *SIAM Journal on Matrix Analysis and Applications*, vol. 18, no. 3, pp. 706–732, 1997.
- [37] S. T. Barnard, A. Pothen, and H. Simon, "A spectral algorithm for envelope reduction of sparse matrices," *Numerical Linear Algebra with Applications*, vol. 2, no. 4, pp. 317–334, 1995.
- [38] J. E. Atkins, E. G. Boman, and B. Hendrickson, "A spectral algorithm for seriation and the consecutive ones problem," *SIAM Journal on Computing*, vol. 28, pp. 297–310, 1998.
- [39] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge University Press, 1990.
- [40] G. W. Stewart, *Matrix Algorithms, Volume II: Eigensystems*. SIAM, 2001.
- [41] W. T. McCormick, P. J. Schweitzer, and T. W. White, "Problem decomposition and data reorganization by a clustering technique," *Operations Research*, vol. 20, no. 5, pp. 993–1009, 1972.
- [42] M. Friendly, "Corrgrams: Exploratory displays for correlation matrices," *The American Statistician*, vol. 56, no. 4, pp. 316–324, 2002.
- [43] Z. Bar-Joseph, D. K. Gifford, and T. S. Jaakkola, "Fast optimal leaf ordering for hierarchical clustering," *Bioinformatics*, vol. 17, no. suppl 1, pp. S22–S29, 2001.
- [44] G. Caraux and S. Pinloche, "Permutmatrix: a graphical environment to arrange gene expression profiles in optimal linear order," *Bioinformatics*, vol. 21, no. 7, pp. 1280–1281, 2005.
- [45] A. Webb, *Statistical Pattern Recognition*. Wiley, 2003.
- [46] S. T. Barnard, A. Pothen, and H. D. Simon, "A spectral algorithm for envelope reduction of sparse matrices," in *ACM/IEEE Conference on Supercomputing*, 1993, pp. 493–502.
- [47] O. Alter, P. O. Brown, and D. Botstein, "Singular value decomposition for genome-wide expression data processing and modeling," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, no. 18, pp. 10101–10106, 2000.
- [48] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *PNAS*, vol. 96, no. 12, pp. 6745–6750, 1999.
- [49] N. Slavov and D. Botstein, "Coupling among growth rate response, metabolic cycle, and cell division cycle in yeast," *Molecular Biology of the Cell*, vol. 22, no. 12, pp. 1997–2008, 2011.
- [50] D. A. Notterman, U. Alon, A. J. Sierk, and A. J. Levine, "Transcriptional gene expression profiles of colorectal adenoma, adenocarcinoma, and normal tissue examined by oligonucleotide arrays," *Cancer Research*, vol. 61, no. 7, pp. 3124–3130, 2001.
- [51] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature Medicine*, vol. 7, no. 6, pp. 673–679, 2001.
- [52] T. Cox and A. Cox, *Multidimensional Scaling, 2nd Edition*, ser. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. CRC Press, 2000.



John Yannis Goulermas (M'98, S'10) obtained the B.Sc. (1st) degree in computation from the University of Manchester (UMIST), in 1994, and the M.Sc. and Ph.D. degrees from the Control Systems Center, UMIST, in 1996 and 2000, respectively. He is currently a Reader in the Department of Computer Science at the University of Liverpool. His current research interests include machine learning, combinatorial data analysis, data visualization as well as mathematical modeling. He has worked with various application areas including image/video analysis, biomedical engineering and biomechanics, industrial monitoring and control, and security.



Alexandros Kostopoulos received his MEng degree in avionic systems with pilot studies from the University of Liverpool in 2011. He is currently working towards obtaining a PhD in combinatorial data analysis. He is interested in machine learning, data ordering, feature generation and dimensionality reduction. He is a keen diver, having reached BSAC dive leader, and open water instructor, and also enjoys playing field hockey.



Tingting Mu (M'05) received the B.Eng. degree in electronic engineering and information science from the University of Science and Technology of China, Hefei, China, in 2004, and the Ph.D. degree in electrical engineering and electronics from the University of Liverpool in 2008. She is currently a Lecturer in the Department of Electrical Engineering and Electronics at the University of Liverpool. Her current research interests include machine learning, data visualization and mathematical modeling, with applications to information retrieval, text mining and bioinformatics.

Supplementary Material for the Manuscript: A New Measure for Analyzing and Fusing Sequences of Objects.

J. Y. Goulermas, A. Kostopoulos, T. Mu

Version 2

APPENDIX A: SIMPLIFICATION OF THE POSITIONAL PROXIMITY COEFFICIENT γ_P

In this appendix, we show how to simplify Eq.(17) of the manuscript in order to obtain the final version $\gamma_P(\tau, \pi) = \frac{\text{tr}[\mathbf{A}(\tau)\mathbf{A}(\pi)]}{\frac{n^6}{15} - \frac{n^4}{6} + \frac{n^2}{10}}$ given in Eq.(18). We firstly expand on the denominator of Eq.(17) according to

$$\text{tr} [\mathbf{A}(\tau)\mathbf{A}(\tau)^T] = \sum_{i,j=1}^n (\tau(i) - \tau(j))^4 = \quad (\text{A.1})$$

$$\begin{aligned} \sum_{i,j=1}^n (i-j)^4 &= \sum_{i,j=1}^n (i^4 + j^4 - 4i^3j - 4ij^3 + 6i^2j^2) = \\ 2n \sum_{i=1}^n i^4 - 8 \left(\sum_{i=1}^n i^3 \right) \left(\sum_{i=1}^n i \right) + 6 \left(\sum_{i=1}^n i^2 \right)^2. \end{aligned} \quad (\text{A.2})$$

We can then directly replace the four summation terms in Eq.(A.2) with the first four Faulhaber power-sum formulae [1] that are

$$f_1 \equiv \sum_{i=1}^n i = \frac{n^2 + n}{2}, \quad (\text{A.3})$$

$$f_2 \equiv \sum_{i=1}^n i^2 = \frac{2n^3 + 3n^2 + n}{6}, \quad (\text{A.4})$$

$$f_3 \equiv \sum_{i=1}^n i^3 = \frac{n^4 + 2n^3 + n^2}{4}, \quad (\text{A.5})$$

$$f_4 \equiv \sum_{i=1}^n i^4 = \frac{6n^5 + 15n^4 + 10n^3 - n}{30}. \quad (\text{A.6})$$

After applying thorough simplifications to the substituted expression, we obtain the following quantity which is equal to Eq.(A.2) and also the denominator of Eq.(18)

$$\begin{aligned} 2n \left(\frac{n^5}{5} + \frac{n^4}{2} + \frac{n^3}{3} - \frac{n}{30} \right) - \frac{n^2 + n}{2} (2n^4 + 4n^3 + 2n^2) \\ + 6 \left(\frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6} \right)^2 = \frac{n^6}{15} - \frac{n^4}{6} + \frac{n^2}{10} \equiv \zeta. \end{aligned} \quad (\text{A.7})$$

APPENDIX B: DERIVATION OF THE MEAN AND VARIANCE OF $\hat{\gamma}_P$

Here we show the derivation of the mean $E[\hat{\gamma}_P]$ and variance $\sigma^2(\hat{\gamma}_P)$ in Eqs.(19,20). For both quantities, we firstly

employ formulae for estimating the mean and variance of generic QAP expressions, such as those of Eq.(5)

$$\Gamma(\pi) \equiv \sum_{i,j=1}^n \mathbf{X}_{\pi(i),\pi(j)} \mathbf{Y}_{ij} = \text{tr} [\mathbf{P}_\pi \mathbf{X} \mathbf{P}_\pi^T \mathbf{Y}^T]. \quad (\text{B.1})$$

The distance matrix \mathbf{X} and the flow one \mathbf{Y} are assumed fixed $n \times n$ matrices of zero diagonals and $\Gamma(\pi)$ is parameterized by the permutation $\pi \in S_n$. As stated in [2] the mean is given by

$$E[\Gamma(\pi)] = \frac{1}{n(n-1)} \sum_{i,j} \mathbf{X}_{ij} \sum_{i,j} \mathbf{Y}_{ij}, \quad (\text{B.2})$$

where all summation limits 1 and n are henceforth dropped for simplicity.

In the case of $\hat{\gamma}_P$, however, we have the special case of $\mathbf{X} = \mathbf{Y} = \mathbf{A}(e)$, because

$$\hat{\gamma}_P(\rho) = \frac{\text{tr} [\mathbf{A}(\rho)\mathbf{A}(e)]}{\zeta} = \frac{\text{tr} [\mathbf{P}_\rho \mathbf{A}(e) \mathbf{P}_\rho^T \mathbf{A}(e)]}{\zeta} \quad (\text{B.3})$$

where ζ is the normalizing quantity derived in Eq.(A.7). Therefore, the quantity $\sum_{i,j} \mathbf{X}_{ij}$ corresponds to

$$\begin{aligned} \sum_{i,j} \mathbf{A}(e) &= \sum_{i,j} (i-j)^2 = 2n \sum_i i^2 - 2 \left(\sum_i i \right)^2 \\ &= 2nf_2 - 2f_1^2 = \frac{n^4 - n^2}{6} \equiv \zeta_1, \end{aligned} \quad (\text{B.4})$$

where the last term is obtained by substituting with Eqs.(A.3,A.4) and simplifying.

Finally, by combining Eqs.(B.2,B.3,B.4) and thoroughly simplifying, we obtain the following expression for the mean

$$E[\hat{\gamma}_P] = \frac{1}{n(n-1)} \frac{\zeta_1^2}{\zeta} = \frac{5n(n+1)}{12n^2 - 18}. \quad (\text{B.5})$$

For the derivation of the variance, we also employ a generic formula from [2] suited for QAP expressions with symmetric data matrices \mathbf{X} and \mathbf{Y} . It is given by

$$\begin{aligned} \sigma^2[\Gamma(\pi)] &= -\frac{\alpha_1}{n^2(n-1)^2} + \frac{2\alpha_2}{n(n-1)} + \frac{4\alpha_3}{n(n-1)(n-2)} \\ &+ \frac{\alpha_4}{n(n-1)(n-2)(n-3)}, \end{aligned} \quad (\text{B.6})$$

where

$$\alpha_1 = \left(\sum_{i,j} \mathbf{X}_{ij} \right)^2 \left(\sum_{i,j} \mathbf{Y}_{ij} \right)^2, \quad (\text{B.7})$$

$$\alpha_2 = \sum_{i,j} \mathbf{X}_{ij}^2 \sum_{i,j} \mathbf{Y}_{ij}^2, \quad (\text{B.8})$$

$$\alpha_3 = \left[\sum_i \left(\sum_j \mathbf{X}_{ij} \right)^2 - \sum_{i,j} \mathbf{X}_{ij}^2 \right] \times \left[\sum_i \left(\sum_j \mathbf{Y}_{ij} \right)^2 - \sum_{i,j} \mathbf{Y}_{ij}^2 \right], \quad (\text{B.9})$$

$$\alpha_4 = \left[\left(\sum_{i,j} \mathbf{X}_{ij} \right)^2 - 4 \sum_i \left(\sum_j \mathbf{X}_{ij} \right)^2 + 2 \sum_{i,j} \mathbf{X}_{ij}^2 \right] \times \left[\left(\sum_{i,j} \mathbf{Y}_{ij} \right)^2 - 4 \sum_i \left(\sum_j \mathbf{Y}_{ij} \right)^2 + 2 \sum_{i,j} \mathbf{Y}_{ij}^2 \right]. \quad (\text{B.10})$$

Similarly as before, since $\mathbf{X} = \mathbf{Y} = \mathbf{A}(e)$, all terms α_i depend only on n . Firstly, from Eq.(B.4) we have

$$\alpha_1 = \zeta_1^4, \quad (\text{B.11})$$

and from Eqs.(A.1,A.7) we have

$$\alpha_2 = \zeta^2. \quad (\text{B.12})$$

For α_3 and α_4 we need to calculate the quantity $\sum_i \left(\sum_j \mathbf{X}_{ij} \right)^2$, which corresponds to

$$\begin{aligned} \sum_i \left(\sum_j (i-j)^2 \right)^2 &= \sum_i \left(ni^2 + \sum_j j^2 - 2i \sum_j j \right)^2 \\ &= \sum_i (ni^2 + f_2 - 2if_1)^2 \\ &= \sum_i (4f_1^2 i^2 - 4f_1 f_2 i - 4nf_1 i^3 + f_2^2 + 2nf_2 i^2 + n^2 i^4) \\ &= -4nf_1 f_3 + 3nf_2^2 + n^2 f_4 = \frac{n^7}{30} - \frac{n^5}{12} + \frac{n^3}{20} \equiv \zeta_2 \end{aligned} \quad (\text{B.13})$$

where the last quantity is obtained by substituting the f_i terms from Eqs.(A.3-A.6) and simplifying. From Eq.(B.9), we can then set and simplify as

$$\alpha_3 = (\zeta_2 - \zeta)^2 \quad (\text{B.14})$$

$$= \left[\frac{n^2 (2n^5 - 4n^4 - 5n^3 + 10n^2 + 3n - 6)}{60} \right]^2.$$

Similarly for the last term α_4 we have

$$\alpha_4 = (\zeta_1^2 - 4\zeta_2 + 2\zeta)^2 \quad (\text{B.15})$$

$$= \left[\frac{n^2 (n-1)^2 (5n^4 - 14n^3 - 19n^2 + 36n + 36)}{180} \right]^2.$$

Substituting now Eqs.(B.11,B.12,B.14,B.15) into Eq.(B.6) and normalizing with ζ^2 , we finally obtain the variance of the coefficient as it appears in Eq.(20)

$$\sigma^2(\hat{\gamma}_P) = \frac{(n-2)(2n^4 + 37n^3 + 42n^2 - 45n - 54)}{18n(n-1)(2n^2 - 3)^2}. \quad (\text{B.16})$$

APPENDIX C: CALCULATION OF THE MINIMUM CO-EFFICIENT VALUE $L_{\gamma_P}(n)$

This appendix shows how to calculate the lowest value $L_{\gamma_P}(n) = \hat{\gamma}_P(\xi)$, given that the bounce permutation ξ as defined in Eq.(21) is the minimizing one (for even $n = 2m$

and for some positive integer m). We firstly estimate the quantity $2d_P(\xi, e)$ which is equal to

$$\text{tr}[\mathbf{A}(\xi)\mathbf{A}(e)] = \sum_{i,j} (\xi(i) - \xi(j))^2 (i-j)^2 \quad (\text{C.1})$$

$$= \sum_{i,j} \left(2i^2 \xi(i)^2 + 2j^2 \xi(j)^2 - 4i^2 \xi(i) \xi(j) - 4ij \xi(i)^2 + 4i \xi(i) j \xi(j) \right) \quad (\text{C.2})$$

$$= 2n \underbrace{\sum_i i^2 \xi(i)^2}_{c_4} + 2f_2^2 - 4f_1 \underbrace{\sum_i i^2 \xi(i)}_{c_3} - 4f_1 \underbrace{\sum_i i \xi(i)^2}_{c_2} + 4 \left(\underbrace{\sum_i i \xi(i)}_{c_1} \right)^2. \quad (\text{C.3})$$

We now examine each of the summation terms c_i defined above with respect to the given ξ . Firstly, we assume $\frac{n}{2}$ is even (i.e., that $m = 2k$ for some k). For c_1 we have

$$c_1 = \sum_{i=1}^k \left(2i(m+2i) + (2i-1)(m-2(i-1)) + (n+1-2i)(n+1-m-2i) + (n+1-2i+1)(n+1-m+2(i-1)) \right) \quad (\text{C.4})$$

$$= \sum_{i=1}^k (n^2 + n + 8i - 3) = \frac{n^3 + 2n^2 + n}{4}. \quad (\text{C.5})$$

This is because, it can be seen from the definition of the bounce permutation ξ that it is generated as the sequence

$$[m, m+2, m-2, m+4, m-4, m+6, \dots, m-5, m+5, m-3, m+3, m-1, m+1]^T.$$

Its first m elements can be written concisely as

$$\xi(i) = m + (-1)^i (i - \text{mod}(i, 2)), \quad (\text{C.6})$$

for $i = 1, \dots, m$, or by separating odd and even indices, as

$$\begin{aligned} \xi(2i-1) &= m - 2(i-1), \\ \xi(2i) &= m + 2i, \end{aligned} \quad (\text{C.7})$$

for $i = 1, \dots, k$. Therefore, for the calculation of $c_1 = \sum_i i \xi(i)$, the first two summands in Eq.(C.4) rely directly on Eq.(C.7). For the last two summands, we make use of the observation from Eq.(21) that for each $i = 1, \dots, n$ we have

$$\xi(i) + \xi(n+1-i) = e(i) + e(n+1-i) = n+1, \quad (\text{C.8})$$

that is, adding ξ and e to their flipped versions $\tilde{\xi}$ and \tilde{e} , respectively, yields constant vectors of elements $n+1$.

To calculate c_2 we proceed in similar fashion as

$$\begin{aligned} c_2 &= \sum_{i=1}^k \left(2i(m+2i)^2 + (2i-1)(m-2(i-1))^2 + (n+1-2i)(n+1-m-2i)^2 + (n+1-2i+1)(n+1-m+2(i-1))^2 \right) \\ &= \frac{1}{2} \sum_{i=1}^k \left((n+1)(16i^2 + n^2 - 2) \right) \\ &= \frac{2n^4 + 5n^3 + 4n^2 + n}{12}. \end{aligned} \quad (\text{C.9})$$

Similarly, we can obtain

$$c_3 = \sum_{i=1}^k \left((n+1)(8i^2 - 4in + n^2 + 2n - 1) \right) = c_2, \quad (\text{C.10})$$

and

$$\begin{aligned} c_4 &= \sum_{i=1}^k \left(64i^4 - 32i^3n - 128i^3 + 12i^2n^2 + 64i^2n + 128i^2 \right. \\ &\quad \left. - 2in^3 - 8in^2 - 32in - 56i + \frac{n^4}{2} + \frac{3n^3}{2} + \frac{3n^2}{2} + 5n + 9 \right) \\ &= \frac{17n^5}{160} + \frac{n^4}{3} + \frac{3n^3}{8} + \frac{n^2}{6} + \frac{n}{20}. \end{aligned} \quad (\text{C.11})$$

Finally, substituting c_1, \dots, c_4 within Eq.(C.3) and simplifying we get

$$\begin{aligned} d_P(\xi, e) &= nc_4 + f_2^2 - 4f_1c_2 + 2c_1^2 \\ &= \frac{13n^6 - 20n^4 + 52n^2}{1440}. \end{aligned} \quad (\text{C.12})$$

We now examine the second case where $\frac{n}{2}$ is odd, by setting $m = 2k + 1$. Eq.(C.6) is still valid, but the top part of Eq.(C.7) runs till $k+1$, i.e., for $i = 1, \dots, k+1$. Based on this observation, we proceed in a similar fashion to calculate the corresponding summations of Eq.(C.3), denoted now by \bar{c}_i . For example, for the first term we have

$$\begin{aligned} \bar{c}_1 &= \sum_{i=1}^k \left(2i(m+2i) + (n+1-2i)(n+1-m-2i) \right) \\ &\quad + \sum_{i=1}^{k+1} \left((2i-1)(m-2(i-1)) \right. \\ &\quad \left. + (n+1-2i+1)(n+1-m+2(i-1)) \right) \\ &= c_1 + \frac{1}{2}, \end{aligned} \quad (\text{C.13})$$

where the last value is obtained after simplifications and relies on the expression of c_1 with respect to n in Eq.(C.5). Similarly, for the remaining terms we obtain $\bar{c}_2 = \bar{c}_3 = c_2 + \frac{n+1}{2}$ and $\bar{c}_4 = c_4 + \frac{(n+1)^2}{2}$. Finally, substituting in Eq.(C.3) results in

$$d_P(\xi, e) = \frac{13n^6 - 20n^4 + 52n^2}{1440} + \frac{1}{2}, \quad (\text{C.14})$$

which together with Eq.(C.12) for even $m = \frac{n}{2}$, leads to Eq.(23). To obtain $L_{\gamma_P}(n)$, we calculate $\frac{2d_P(\xi, e)}{\zeta}$ using Eqs.(C.12, C.14) and the normalizing quantity ζ from Eq.(A.7). After simplifying, we obtain the value

$$L_{\gamma_P}(n) = \frac{13n^4 - 20n^2 + 52 + \frac{720}{n^2} \text{mod}(m, 2)}{24(n-1)(n+1)(2n-3)}, \quad (\text{C.15})$$

for even or odd m .

APPENDIX D: PROOF THAT ξ MINIMIZES d_P FOR EVEN n

We now provide a possible proof that $d_P(\xi, e) \leq d_P(\pi, e)$, for all $\pi \in S_n$. Using Eq.(10), we can rewrite d_P as

$$d_P(\tau, \pi) = n\langle \bar{\tau}^2, \bar{\pi}^2 \rangle + 2\langle \bar{\tau}, \bar{\pi} \rangle^2 + \frac{(n^3 - n)^2}{12^2}, \quad (\text{D.1})$$

since $\mu_{\tau\pi}^{(2,2)} = \frac{\langle \bar{\tau}^2, \bar{\pi}^2 \rangle}{n}$ and $\mu_{\tau\pi}^{(1,1)} = \frac{\langle \bar{\tau}, \bar{\pi} \rangle}{n}$. In the above, for simplicity we denote by $\langle \cdot, \cdot \rangle$ the inner product, by τ^2 the vector $\tau \odot \tau$, and by $\bar{\tau}$ the centred version of τ with $\bar{\tau}(i) = \tau(i) - \mu$, where $\mu = \frac{f_1}{n} = \frac{n+1}{2}$ is the mean of the elements of a permutation. Eq.(D.1) can also be obtained by expanding $\frac{1}{2} \text{tr} [\mathbf{A}(\bar{\tau})\mathbf{A}(\bar{\pi})] = \frac{1}{2} \sum_{ij} (\bar{\tau}(i) - \bar{\tau}(j))^2 (\bar{\pi}(i) - \bar{\pi}(j))^2$ as in Eqs.(C.1, C.2), because for any τ we have $\mathbf{A}(\bar{\tau}) = \mathbf{A}(\tau)$.

Now, we will show that ξ minimizes $d_P(\tau, e)$, because it minimizes simultaneously the first two terms in Eq.(D.1); that is

$$\xi = \underset{\tau \in S_n}{\text{argmin}} \langle \bar{\tau}^2, \bar{e}^2 \rangle = \underset{\tau \in S_n}{\text{argmin}} \langle \bar{\tau}, \bar{e} \rangle^2. \quad (\text{D.2})$$

To show the former minimization, we firstly note that since $\bar{e}(i) = \frac{2i-1-n}{2}$, we have

$$\begin{aligned} \bar{e}^2 &= \left[\left(\frac{1-n}{2} \right)^2, \left(\frac{3-n}{2} \right)^2, \dots, \left(\frac{-3}{2} \right)^2, \left(\frac{-1}{2} \right)^2, \right. \\ &\quad \left. \left(\frac{1}{2} \right)^2, \left(\frac{3}{2} \right)^2, \dots, \left(\frac{n-3}{2} \right)^2, \left(\frac{n-1}{2} \right)^2 \right]^T. \end{aligned} \quad (\text{D.3})$$

Also, from Eq.(C.6) and using $\mu = m + \frac{1}{2}$, we have $\bar{\xi}(i) = (-1)^i (i - \text{mod}(i, 2)) - \frac{1}{2}$, for $i = 1, \dots, m$, while from Eq.(C.8), for $i = m+1, \dots, n$, we have $\bar{\xi}(i) = -\bar{\xi}(n+1-i)$ (since $\bar{\xi} + \tilde{\xi} = \xi + \tilde{\xi} - 2\mu \mathbf{1}_n = \mathbf{0}_n$). Therefore, we have

$$\begin{aligned} \bar{\xi}^2 &= \left[\left(\frac{-1}{2} \right)^2, \left(\frac{3}{2} \right)^2, \dots, \left(\frac{3-n}{2} (-1)^m \right)^2, \left(\frac{n-1}{2} (-1)^m \right)^2, \right. \\ &\quad \left. \left(\frac{1-n}{2} (-1)^m \right)^2, \left(\frac{n-3}{2} (-1)^m \right)^2, \dots, \left(\frac{-3}{2} \right)^2, \left(\frac{-1}{2} \right)^2 \right]^T. \end{aligned} \quad (\text{D.4})$$

We can now observe from Eqs.(D.3, D.4), that the vectors \bar{e} and $\bar{\xi}$ are, ignoring signs, circularly shifted versions of each other. Formally, by defining $\sigma_k(i) = \text{mod}(i-1-k, n) + 1$ to be the circular shift permutation that shifts all elements by k positions (rightwards for $k > 0$), we have $\bar{\xi}^2 = \mathbf{P}_{\sigma_m} \bar{e}^2$; this is exemplified in Fig.D.1(b).

Additionally, we can see that both vectors are symmetric about their middle, since $\bar{\xi}^2 = (-\bar{\xi})^2 = \tilde{\xi}^2$, and similarly

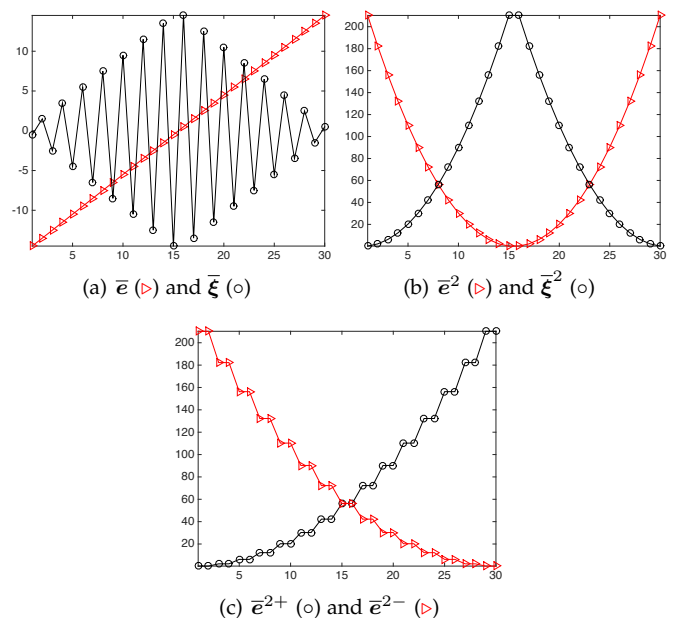


Fig. D.1. Example plots of various transformations of the identity e and the bounce permutation ξ for $n = 30$.

for e . The first m elements of \bar{e}^2 are decreasing, because $(\frac{2i-1-n}{2})^2 > (\frac{2j-1-n}{2})^2$, for $1 \leq i < j \leq m$, and the last m increasing due to the symmetry. On the other hand, because of its shift by m positions, $\bar{\xi}^2$ has the first and last m elements increasing and decreasing, respectively.

Relying on this observation, we can now obtain a sorted version of each of the vectors, by defining a folding permutation $\varphi = [1, n, 2, n-1, \dots, \frac{n}{2}, \frac{n}{2}+1]^T$. Because \bar{e}^2 and $\bar{\xi}^2$ are symmetric, it is easy to see that $P_\varphi \bar{e}^2$ occurs in decreasing order, whereas $P_\varphi \bar{\xi}^2$ in increasing. By using the notations x^+ and x^- to denote two versions of a vector x sorted in ascending and descending order, respectively, we have $P_\varphi \bar{e}^2 = \bar{e}^{2-}$ and $P_\varphi \bar{\xi}^2 = \bar{e}^{2+}$; see the example in Fig.D.1(c).

Finally, because for all permutations $\tau \in S_n$, their corresponding transformations $\bar{\tau}^2$ are composed of the same elements $\{\bar{e}^2(i)\}_{i=1}^n$ occurring in differing positions, we can employ the rearrangement theorem from Eq.(7) to state the inequality

$$\begin{aligned} \langle \bar{\xi}^2, \bar{e}^2 \rangle &= \langle P_\varphi \bar{\xi}^2, P_\varphi \bar{e}^2 \rangle = \langle \bar{e}^{2+}, \bar{e}^{2-} \rangle \\ &\leq \langle \bar{\tau}^2, \bar{\pi}^2 \rangle, \quad \forall \tau, \pi \in S_n. \end{aligned} \quad (D.5)$$

The quantity $\langle \bar{e}^{2+}, \bar{e}^{2-} \rangle$ which is constant and can be shown to be $\frac{n^3+14n}{480}$, is the minimum possible value the inner product can assume. The above establishes that¹ $\xi = \operatorname{argmin}_{\tau \in S_n} \langle \bar{\tau}^2, \bar{e}^2 \rangle$.

To conclude Eq.(D.2) we need to additionally show that $\xi = \operatorname{argmin}_{\tau \in S_n} \langle \bar{\tau}, \bar{e} \rangle^2$. We firstly re-express the squared inner product of any two centred vectors as

$$\langle \bar{\tau}, \bar{\pi} \rangle^2 = \langle \tau, \pi \rangle^2 - 2n\mu^2 \langle \tau, \pi \rangle + n^2\mu^4. \quad (D.6)$$

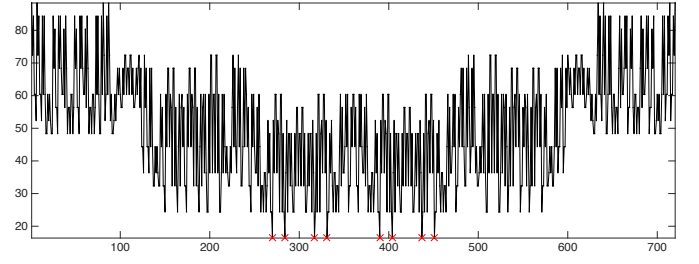
We can then see that $\langle \bar{\xi}, \bar{e} \rangle^2$ can be calculated by re-using the term $\langle \xi, e \rangle = c_1 = \frac{n^3+2n^2+n}{4}$ from Eq.(C.5), assuming for the moment an even $m = \frac{n}{2}$. By substituting in Eq.(D.6) we have

$$\langle \bar{\xi}, \bar{e} \rangle^2 = c_1^2 - 2n\mu^2 c_1 + n^2\mu^4. \quad (D.7)$$

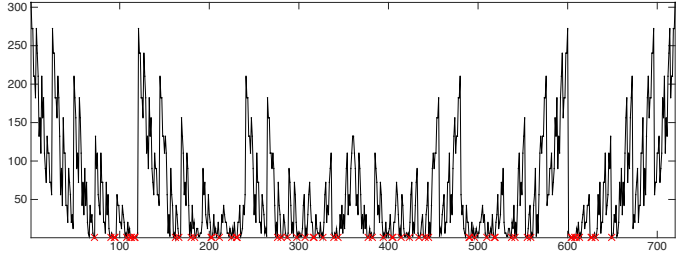
The above expression is, however, independent to n as it simplifies to zero, which is the smallest possible value the non-negative quantity $\langle \bar{\tau}, \bar{\pi} \rangle^2$ can assume.

For the case now where m is odd, from Eq.(C.13) we have $\langle \xi, e \rangle = \bar{c}_1 = \frac{n^3+2n^2+n+2}{4}$. Substituting as above, this leads to $\langle \bar{\xi}, \bar{e} \rangle^2 = \frac{1}{4}$. This value is also the minimum possible value $\langle \bar{\tau}, \bar{e} \rangle^2$ can assume. To see this, we need to examine the minimum of the quadratic in Eq.(D.6). This occurs at $\delta = n\mu^2 = \frac{n(n+1)^2}{4}$. In the previous case, where m is even and we have $n = 4k$ for some k , this quantity becomes $16k^3+8k^2+k$, which is an integer. When m is odd, however, and we have $n = 4k+2$, it becomes $\delta = 16k^3+32k^2+21k+$

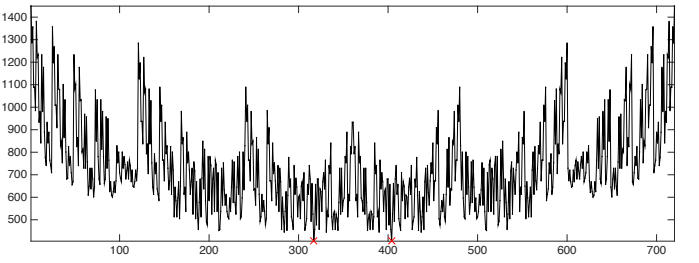
1. ξ is not a unique minimizer of $\langle \bar{\tau}^2, \bar{e}^2 \rangle$. From its definition, it can be seen that ξ is composed by a half-length shift σ_m , followed by further swaps between selected symmetric elements. Formally, if we define ζ_k to be the permutation with elements $\zeta_k(k) = n+1-k$, $\zeta_k(n+1-k) = k$ and $\zeta_k(i) = i$ for the remaining elements, then $\xi = \sigma_m \zeta_1 \zeta_3 \zeta_5 \dots \zeta_{m+\operatorname{mod}(m,2)-1}$. For $\langle \bar{\tau}^2, \bar{e}^2 \rangle$ however, all possible 2^m permutations created by the composition of σ_m with arbitrary sequences of ζ_k compositions are also its minimizers (see Fig.D.2(a)), since $\sigma_m \zeta_k^2 = \bar{\sigma}_m^2 = \bar{\xi}^2$.



(a) $\langle \bar{\tau}^2, \bar{e}^2 \rangle$



(b) $\langle \bar{\tau}, \bar{e} \rangle^2$



(c) $d_P(\tau, e)$

Fig. D.2. Example plots of the minimizing quantities of interest and for all possible permutations $\tau \in S_6$ (generated from the factoradic of the index values $1, \dots, n!$ in the horizontal axes). All the minima (x) are marked at the bottom of each plot.

$4 + \frac{1}{2}$, which is not an integer for the inner product $\langle \tau, e \rangle$ to assume and make Eq.(D.7) zero.

Nevertheless, it is easy to see that there are always two permutations whose inner product assumes every possible integer value from within the range $\Delta = [\langle e^-, e^+ \rangle, \langle e^+, e^+ \rangle] = [(n+1)f_1 - f_2, f_2]$; in fact, δ is always in the middle of Δ . Therefore, the minimum value of the quadratic $\langle \bar{\xi}, \bar{e} \rangle^2$ must occur in any of the integer values immediately adjacent to δ , that is $\delta \pm \frac{1}{2}$. Substituting these into Eq.(D.6) and simplifying, gives $\langle \bar{\tau}, \bar{\pi} \rangle^2 = \frac{1}{4}$ for both sides, which completes the proof that² $\xi = \operatorname{argmin}_{\tau \in S_n} \langle \bar{\tau}, \bar{e} \rangle^2$.

From the above, Eq.(D.2) holds and via Eq.(D.1) ξ is shown to minimize d_P and γ_P for even values of n . We have not investigated the case for odd n and its associated

2. Again, ξ is not a unique minimizer of $\langle \bar{\tau}, \bar{e} \rangle^2$, because any permutation τ such that $\langle \tau, e \rangle = \frac{n(n+1)^2}{4}$ for even m (offset by $\pm \frac{1}{2}$ for odd m) is also a minimizer. Fig.D.2(b) exemplifies this by showing the many possible minima for a specific n . In general, for even n and as demonstrated in the example of Fig.D.2(c), the set of minimizers of d_P and γ_P is the intersection of the sets of minimizers of $\langle \bar{\tau}^2, \bar{e}^2 \rangle$ and $\langle \bar{\tau}, \bar{e} \rangle^2$. This includes at least two permutations presented in this work, that is ξ and $\bar{\xi}$, but it should be noted that there can be additional ones.

$L_{\gamma_P}(n)$, but it can be analyzed similarly³.

APPENDIX E: ADDITIONAL EXPERIMENTATION AND RESULTS

In this appendix, we provide additional application examples for the proposed contributions using data from a domain other than bioinformatics (given in Section 4 of the manuscript).

E.1 Comparing Different Seriation Algorithms

This section presents experiments with eight datasets from the text-mining domain, where document samples are represented through word frequencies. Four of these datasets (CT3, CT4, CT6 and CT9) are extracted from a large clinical trial collection [3] where topics are based on queries such as asthma, breast cancer, lung cancer, prostate cancer, cardiovascular, HIV, leukemia, depression, and schizophrenia. Each of the four datasets contains 500 randomly selected clinical trial samples from 3, 4, 6 and 9 topics, respectively. The remaining four datasets (R2, R5, R8 and R12) are from the Reuters-21578 Text Categorisation Test Collection [4] containing articles from the Reuters newswire. They were originally annotated with topics such as earn, trade, money-fx, interest, ship, sugar, money-supply, coffee. Each of the four datasets contains 500 randomly selected articles from 2, 5, 8, and 12 topics, respectively. All distance matrices are computed from a second-order similarity measure relying on the cosine similarity between inner-products of articles, which reflects second-order links between articles via word co-occurrences.

Table E.1 summarizes the proximity coefficient γ_P values averaged across the eight datasets for the sixteen algorithms described in Section 4.1, and also the respective standard deviations. As can be seen, for this domain's datasets the algorithm pairs with the highest similarities are {MDS1, PCA}, {MDS1, SS2} and {PCA, SS2}, with average γ_P values of 0.942, 0.942 and 0.985, respectively. The corresponding dendrogram constructed from Table E.1 is shown in Fig.E.1. The most similar algorithms {MDS1, SS2, PCA} are shown to form a tight cluster, since all pairwise coefficients are over 0.94. Other groups show to be less similar, such as the cluster {OLO, R2E, CO} with pairwise similarities around 0.73. Overall, compared to the bioinformatics datasets in

3. It has to be noted, that for this case the corresponding ξ has a different pattern due to the lack of symmetry. However, the analysis can be similar. For example, we can verify that $\langle \bar{\tau}^2, \bar{e}^2 \rangle$ also assumes the minimum value of $\langle \bar{e}^{2+}, \bar{e}^{2-} \rangle$ which, for the associated \bar{e}^2 , is now equal to $\frac{n^5 + 10n^3 - 11n}{480}$. Also, the quantity $\langle \bar{\tau}, \bar{e} \rangle^2$ can decrease down to zero (for $n > 3$), because for both cases of $n = 4k + 1$ and $n = 4k + 3$, $\langle \bar{\tau}, \bar{\pi} \rangle$ assumes an integer value at $\delta \in \Delta$. It is straightforward to find that when both of these quantities have common minimizers, then from Eq.(D.1) we have $\min_{\tau, \pi \in S_n} d_P(\tau, \pi) = \frac{13n^6 + 10n^4 - 23n^2}{1440}$ and the positional proximity coefficient has lower bound $L_{\gamma_P}(n) = \frac{13n^2 + 23}{24(2n^2 - 3)}$ with the same limit of $\frac{13}{48}$ as the even case. However, for certain cases (e.g., $n = 5$ or 11), the quantities $\langle \bar{\tau}^2, \bar{e}^2 \rangle$ and $\langle \bar{\tau}, \bar{e} \rangle^2$ are not minimized simultaneously by any single permutation. In such cases, a correction offset of +2 needs to be added to the minimum value of d_P , to account for the fact that the least suboptimal value of $\langle \bar{\tau}, \bar{e} \rangle^2$, where $\langle \bar{\tau}^2, \bar{e}^2 \rangle$ is minimized, is exactly 1. In general, for the case of odd n , we have empirically observed that d_P has many more minimizing permutations than the even case.

Section 4, different similarity patterns arise here between algorithm pairs. This is to be expected, as different domains and different ways of estimating object (dis)similarities, can have diverse effects in the operation of each algorithm. In general, it is reasonable to expect that within a particular domain a group of datasets is much more likely to exhibit similar ordering patterns by certain algorithms, than for datasets from different domains.

Fig.E.2 displays various ordered distance maps for the qualitative comparison between similar and dissimilar sequences. It shows the image maps of two similar algorithms PCA and SS2, along with the two dissimilar ones SPIN2 and CO for a selected dataset. Visually, the map of PCA is almost identical to a rotated version of the map of SS2, with the largest block in the top left rather than the bottom right. This is related to the fact that in Table E.1 these two algorithms show to be on average very similar ($\gamma_P = 0.985$). The four blocks in the maps correspond to the four main document topics of the dataset CT4. The map of CO also identifies four blocks as PCA, but in a different order and sharpness (average γ_P between CO and PCA is 0.702). SPIN2 identifies multiple sample concentrations and varies significantly from PCA, SS2 and CO (with corresponding average γ_P values in Table E.1 of 0.403, 0.402 and 0.411).

Fig.E.3 presents a further example for four algorithms and a different dataset. The images of CO and R2E show relatively similar when rotated ($\gamma_P = 0.746$). The maps of HP and SA, although they have also identified the three main document concentrations of CT3, appear with very different sample orderings within each block ($\gamma_P = 0.574$). Also, within each block of the HP map, more local structure has been identified compared to the more gradually varying global structure captured in the blocks of the other methods.

Another example is given in Fig.E.4 using dataset R5. The maps of MDS1 and OLO show relatively similar when rotated (with average $\gamma_P = 0.741$), both having a large and

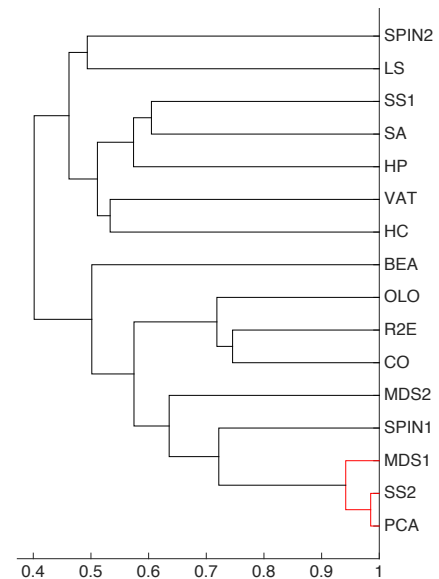
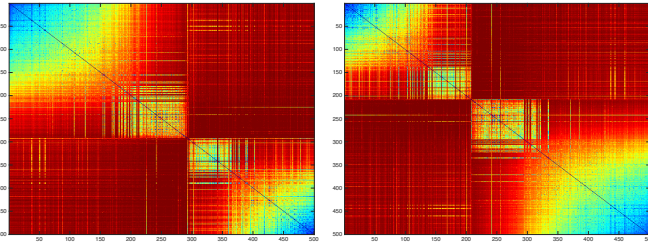


Fig. E.1. Dendrogram (of cophenetic correlation coefficient = 0.86) computed using complete linkage and the γ_P values from Table E.1. Groups with linkage values (shown on the horizontal axis) of 0.8 and over are drawn in different shades.

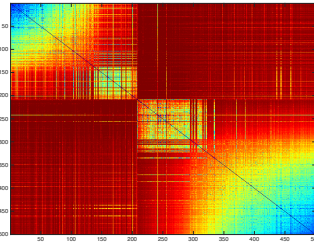
TABLE E.1

Comparisons between the 16 evaluated algorithms based on the proximity coefficient γ_P . The upper triangle of the table contains the mean γ_P taken across all 8 text datasets (values ≥ 0.8 are boldfaced), whereas the lower triangle displays the corresponding standard deviations.

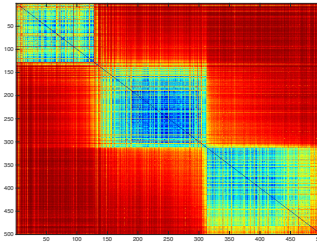
	BEA	CO	HC	OLO	HP	LS	MDS1	MDS2	PCA	R2E	SA	SPIN1	SPIN2	SS1	SS2	VAT
BEA	•	0.503	0.500	0.569	0.488	0.450	0.570	0.501	0.579	0.526	0.503	0.520	0.448	0.486	0.582	0.428
CO	0.083	•	0.525	0.718	0.527	0.467	0.741	0.627	0.702	0.746	0.600	0.585	0.411	0.521	0.703	0.496
HC	0.112	0.065	•	0.554	0.511	0.514	0.490	0.490	0.512	0.551	0.603	0.510	0.502	0.540	0.517	0.533
OLO	0.140	0.153	0.099	•	0.584	0.471	0.741	0.632	0.699	0.735	0.652	0.614	0.457	0.563	0.699	0.503
HP	0.151	0.139	0.127	0.171	•	0.487	0.544	0.496	0.508	0.580	0.574	0.518	0.478	0.587	0.504	0.524
LS	0.052	0.055	0.061	0.074	0.047	•	0.475	0.466	0.476	0.494	0.547	0.496	0.494	0.485	0.474	0.462
MDS1	0.146	0.157	0.050	0.095	0.133	0.095	•	0.733	0.942	0.714	0.605	0.722	0.403	0.526	0.942	0.544
MDS2	0.134	0.147	0.063	0.117	0.089	0.061	0.272	•	0.719	0.594	0.562	0.636	0.454	0.519	0.705	0.534
PCA	0.143	0.183	0.079	0.141	0.107	0.093	0.103	0.295	•	0.665	0.569	0.741	0.403	0.490	0.985	0.522
R2E	0.129	0.222	0.073	0.110	0.157	0.069	0.099	0.117	0.119	•	0.717	0.575	0.469	0.569	0.670	0.518
SA	0.130	0.169	0.093	0.159	0.162	0.083	0.094	0.065	0.077	0.163	•	0.529	0.540	0.605	0.565	0.540
SPIN1	0.110	0.165	0.075	0.157	0.094	0.123	0.225	0.239	0.207	0.143	0.083	•	0.509	0.508	0.738	0.465
SPIN2	0.104	0.055	0.143	0.108	0.085	0.063	0.033	0.073	0.044	0.107	0.120	0.109	•	0.515	0.402	0.483
SS1	0.129	0.191	0.111	0.207	0.171	0.055	0.175	0.126	0.137	0.240	0.219	0.146	0.130	•	0.482	0.537
SS2	0.141	0.187	0.084	0.141	0.107	0.091	0.110	0.288	0.022	0.116	0.075	0.204	0.048	0.140	•	0.507
VAT	0.094	0.097	0.138	0.094	0.076	0.052	0.051	0.035	0.045	0.116	0.104	0.094	0.051	0.104	0.053	•



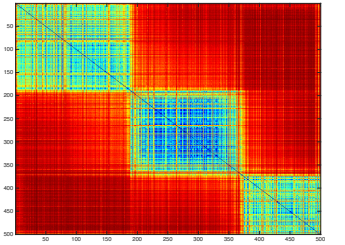
(a) PCA



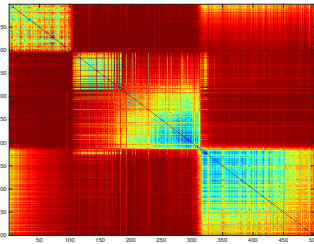
(b) SS2



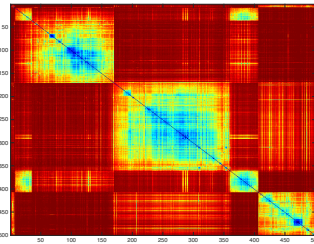
(a) CO



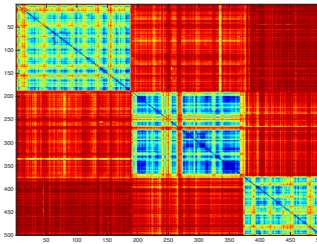
(b) R2E



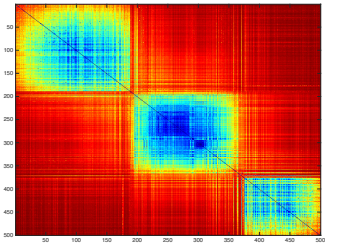
(c) CO



(d) SPIN2



(c) HP



(d) SA

Fig. E.2. Dissimilarity matrices of the CT4 dataset, ordered using the output from four different algorithms.

Fig. E.3. Dissimilarity matrices of the CT3 dataset, ordered using the output from four different algorithms.

a smaller block in either corner and a lot of fragmentation in the remaining regions. Although the maps of HP and SS1 also identify two main blocks, they vary considerably ($\gamma_P = 0.587$) in the overall order of the blocks and the object distributions within them.

E.2 Consensus Seriation Experiments

In this section, we use some datasets from Section D.1, to compare the consensus seriation results quantitatively and qualitatively. The five subsets used for the analysis are $T_1 = \{HC, MDS1, R2E\}$, $T_2 = \{MDS2, OLO, SS1\}$, $T_3 = \{HC, LS, OLO, R2E, SS2\}$, $T_4 = \{CO, HC, MDS2, OLO, PCA\}$, and $T_5 = \{LS, MDS1, MDS2, OLO, R2E, SPIN1, SS1\}$. The individual members of each subset are mostly chosen randomly, but more dissimilar pairs and smaller overlap is preferred for the shorter subsets. For example, the most similar pair in T_1 and T_2 is the $\{MDS1, R2E\}$ with average $\gamma_P = 0.714$, while the remaining pairs have $\gamma_P \leq 0.632$. Larger subsets are allowed to have more similar members, but not over 0.8.

Fig.E.5 displays the ARGAR errors for each subset, calculated for the proposed method, the existing consensus methods Hamming, Spearman and Kendall, and the average of the individual algorithms in each subset. Each of these five subsets is also applied to the four different datasets CT3, CT4, R5 and R12, chosen to cover a wide base of the data. It can be seen, that in over fifteen out of the total of twenty experiments, the proposed consensus method performs better than the average of the individual components. Also, it always outperforms Hamming, and the Spearman and Kendall methods in all but three cases.

In Fig.E.6, the maps of the three individual algorithms in $T_1 = \{HC, MDS1, R2E\}$ applied to dataset CT4 are shown, together with the map of the proposed consensus method. The map of HC seems to identify blocks with apparent local structure within each block. Both MDS1 and R2E also have four visible blocks, but within each block more of the global structure is captured. The consensus map shows to be different from the contributing three maps. The blocks are not as fragmented as in HC, but also not as smoothly

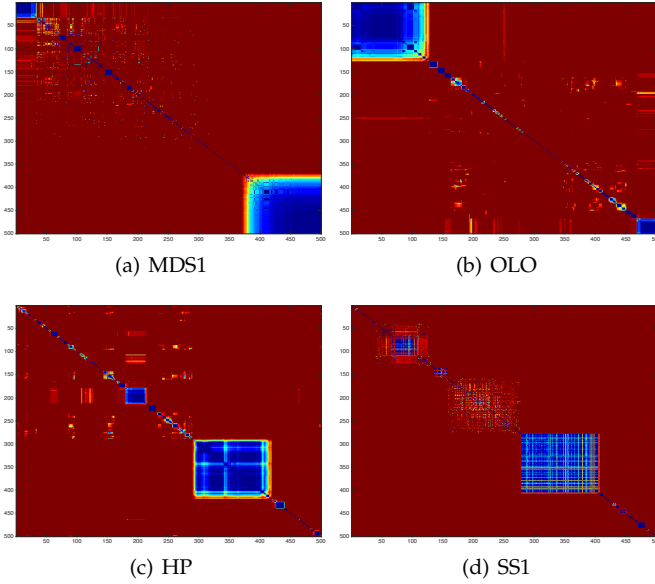


Fig. E.4. Dissimilarity matrices of the R5 dataset, ordered using the output from four different algorithms.

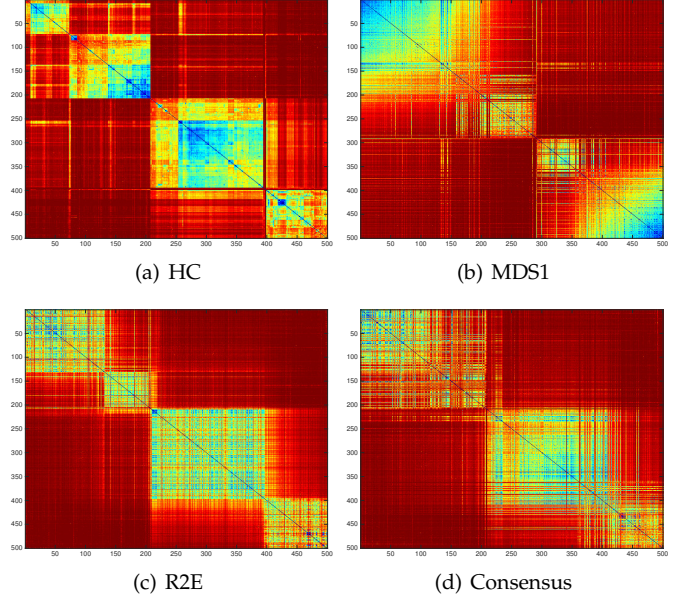


Fig. E.6. Seriated maps for the individual algorithms of subset T_1 as well as the proposed consensus method, for the dataset CT4.

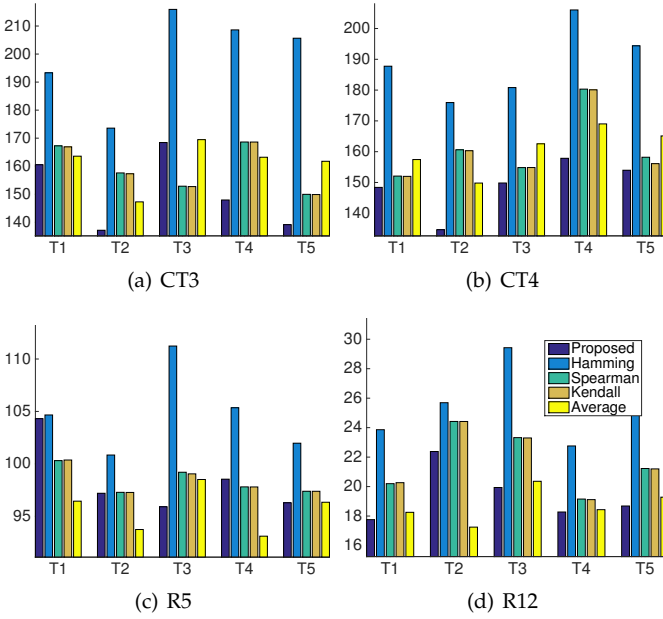


Fig. E.5. ARGAR errors for consensus seriation results obtained with the proposed, Hamming, Spearman and Kendall measures showing also the average of the original algorithms comprising each T_i .

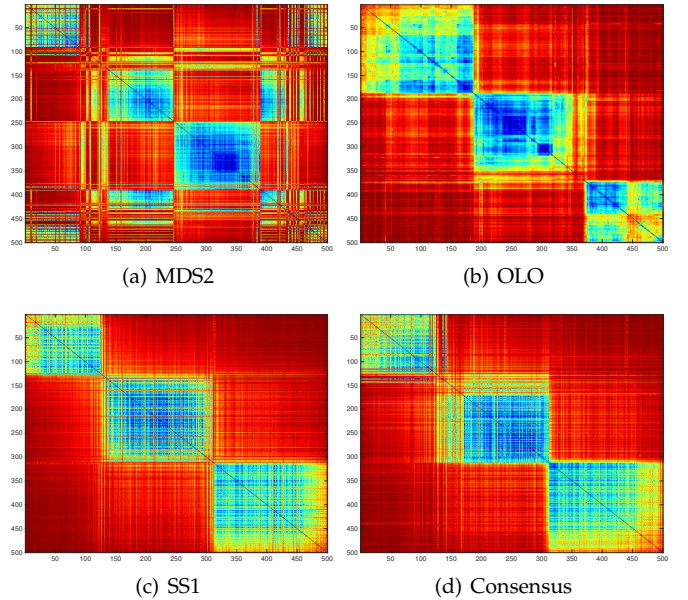


Fig. E.7. Seriated maps for the individual algorithms of subset T_2 as well as the proposed consensus method, for the dataset CT3.

varying as some regions of the MDS1 and R2E maps.

Fig.E.7 is another such example and displays the maps of the individual algorithms in $T_2 = \{\text{MDS2, OLO, SS1}\}$ and the proposed method applied to dataset CT3. The map of MDS2 shows a lot of fragmentation of the three main topics. OLO and SS1 capture the main blocks, with OLO showing more local structure within blocks than SS1. The consensus map is more similar to that of SS1, but it exhibits more local within-block structure and less sharp transition between the top two blocks of SS1.

A third example is given in Fig.E.8 that contains the maps for the T_3 member algorithms and their consensus, for the R2 dataset. All methods seem to identify one of the two

document topics with partial weaker concentrations further away. The map of the combined sequence shows to have the main block defined sharper than LS and R2E, without the very strong local structure within the block in HC, OLO and SS2. Overall, it appears to be different from all five contributing methods.

Finally, Fig.E.9 plots the full range of RGAR errors against the window sizes, for the original MDS2, OLO and SS1 algorithms in T_2 , the consensus methods Hamming, Spearman and Kendall and also the proposed one, applied to the CT3 dataset. It can be seen, that OLO starts off with a very low RGAR error of 0.28 and rapidly increases to 0.4 at $\delta = 27$. No other algorithm or consensus method outperforms OLO for small windows due to its local struc-

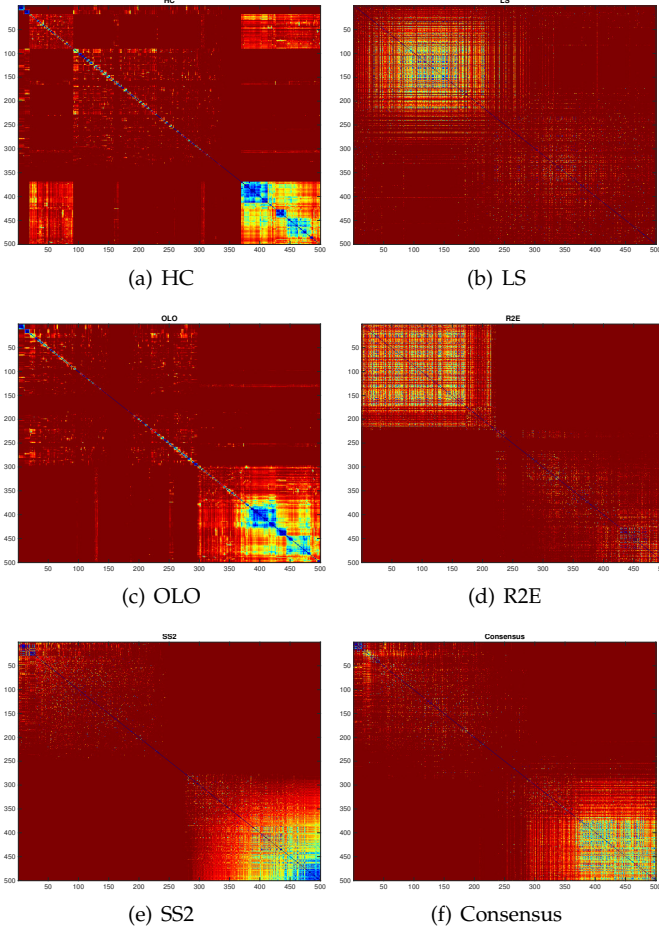


Fig. E.8. Seriated maps for the individual algorithms of subset T_3 as well as the proposed consensus method, for the dataset R2.

ture concentration through its leaf ordering. SS1 seems to work better on global structure, since it outperforms all algorithms and gradually reaches the smallest possible error. MDS2 works better in the early to middle structure region, while subsequently fluctuates. The proposed consensus starts off with a high RGAR error of 0.53, and thereafter follows a gradual decrease to a minimum of 0.18. Although in this dataset, it does not outperform the contributing SS1, it shows that on average it keeps a balance between SS1 and the other two contributors MDS2 and OLO, and it does not get affected by the high errors of MDS2. For the initial ranges, however, it shows to be worse than all three contributors, which can be owed to being unable to reconcile the ordering information at local scales from the contributing sequences. This could be related to the strong class separability of the documents in these datasets to different topics, which gives rise to the strong blocky structures visible in the maps. Fig.E.9 also shows that Hamming, Spearman and Kendall have the highest error for the majority of window ranges, apart from MDS2 for the second half of the window range.

REFERENCES

- [1] D. E. Knuth, "Johann Faulhaber and sums of powers," *Mathematics of Computation*, vol. 61, no. 203, pp. 277–294, 1993.

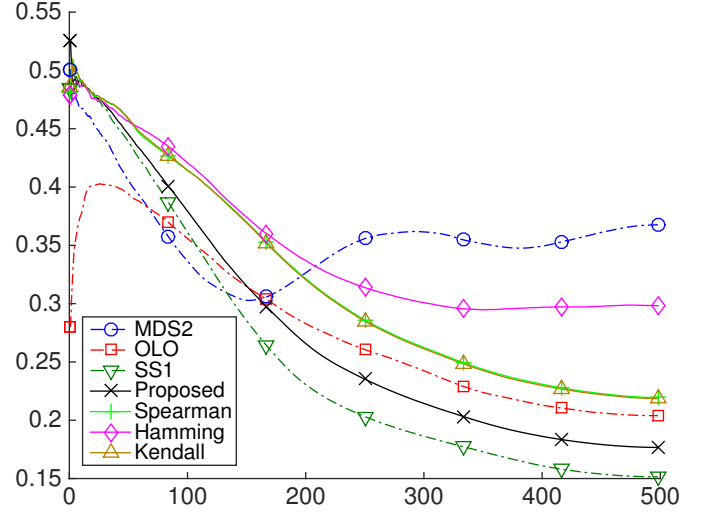


Fig. E.9. RGAR errors for the algorithms in T_2 and all consensus methods plotted against all window sizes δ .

- [2] L. Hubert and J. Schultz, "Quadratic assignment as a general data analysis strategy," *British Journal of Mathematical and Statistical Psychology*, vol. 29, no. 2, pp. 190–241, 1976.
- [3] T. Mu, J. Y. Goulermas, I. Korkontzelos, and S. Ananiadou, "Descriptive document clustering via discriminant learning in a co-embedded space of multilevel similarities," *Journal of the Association for Information Science and Technology*, in press (DOI=10.1002/asi.23374).
- [4] D. D. Lewis, "Reuters-21578 text categorization test collection," 1997. [Online]. Available: <http://www.daviddlewis.com/resources/testcollections/reuters21578>