computation

MDPI

Article

Weighted Consensus Segmentations

Halima Saker [1,2], Rainer Machné [3], Jörg Fallmann [1], Douglas B. Murray [4,5], Ahmad M. Shahin [2] and Peter F. Stadler [1,6,7,8,9,*]

1 Bioinformatics Group, Department of Computer Science, and Interdisciplinary Center for Bioinformatics, Universität Leipzig, D-04107 Leipzig, Germany; halima@bioinf.uni-leipzig.de (H.S.); fall@bioinf.uni-leipzig.de (J.F.)
2 Doctoral School of Science and Technology, Lebanese University, Tripoli, Lebanon; ashahin@ul.edu.lb
3 Institute for Synthetic Microbiology and Institute for Quantitative and Theoretical Biology, Heinrich Heine University, D-40225 Düsseldorf, Germany; machne@hhu.de
4 Lakeland University Japan Shinjuku-ku, Tokyo 160-0022, Japan; mabawsa@gmail.com
5 University of Maryland Global Campus—Asia, Yokota Air Base, Fussa-shi, Tokyo 197-0001, Japan
6 German Centre for Integrative Biodiversity Research (iDiv) Halle-Jena-Leipzig, Competence Center for Scalable Data Services and Solutions, Leipzig Research Center for Civilization Diseases, and Leipzig Research Center for Civilization Diseases (LIFE), Leipzig University, D-04103 Leipzig, Germany
7 Institute for Theoretical Chemistry, University of Vienna, A-1090 Wien, Austria
8 Facultad de Ciencias, Universidad National de Colombia, Bogotá CO-111321, Colombia
9 Santa Fe Institute, Santa Fe, NM 87501, USA
* Correspondence: studla@bioinf.uni-leipzig.de

Abstract: The problem of segmenting linearly ordered data is frequently encountered in time-series analysis, computational biology, and natural language processing. Segmentations obtained independently from replicate data sets or from the same data with different methods or parameter settings pose the problem of computing an aggregate or consensus segmentation. This SEGMENTATION AGGREGATION problem amounts to finding a segmentation that minimizes the sum of distances to the input segmentations. It is again a segmentation problem and can be solved by dynamic programming. The aim of this contribution is (1) to gain a better mathematical understanding of the SEGMENTATION AGGREGATION problem and its solutions and (2) to demonstrate that consensus segmentations have useful applications. Extending previously known results we show that for a large class of distance functions only breakpoints present in at least one input segmentation appear in the consensus segmentation. Furthermore, we derive a bound on the size of consensus segments. As show-case applications, we investigate a yeast transcriptome and show that consensus segments provide a robust means of identifying transcriptomic units. This approach is particularly suited for dense transcriptomes with polycistronic transcripts, operons, or a lack of separation between transcripts. As a second application, we demonstrate that consensus segmentations can be used to robustly identify growth regimes from sets of replicate growth curves.

Keywords: segmentation aggregation; consensus segmentation; boundedly convex functions; dynamic programming; yeast transcriptome; microbial growth curves

Citation: Saker, H.; Machné, R.; Fallmann, J.; Murray, D.B.; Shahin, A.M.; Stadler, P.F. Weighted Consensus Segmentations. *Computation* **2021**, *9*, 17. https://doi.org/10.3390/computation9020017

Academic Editor: Shizuka Uchida
Received: 31 December 2020
Accepted: 27 January 2021
Published: 5 February 2021

1. Introduction

One-dimensional segmentation problems naturally appear in time series analysis across diverse application areas (often referred to as change point or jump point detection in this context). In computational biology, 1D-segmentation problems arise in the analysis of micro-array and high-throughput DNA or RNA sequencing data. Copy number variations in genomes, for example, can be detected by segmenting array-CGH or DNA coverage data into piece-wise constant segments; see Reference [1] for a review. In epigenomics, recurrent patterns of histone modifications define genomic intervals that can be associated with functional units including promoters, enhancers, or gene bodies, see, e.g., Reference [2] and the references therein. The identification of transcriptional units is

also a segmentation problem, consisting in the distinction of expressed and non-expressed loci [3,4] or operons [5] or, more generally, in the distinction of adjacent or even overlapping transcripts without the benefit of non-expressed spacers between them. The latter task is particularly relevant in organisms with "compact" genomes, such as bacteria [5] or yeast [6,7], where transcribed loci are rarely separated non-expressed regions. Boundaries between transcriptional units are detectable by differences in RNA levels [6], see, e.g., the SRG1 ncRNA in Figure 1. A plethora of segmentation algorithms for genomic features, as well as time series data, have become available, reviewed and benchmarked, e.g., in Reference [8–10].

The boundaries between segments typically are not clearly visible in individual data tracks. Such limitations can often be alleviated by aggregating multiple experiments or measurements. In the yeast RNA-seq data shown in Figure 1, for instance, transcriptome samples at different time points of the respiratory cycle are aggregated to produce a more informative signal than just the RNA expression level at a single time point. Still, any particular choice of parameters (here the choice of the similarity measure for the temporal coverage profiles at adjacent nucleotides), produce both false positive and false negative segment boundaries.

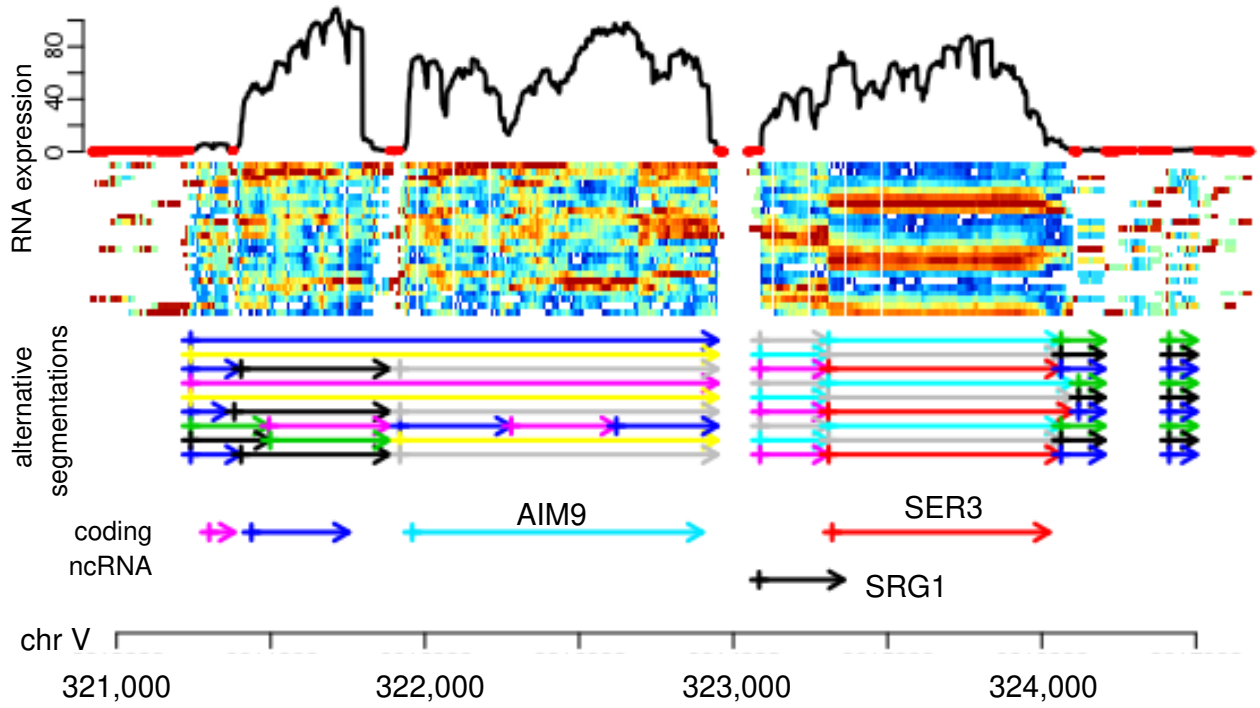


Figure 1. Segmentation of RNA expression patterns. Top: RNA expression in 24 samples taken every 4 min from *Saccharomyces cerevisae* strain IFO 0233 (shown as color scale, with the total of all experiments shown above). The sequencing data are strand-specific, only the plus strand of about 5000 nt on chromosome V are shown. Middle: nine different segmentations computed with `segmenTier` using different parameter settings; see Reference [11] for details on data and segmentations. Below: annotated coding and non-coding yeast genes. The rightmost block of short segments is a candidate for an unannotated ncRNA located anti-sense to the much longer protein-coding gene Utp7p on the minus strand. The data are the same as those in Figure 3 of Reference [11].

Figure 1 suggests that an improvement could be achieved by aggregating the different segmentations to a single consensus. In a similar vein, Reference [4] employed a simple heuristic segmentation to detect candidate loci as intervals that are scored by a statistical model for each RNA-seq experiment, followed by a problem-specific greedy heuristic to determine consensus interval boundaries for expressed ncRNA loci. The segmentation method for bacterial RNA-seq data in Reference [5] computes optimal segmentations with different numbers K of segments and uses a voting procedure to obtain a consensus over

different values of K. These examples beg the question whether there is a more principled way to aggregated segmentations in a single consensus segmentation.

This question also appears in a much more general context. Modern *omics studies often report their results in the form of genome browser tracks, i.e., as segmentation of the reference genome into intervals. The comparison and consolidation of such data naturally asks for a consensus or reference. This is particularly the case for annotation based on epigenomic or transcriptomic data. Here, principled, efficient methods, to compare annotations beyond quantifying overlaps would be highly desirable to avoid a complete reanalysis of the underlying raw data. In contrast to genome browser tracks, raw data typically require extensive processing and are by no means straightforward to access in all cases. Despite the obvious potential usefulness of consensus segmentations, the literature on systematic comparisons of segmentations is surprisingly sparse. Two natural ways to approach the problem have been considered:

(i) Focusing on the breakpoints between segments, one can treat them as signal. Significant (consensus) breakpoints are then detectable as unexpected accumulations across multiple data sets by the `C-KS` algorithm [12]. Somewhat more generally, this can be seen as clustering problem for breakpoints [13].

(ii) Segmentations of linearly ordered data form partitions of an interval. (Dis)similarity measures for partitions, such as the Rand [14], Fowlkes-Mallows [15], Jaccard [16], and Hubert-Arabie [17] indices, or the Mirkin [18], and Van Dongen [19] metrics, are also applicable to the special case of segmentations. The MEDIAN PARTITION problem, also known as `Consensus Clustering` Problem, consists in finding a partition that is as close as possible to a given collection w.r.t. to one of these (dis)similarity measures [20,21]. MEDIAN PARTITION is NP-complete [22,23]. It is this second approach that we pursue further in this contribution.

The SEGMENTATION AGGREGATION problem [24] is the specialization to partitions of a linearly ordered set, such as a time series or genomic sequence: Given a set of m segmentations $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_m$ on an interval, and a distance function D between segmentations, the task is to find a segmentation $\mathbf{C}$ that minimizes the distance sum

$$f(\mathbf{C}) := \sum_{q=1}^{m} D(\mathbf{C}, \mathbf{S}_q)\,. \tag{1}$$

We assume that $D(\,.\,,\,.\,)$ is a dissimilarity, i.e., that (i) $D(\mathbf{S}, \mathbf{S}') \geq 0$, and (ii) $D(\mathbf{S}, \mathbf{S}') = 0$ **if** $\mathbf{S} = \mathbf{S}'$. In most cases, $D(\,.\,,\,.\,)$ will be a metric. However, neither symmetry nor the triangle inequality are necessary. It is shown in Reference [24] that SEGMENTATION AGGREGATION, in contrast to MEDIAN PARTITION in the general case, can be solved exactly by dynamic programming for several interesting distance measures, including the disagreement or Mirkin metric and the information distance. Mailă [25] showed, using an axiomatic approach based on certain additivity conditions in the lattice of partitions, that the "variation of information" [26], i.e., the information distance [24] serves as the essentially unique natural distance between partitions. Nevertheless we consider here a much broader class of dissimilarity measures. Despite its appealing features, namely the almost complete absence of model assumptions and the fact that no detailed knowledge on the provenance of the input segmentations is required, SEGMENTATION AGGREGATION has rarely been used in practical data analysis. Here, we demonstrate that it can be a useful and efficient approach.

A given set of input data is frequently subject to biases, such as an uneven phylogenetic distribution of taxa in comparative genomics or unbalanced distributions of samples between treatment groups. For the purpose of consensus formation, it is usually desirable to retain all data. As a remedy for sampling biases, a plethora of weighting schemes have been proposed to correct for biases by giving larger weights to underrepresented and smaller weights to over-represented data; see Reference [27] for a comparison of different approaches. In the context of segmentations of genomic features or time series, the confi-

dence in individual segmentations $\mathbf{S}_q$ may be different, e.g., due to different noise levels in individual data tracks. It may also be desirable to treat biological replicates different from technical replicates. Naturally, such differences can be expressed by introducing segmentation-specific weights w_q. We shall see below that such weights can be introduced in SEGMENTATION AGGREGATION in a straightforward manner.

In this contribution, we investigate the weighted version of the Segmentation Aggregation problem with the aim of getting insights into the properties of consensus segmentations. In particular, we generalize previous results of the positioning of consensus break points and we derive an upper bound on length of consensus segments for a large class of distance functions. We then consider two very different applications of consensus segmentations: the identification of transcriptional units, using yeast transcriptomes as a show-case example, and the segmentation of microbial growth curves. We close with a brief discussion of several open problems both regarding the theory behind consensus segmentations and their practical applications.

2. Theory

2.1. Dynamic Programming Algorithm

The WEIGHTED SEGMENTATION AGGREGATION problem is a moderate generalization of the unweighted version considered in Reference [24]. Given a set $\{\mathbf{S}_q | 1 \leq q \leq m\}$ of input segmentations and corresponding weights $w_q > 0$ that quantify the relative importance of the contributing segmentations S_q, the task is to minimize the objective function

$$f(\mathbf{C}) := \sum_{q=1}^{m} w_q D(\mathbf{C}, \mathbf{S}_q)\,, \tag{2}$$

i.e., the weighted total dissimilarity of the unknown consensus segmentation $\mathbf{C}$. Without loosing generality we may assume that $\sum_{q=1}^{m} w_q = 1$.

Following Reference [24], we consider a distance measure D that can be expressed in terms of the *common refinement* $\mathbf{S}' \wedge \mathbf{S}'' := \{A \cap B | A \in \mathbf{S}',\, B \in \mathbf{S}''\}$ of two segmentations. $\mathbf{S}' \wedge \mathbf{S}''$ consists of all intersections of the segments of $\mathbf{S}'$ and $\mathbf{S}''$. The common refinement is also known as the *union segmentation* since its set of segment boundaries is exactly the union of the boundaries of $\mathbf{S}'$ and $\mathbf{S}''$. In particular, therefore, $\mathbf{S} \wedge \mathbf{S} = \mathbf{S}$. Now, define the *potential* of a segmentation $\mathbf{S}$ as

$$E(\mathbf{S}) = \sum_{A \in \mathbf{S}} \mathfrak{e}(A)\,, \tag{3}$$

where $\mathfrak{e}$ is a potential function evaluating the individual segments. This gives rise to a class of distances between segmentations defined by

$$D(\mathbf{S}', \mathbf{S}'') = E(\mathbf{S}') + E(\mathbf{S}'') - 2E(\mathbf{S}' \wedge \mathbf{S}''). \tag{4}$$

Substituting for D in Equation (2) yields

$$f(\mathbf{C}) = \sum_{q=1}^{m} w_q E(\mathbf{C}) + \sum_{q=1}^{m} w_q E(\mathbf{S_q}) - 2 \sum_{q=1}^{m} w_q E(\mathbf{C} \wedge \mathbf{S_q}), \tag{5}$$

where the middle term depends only on the input. It is, therefore, a constant that can be dropped for the purpose of optimization. Making use of the fact that the weights are normalized, we obtain the objective function

$$\tilde{f}(\mathbf{C}) := f(\mathbf{C}) - \sum_{q=1}^{m} w_q E(\mathbf{S_q}) = E(\mathbf{C}) - 2 \sum_{q=1}^{m} w_q E(\mathbf{C} \wedge \mathbf{S_q}). \tag{6}$$

Now, we explicitly consider $\mathbf{C}$ as a sequence of intervals A. Using the additivity of the potential E, we obtain

$$\tilde{f}(\mathbf{C}) = \sum_{A\in\mathbf{C}} \left(\mathfrak{e}(A) - 2\sum_{q=1}^{m} w_q \sum_{\substack{B\in\mathbf{S}_q \\ B\cap A\neq\emptyset}} \mathfrak{e}(A\cap B) \right) =: \sum_{A\in\mathbf{C}} \Delta(A). \tag{7}$$

The additive form of Equation (7) as a sum of the contributions $\Delta(A)$ for the consensus segments $\mathbf{C}$ makes it possible to minimize $\tilde{f}(\mathbf{C})$ by dynamic programming [24]. To this end, consider the subset $\mathcal{C}_{|k}$ of segmentations that have a segment boundary at k, i.e. , position k is the endpoint of a segment. For a given segmentation $\mathbf{C} \in \mathcal{C}_{|k}$, denote by $\tilde{f}(\mathbf{C}|k)$ the sum of the contributions $\Delta(A)$ with $\max A \leq k$. Write $F_k := \min_{\mathbf{C}\in\mathcal{C}_{|k}} \tilde{f}(\mathbf{C}|k)$ for the minimal value of $\tilde{f}(\mathbf{C}|k)$. Since k is a segment boundary, the last segment A before k is necessarily of the form $[j+1,k]$, where $j<k$ denotes the segment boundary immediately preceding k. Using this notation, we can compute

$$\begin{aligned} F_k &= \min_{\mathbf{C}\in\mathcal{C}_{|k}} \tilde{f}(\mathbf{C}|k) = \min_{j<k} \min_{\mathbf{C}\in\mathcal{C}_{|j}} \left(\Delta([j+1,k]) + \tilde{f}(\mathbf{C}|j)\right) \\ &= \min_{j<k} \left(\Delta([j+1,k]) + \min_{\mathbf{C}\in\mathcal{C}_{|j}} \tilde{f}(\mathbf{C}|j) \right) = \min_{j<k} \left(\Delta([j+1,k]) + F_j\right). \end{aligned} \tag{8}$$

Thus, we obtain a simple dynamic programming recursion that has the same form for the weighted and unweighted consensus segmentation; also see [24]. The weights appear only in the scoring function Δ. We note, furthermore, that the recursion (8) is the same as for segmentation problems in general [28]. It appears, e.g., in Reference [29] for financial time series, in Reference [30] in context of text segmentation, in Reference [31] for the analysis of array CGH data, and in Reference [5,7,32] for the identification of transcripts in tiling array and RNA-seq data. It is discussed in the setting of very general similarity measures in Reference [11]. As we shall see below, the effort to compute F_k is dominated by the effort to compute the score $\Delta[i,j]$.

Before we proceed, we briefly consider a general condition on the form of the potential function $\mathfrak{e}(\,.\,)$. Denote by $\mathbf{D}$ the discrete segmentation in which every interval is a single point and by $\mathbf{J}$ the indiscrete segmentation consisting of a single interval. A function $\mathfrak{e}$ is *subadditive* if $\mathfrak{e}(A) \leq \mathfrak{e}(A_1) + \mathfrak{e}(A_2)$ for every A and every subdivision $A_1 \dot\cup A_2 = A$ of A. This inequality is strict for at least one interval if and only if $\mathfrak{e}([1,n]) < \sum_{i=1}^{n} \mathfrak{e}([i,i])$. Comparing $\mathbf{D}$ and $\mathbf{J}$, we observe that, in this case, $D(\mathbf{D},\mathbf{J}) = \mathfrak{e}([1,n]) - \sum_{i=1}^{n}[i,i] < 0$, violating that D is a proper distance function. For the limiting case of an additive potential, $\mathfrak{e}(A) = \mathfrak{e}(A_1) + \mathfrak{e}(A_2)$ for all intervals and their subdivisions, we obtain $D(\mathbf{S},\mathbf{S}') = 0$ for any two segmentations $\mathbf{S}$ and $\mathbf{S}'$. Thus, only potentials that satisfy $\mathfrak{e}(A) > \mathfrak{e}(A_1) + \mathfrak{e}(A_2)$ for at least some $A_1 \dot\cup A_2 = A$ are of interest. A function is *superadditive* if $\mathfrak{e}(A) \geq \mathfrak{e}(A_1) + \mathfrak{e}(A_2)$ for all $A_1 \dot\cup A_2 = A$. One easily checks that D is a metric whenever $\mathfrak{e}$ is superadditive. This condition is not necessary, however. For example, the negentropy defined in Equation (17) below, is not superadditive.

2.2. Efficient Computation of the Segment Scores $\Delta[i,j]$

The direct evaluation of $\Delta([i,k])$ according to its definition, Equation (8), for given i and k, requires $O(n\,m)$ operations because this entails the summation over $O(n)$ segments for each of the m input segmentations. This results in an impractical total effort of $O(n^3\,m)$ compared to the quadratic cost of the dynamic programming recursion itself. It is of considerable practical interest, therefore, to find a more efficient way of computing the scoring function. The key idea is to consider, for a given position i, two slightly different partial sums:

$$\delta_<(i) := \sum_{q=1}^{m} w_q \sum_{\substack{B\in\mathbf{S}_q \\ \max B\le i}} \mathfrak{e}(B) \qquad \text{and} \qquad \delta_\le(i) := \sum_{q=1}^{m} w_q \sum_{\substack{B\in\mathbf{S}_q \\ \min B\le i}} \mathfrak{e}(B). \tag{9}$$

For a given boundary i in $\mathbf{C}$, the first term sums all intervals in $\mathbf{S}_q$ that do not extend beyond i, while the second sum also includes those that begin before or at i and extend beyond i. Thus, $\delta_<(k) - \delta_\le(j)$ captures all segments of the $\mathbf{S}_q$ that are contained within $[j+1,k]$ with one important exception: Segments, such as B in Figure 2, that contain $[j+1,k]$ contribute to $\delta_\le(j)$ but not to $\delta_<(k)$. Such overlapping segments will be taken care of in a correction term discussed below. Using the notation $B_{\le i} := \{b \in B | b \le i\}$ and $B_{\ge i} := \{b \in B | b \ge i\}$ we, furthermore, define terms

$$\delta^\cap_<(i) := \sum_{q=1}^{m} w_q \sum_{\substack{B\in\mathbf{S}_q \\ i\in B, i\neq \max B}} \mathfrak{e}(B_{\le i}) \qquad \text{and} \qquad \delta^\cap_>(i) := \sum_{q=1}^{m} w_q \sum_{\substack{B\in\mathbf{S}_q \\ i\in B, i\neq \min B}} \mathfrak{e}(B_{\ge i}), \tag{10}$$

where each sum contains at most a single term, namely the interval $B \in \mathbf{S}_q$ that extends across i. Note that intervals that begin or end in position i do not contribute to $\delta^\cap_>(i)$ or $\delta^\cap_<(i)$, respectively. The correction terms correspond to the parts of segments that are non-trivially intersected by $[j+1,k]$, shown in magenta and cyan, resp., in Figure 2. Thus $\delta_<(k) - \delta_\le(j) + \delta^\cap_<(k) + \delta^\cap_>(j+1)$ covers exactly all the intervals contributing to $[j+1,k]$ – with the exception of segments $B \in \mathcal{S}_q$ that begin at $minB < j+1$ and end at $\max B > k$ as mentioned above. For such segments, instead of the contributions for $B_{\le k}$ and $B_{\ge j+1}$, a single contribution for the interval $[j+1,k] \cap B = [j+1,k]$ has to be used. In addition, the contribution for B that is erroneously subtracted with $\delta_\le(j)$, needs to be restored. Collecting these contributions, we obtain the following correction term for intervals that span across the interval $[i',i'']$ of interest:

$$\delta^*(i',i'') := \sum_{q=1}^{m} w_q \sum_{\substack{B\in\mathbf{S}_q \\ i',i''\in B \\ \min B<i'\le i''<\max B}} \big(\mathfrak{e}(B) + \mathfrak{e}([i',i'']) - \mathfrak{e}(B_{\le i''}) - \mathfrak{e}(B_{\ge i'})\big). \tag{11}$$

For the interval $[j+1,k]$, the correction term $\delta^*(j+1,k-1)$ defined in Equation (11) can be understood as follows: the first term accounts for the correct contribution of $B \cap [j+1,k] = [j+1,k]$, the second term compensates for the error introduced by $\delta_\le(j)$, and the remaining two terms remove the superfluous contributions introduced by $\delta^\cap_<(k)$ and $\delta^\cap_>(j+1)$. We summarize this derivation in the following form:

Theorem 1. *The potential-dependent segment scores defined in Equation (7) can be expressed as*

$$\Delta([j+1,k]) = \mathfrak{e}([j+1,k]) - 2\big(\delta_<(k) - \delta_\le(j) + \delta^\cap_<(k) + \delta^\cap_>(j+1) + \delta^*(j+1,k)\big). \tag{12}$$

The only term that depends on both $j+1$ and k is the correction of long input intervals $\delta^*(j+1,k)$. The restricted sum over the $B \in \mathbf{S}_q$ in Equation (11) contains at most one segment for each input segmentation and, thus, can be evaluated in $O(m)$ time for a given interval. Furthermore, the sum is certainly empty whenever $[j+1,k]$ is larger than the largest segment in any of the $\mathbf{S}_q$; this can be used to speed up the evaluation from $O(m)$ to $O(1)$ if the segment lengths in the input are bounded by a constant, except for the short intervals.

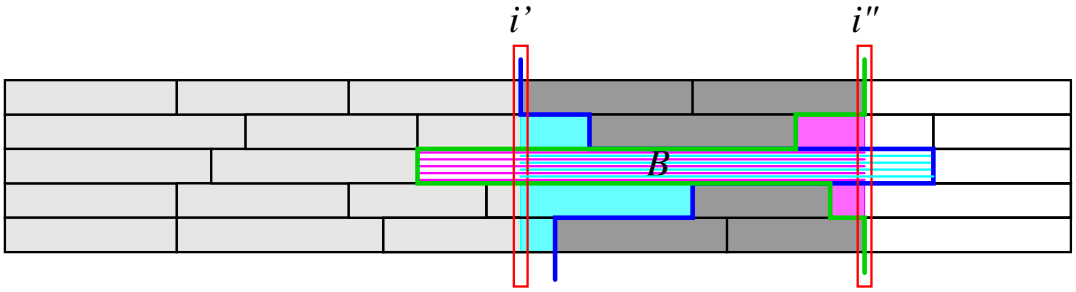


Figure 2. Definition of auxiliary variables. The input segments contributing to $\delta_<(i'')$ are all those to the left of the green line (i.e., the ones shown in light and dark gray. $\delta_\leq(i')$ are to the left of the blue line, i.e, those shown in light gray. The large interval B is included in $\delta_\leq(i')$ but not in $\delta_<(i'')$. The correction terms $\delta^\cap_>(i')$ and $\delta^\cap_<(i'')$ comprise the cyan and magenta parts, respectively. The correction term $\delta^*(i', i'')$, finally adds takes care of the interval B.

Lemma 1. *The arrays of correction terms $\delta_<$, $\delta_\leq$, $\delta^\cap_<$, and $\delta^\cap_>$ can be computed in $O(nm)$ total time.*

Proof. The values of $\delta_<(i)$ and $\delta_\leq(i)$ can be computed iteratively: we obtain $\delta_<(i)$ by adding the contribution $w_q\mathfrak{e}(B_q)$ to $\delta_<(i-1)$ whenever $i = \max B_q$ for the segmentation $\mathbf{S_q}$. Similarly, $\delta_\leq(i)$ is obtained by adding $w_q\mathfrak{e}(B_q)$ to $\delta_\leq(i-1)$ if $i = \min B_q$. For each i, therefore, we require $O(m)$ operations. The sums in Equation (10) comprise at most one segment of $\mathbf{S}_q$ for every q. All terms can be computed in constant time using auxiliary arrays that return, for each i and q, the values of $\min B$ and $\max B$ for $i \in B$ and $B \in \mathbf{S}_q$. These auxiliary arrays in turn can obviously be constructed in $O(nm)$ time for the breakpoint list of the input segmentations. □

It, therefore, makes sense to precompute the arrays $\delta_<$, $\delta_\leq$, $\delta^\cap_<$, and $\delta^\cap_<(k)$.

Corollary 1. *The score $\Delta[j+1,k]$ can be computed in $O(m)$ time with $O(nm)$ preprocessing cost to compute the arrays $\delta_<$, $\delta_\leq$, $\delta^\cap_<$, and $\delta^\cap_>$.*

It is worth noting, finally, that there is nothing to be gained by storing the score values $\Delta[j+1,k]$ since each entry is used only once in the recursion.

2.3. Boundaries of Consensus Segments

For a function g on $\mathbb{Z}$, we define the local curvature at x as $\partial^2_x g(x) := g(x+1) + g(x-1) - 2g(x)$. A function g is (strictly) *convex* at x if $\partial^2_x g(x) > 0$. This condition immediately implies that x is not a local maximum of g since at least one of $g(x+1)$ or $g(x-1)$ is larger than $g(x)$. Correspondingly, g is (strictly) *concave* in x if $\partial^2_x g(x) < 0$, whence x is not a local minimum.

Definition 1. *The potential $\mathfrak{e}$ is* boundedly convex *if satisfied for all intervals $p' \leq p \leq x \leq q \leq q'$*

$$\partial^2_x\mathfrak{e}([x,q]) \geq \partial^2_x\mathfrak{e}([x,q']) > 0 \quad \textit{and} \quad \partial^2_x\mathfrak{e}([p,x]) \geq \partial^2_x\mathfrak{e}([p',x]) > 0\,. \tag{13}$$

For boundedly convex $\mathfrak{e}$, the curvature in non-increasing as the intervals become larger. In particular, suppose $\mathfrak{e}([p,q])$ depends only on the length $z := q - p + 1$ of the interval and is a smooth function, then $\mathfrak{e}''(z) > 0$ and $\mathfrak{e}'''(z) \leq 0$ for all $z > 0$ implies $\mathfrak{e}$ is boundedly convex.

Theorem 2. *Let $\{\mathbf{S}_1, \mathbf{S}_1, \ldots, \mathbf{S}_m\}$ be a set of segmentations with union segmentation $\hat{\mathbf{S}}$ and suppose $\mathfrak{e}$ is boundedly convex. Then, the consensus $\mathbf{C}$ is refined by the union segmentation $\hat{\mathbf{S}}$.*

Proof. Following Reference [24,33], we assume, for contradiction, that the optimal consensus $\mathbf{C}$ has a segment boundary $\hat{\jmath}$ that is not contained in the union segmentation $\hat{\mathbf{S}}$. We aim to show that moving $\hat{\jmath}$ to some close-by position x will reduce the cost $f(\mathbf{C})$. We focus on a fixed input segmentation $\mathbf{S}_q$ and denote by $\hat{\mathbf{S}}_\mathbf{q} := \mathbf{S}_\mathbf{q} \wedge \mathbf{C}$. Denote by $p-1$ and q the first boundaries to the left and to the right of $\hat{\jmath}$ in $\mathbf{C}$. Analogously, $\hat{p}$ and $\hat{q}$ are the first boundary to the left and to the right of $\hat{\jmath}$ in $\hat{\mathbf{S}}_\mathbf{q}$, respectively. Thus, $\mathbf{C}$ contains the two segments $[p, \hat{\jmath}]$ and $[\hat{\jmath}+1, q]$. Since every segment of $\hat{\mathbf{S}}_\mathbf{q} \wedge \mathbf{C}$ is a subset of a unique segment of $\mathbf{C}$, we have $[\hat{p}, \hat{q}] \subseteq [p, q]$.

We proceed by evaluating how $D(\mathbf{S}_\mathbf{q}, \mathbf{C}) = E(\mathbf{S}_\mathbf{q}) + E(\mathbf{C}) - 2E(\mathbf{S}_\mathbf{q} \wedge \mathbf{C})$ varies when the boundary $\hat{\jmath}$ is perturbed. Let x be the perturbed boundary position. Since only $E(\mathbf{C})$ and $2E(\mathbf{S}_\mathbf{q} \wedge \mathbf{C})$ depends on x and all boundaries except $\hat{\jmath}$ are fixed, it suffices to focus on the intervals $[p, q]$ and $[\hat{p}, \hat{q}]$, respectively. Collecting all constant terms in D_0, we obtain

$$D(x) = D_0 + \mathfrak{e}([p, x]) + \mathfrak{e}([x+1, q]) - 2\mathfrak{e}([\hat{p}, x]) - 2\mathfrak{e}([x+1, \hat{q}]). \tag{14}$$

Since $\mathfrak{e}$ is boundedly convex, we have $0 < \partial_x^2 \mathfrak{e}([p, x]) \leq \partial_x^2 \mathfrak{e}([\hat{p}, x])$ and $0 < \partial_x^2 \mathfrak{e}([x+1, q]) \leq \partial_x^2 \mathfrak{e}([x+1, \hat{q}])$, whence $\partial_x^2 D(x) < -\partial_x^2 \mathfrak{e}([\hat{p}, x]) - \partial_x^2 \mathfrak{e}([x+1, \hat{q}]) < 0$. Thus, $D(x)$ is concave at x for every $\mathbf{S}_\mathbf{q}$ and, thus, also for any non-negative contribution to the linear combination of input segmentations. Thus, $f(\mathbf{C})$ as a function of the moving boundary x cannot have a minimum in the interior of the interval $[\hat{p}, \hat{q}]$, contradicting the assumption that $\hat{\jmath}$ is a boundary in the optimal consensus $\mathbf{C}$. □

Theorem 2 establishes a very useful property: All segment boundaries of the consensus are contained in the union segmentation. This property was observed for disagreement distance and information distance (see below) in the unweighted setting [24,33]. Here, we show it holds for a broader class of distance functions and arbitrary weighting schemes. The techniques used in the proof of Theorem 2 do not seem to generalize to potentials with increasing curvature. Numerical data, however, indicate that the union segmentation refines the consensus for a much larger class of potential functions.

From an algorithmic point of view, it implies that it suffices to compute the F_k for those values of k where segment boundaries are in the union segmentation of the inputs $\hat{\mathbf{S}}$. Correspondingly, we need to store the auxiliary variables only for the intervals of the union segmentation, instead of each i. That is, the recursion (8) reduced to

$$F_{j_k} = \min_{\substack{i<k \\ j_i \in \partial\hat{\mathbf{S}}}} \big(\Delta([j_i+1, j_k]) + F_{j_i}\big), \tag{15}$$

where $j_i \in \partial\hat{\mathbf{S}}$ denotes the i-th segment boundary in the union segmentation $\hat{\mathbf{S}}$.

Recursion (15) also speeds up the computation of the scoring function Δ, which now is also needed only for the segment boundaries. First, note that we still obtain $\delta_<(i_k)$ from $\delta_<(i_{k-1})$ by adding the contributions $\mathfrak{e}(B)$ for the intervals ending at the boundary i_k to $\delta_<(i_{k-1})$ since by definition i_{k-1} and i_k are consecutive breakpoints. Analogously, $\delta_>(i_k)$ is obtained by adding $\mathfrak{e}(B)$ for all blocks beginning at i_{k-1} to $\delta_>(i_{k-1})$. The terms $\delta_>^\cap(i_k)$ and $\delta_<^\cap(i_k)$ remain the same. The correction term δ^* could be stored for all pairs of the boundaries in $\hat{\mathbf{S}}$. Alternatively, it suffices to store the m boundaries at which the intervals crossing i_k start and to keep track of the correct correction term directly in recursion (15). Equation (15), thus, can be evaluated in $O(s^2)$, where s is the number of breakpoints in the union segmentation.

2.4. Length Bounds on Consensus Segments

It is reasonable to expect that a consensus segmentation cannot be a lot coarser than the individual input segmentations. To see that this is indeed the case, we start with a technical observation.

Lemma 2. *Consider intervals* $A = [i,k]$, $A' = [i,x]$ *and* $A'' = [x+1,k]$. *Then* $\Delta(A) > \Delta(A') + \Delta(A'')$ *if for every* $\mathbf{S_q}$ *there is* $B \in \mathbf{S_q}$ *with* $x \in B$ *such that*

$$\mathfrak{e}(A) - \big(\mathfrak{e}(A') + \mathfrak{e}(A'')\big) > 2\big[\mathfrak{e}(B) - \big(\mathfrak{e}(B \cap A') + \mathfrak{e}(B \cap A'')\big)\big] \tag{16}$$

Proof. Equation (7) implicitly defined $\Delta(A)$ as the term in parentheses, which in turn is the w_q-weighted sum of contributions for each $\mathbf{S_q}$. Consider $B \in \mathbf{S_q}$ with $B \subset A$. The contribution $d_q(A)$ of $\mathbf{S_q}$ to $\Delta(A)$ is

$$\begin{aligned} d_q(A) &= \mathfrak{e}(A) - 2\sum_{B' \in \mathbf{S_q}} \mathfrak{e}(A \cap B') \\ &= \mathfrak{e}(A) - 2\left(\sum_{\substack{B' \in \mathbf{S_q}: \\ \max B' < \max B}} \mathfrak{e}(A \cap B') + \sum_{\substack{B' \in \mathbf{S_q} \\ \max B' > maxB}} \mathfrak{e}(A \cap B')\right) - 2\mathfrak{e}(B). \end{aligned}$$

Now, consider an alternative segmentation in which A is subdivided into $A' \cup A''$ at some position x inside B. Then, A contributes

$$\begin{aligned} d_q(A') + d_q(A'') &= \mathfrak{e}(A') + \mathfrak{e}(A'') - 2\big(\mathfrak{e}(A' \cap B) + \mathfrak{e}(A'' \cap B)\big) \\ &\quad -2\left(\sum_{\substack{B' \in \mathbf{S_q}: \\ \max B' < maxB}} \mathfrak{e}(A' \cap B') + \sum_{\substack{B' \in \mathbf{S_q} \\ \max B' > maxB}} \mathfrak{e}(A'' \cap B')\right). \end{aligned}$$

The terms corresponding to the segments $B' \neq B$ that intersect A are the same as before since either $B' \cap A = B' \cap A'$ or $B' \cap A = B' \cap A''$, depending on whether B' comes before or after B in $\mathbf{S_q}$. Thus, we have $d_q(A) > d_q(A') + d_q(A')$ if and only if Equation (16) is satisfied. Since $\Delta(A)$, $\Delta(A')$, and $\Delta(A'')$ are convex linear combinations of the $d_q(A)$, $d_q(A')$, and $d_q(A'')$, respectively, it is sufficient for $\Delta(A) > \Delta(A') + \Delta(A'')$ that $d_q(A) > d_q(A') + d_q(A')$ holds for all $\mathbf{S_q}$. □

In other words, if A satisfies the condition of Lemma 2, then $\tilde{f}(\mathbf{C})$ strictly decreases when A is subdivided into A' and A''. Thus, we conclude:

Corollary 2. *An interval* A *satisfying the conditions specified in Lemma 2 cannot appear in a consensus segmentation.*

Our goal is now to show that sufficiently long intervals A always satisfy the conditions of Lemma 2 and, thus, can never be part of the consensus segmentation. Here, we need that $\mathfrak{e}$ is superadditive, i.e., $\mathfrak{e}(A) > e(A_1) + e(A_2)$ for all $A = A_1 \dot\cup A_2$ and $A_1, A_2 \neq \emptyset$. This is the case particularly for the polynomial potentials. It fails for the negentropy potential, Equation (17), however, because this function is not monotonically increasing with the segment length $|A|$.

Theorem 3. *Let* $\mathfrak{e}$ *be a superadditive potential. Let* B *be the longest segment in the input segmentations and denote by* ℓ^* *the length of the shortest interval* A *such that* $\mathfrak{e}(A) - 2\mathfrak{e}(A') > 2\mathfrak{e}(B) - 2\min_{B',B'':B' \cup B'' = B}(\mathfrak{e}(B') + \mathfrak{e}(B''))$, *where* $|A'| = \lceil |A|/2 \rceil$ *and* $|B'| = \lfloor |B|/2 \rfloor$. *Then every segment of the consensus segmentation is shorter than* $L^* := \max(2|B|, \ell^*)$.

Proof. If $\mathfrak{e}$ is superadditive, the l.h.s. of Equation (16) is maximal if $|A'| = |A''|$ (for even $|A|$) or $|A'| = |A''| \pm 1$ for odd $|A|$, i.e., we assume that x is located in the middle of A. In order to ensure that segments containing x are completely contained in A we need $|A| \geq 2|B|$. If this condition is satisfied, Equation (16) applies. We obtain a sufficient condition by replacing the r.h.s. with the maximal possible contribution of the subdivided interval B. By superadditivity, this term monotonically increases with the size of B. The

assumption that x equally divides A fixed the l.h.s. of the inequality. Since $\mathfrak{e}$ is strictly superadditive $\mathfrak{e}(A) - 2\mathfrak{e}(A')$ is strictly monotonically increasing with $|A|$, thus, there is a unique smallest value ℓ^* of $|A|$ unless $\mathfrak{e}(A) - 2\mathfrak{e}(A') \leq 2$ for all A, in which case no bound ℓ^* exists. □

Corollary 3. *The consensus segmentation* $\mathbf{C}$ *with superadditive potential* $\mathfrak{e}$ *for* m *input segmentations with length bound* L^* *as specified in Theorem 3 can be computed in* $O(nmL^*)$ *time.*

Proof. We observe that for each k, only values of j between $k - 2\ell^*$ and $k - 1$ appear in Equation (8) since longer segments by Theorem 3 cannot be part of an optimal consensus segmentation. The corollary now follows immediately from Corollary 1. □

The length bound on consensus segments, thus, leads to a reduction of the computational efforts. Although ℓ^* in Theorem 3 may be inconvenient to compute for some choices of the potential $\mathfrak{e}$, we shall see below that a simple, uniform bound can be obtained for an interesting class of potentials.

2.5. Special Potential Functions

Let us now consider plausible distance functions. The *disagreement distance* between segmentation was introduced in Reference [24] using the potential $\mathfrak{e}(A) := (|A|/n)^2/2$. A natural generalization is $\mathfrak{e}(A) = (|A|/n)^{1+\alpha}/(1+\alpha)$ for $0 < \alpha \leq 1$. We note that a linear potential $\mathfrak{e}(|A|) = |A|/n$, i.e., $\alpha = 0$, yields a constant value of $\tilde{f}(\mathbf{C})$ because the sum of all segment lengths adds up to n; thus, $E(\mathbf{S}) = 1$ is independent of the segmentation $\mathbf{S}$.

Recall that the entropy of a discrete distribution is defined as $H = -\sum_i p_i \ln p_i$. Given a segmentation $\mathbf{S}$, we consider the probabilities p_i of randomly picking a point from a segment, i.e., $p_i = |A_i|/n$ is the relative length of a segment $A_i \in \mathbf{S}$, where n denotes the total length of the segmented genome or time series. The *information distance* is the symmetrized conditional entropy, which can also be computed as $D(\mathbf{S}', \mathbf{S}'') = 2H(\mathbf{S}' \wedge \mathbf{S}'') - H(\mathbf{S}') - H(\mathbf{S}'')$ [24,34]. It corresponds to the potential function

$$\mathfrak{e}(A) := (|A|/n)\ln(|A|/n), \tag{17}$$

given by the negative of the entropy (negentropy) contribution of the interval A.

It has been shown in Reference [24,33] that the union segmentation $\hat{\mathbf{S}}$ refines the unweighted consensus segmentation for both the disagreement distance and the information distance. This result generalizes to the weighted case and the α-disagreement distances with $0 < \alpha \leq 1$.

Corollary 4. *The consensus segmentation* $\mathbf{C}$ *is refined by the union segmentation* $\hat{\mathbf{S}}$ *for the disagreement distance, its* α *generalization with* $0 < \alpha \leq 1$*, as well as the information distance.*

Proof. It suffices to show that the potentials $\mathfrak{e}(z)$ are boundedly convex. For the disagreement distance, we have $\mathfrak{e}(z) = z^2/2$, and we have $\mathfrak{e}''(z) = 1$ and $\mathfrak{e}'''(z) = 0$; for $\mathfrak{e}(z) = z^{1+\alpha}/(1-\alpha)$, we have $e''(z) = \alpha z^{\alpha-1} > 0$ and $e'''(z) = \alpha(\alpha-1)z^{\alpha-2} < 0$ for $0 < \alpha \leq 1$. For the negentropy, $\mathfrak{e}(z) = z\ln z$, we have $\mathfrak{e}''(z) = 1/z > 0$ and $\mathfrak{e}'''(z) = -1/z^2 < 0$, where $z := (|A|/n)$. The scaling by $1/n$ obviously does not affect the signs. □

Is does not seem possible to generalize the result to potentials that grow faster than quadratically.

Let us finally consider the consequence of Theorem 3. Reusing the convexity results above we can replace $2\min_{B',B'':B'\cup B''=B}(\mathfrak{e}(B') + \mathfrak{e}(B''))$ by $4\mathfrak{e}(B')$ where $|B'| = \lfloor |B|/2 \rfloor$. A short computation then shows that the inequality in Theorem 3 is satisfied for $|A| > \sqrt[1+\alpha]{2}|B|$. Since $\sqrt[1+\alpha]{2} \leq 2$, we have

Corollary 5. *The consensus segmentation* $\mathbf{C}$ *of a collection of segmentations* $\mathbf{S_q}$ *with respect to the* α*-disagreement potentials contains no segment longer than twice the length of the longest input segment.*

This allows us immediately to limit the range of the indices in recursion (8) to $j_i > j_k - 2\max|B|$.

2.6. Generalization: Symmetrized Boundary Mover's Distance

Equation (7) highlights the fact that the cost function $\tilde{f}(\mathbf{C})$ measures, for each segment $A \in \mathbf{C}$, how well A conforms to the input segmentations. As noted above, the additive structure of Equation (7) is sufficient to enable minimization by dynamic programming for arbitrary choices of Δ. If we retain the idea of weighted contributions for each input segmentation, we may write $\Delta(A) = \sum_q w_q \Delta(A|\mathbf{S}_q)$, where $\Delta(A|\mathbf{S}_q)$ measures how well the segment A "fits" into the segmentation $\mathbf{S}_q$. As a minimal requirement, for any given interval A, the score $\Delta(A|\mathbf{S}_q)$ must attain its minimum value if the interval A is a segment in $\mathbf{S}_q$. Since two segmentations in general do not have segments or breakpoints in common, measures are required that are more fine-grained than the distinction between identical and distinct segments or breakpoints. $\Delta(A|\mathbf{S}_q)$, thus, are similar to a measure of overlap, between A and the segments of $\mathbf{S}_q$ that are covered by A. Clearly, the potential-based measures can be understood in this manner.

An interesting class of dissimilarities utilizes the distance between break points instead of the lengths of intersections between segments. For a segmentation $\mathbf{S}$ with segments S_i, $i = 1, \ldots, n$, we define $s_i = \max S_i$ and set $s_0 = 0$, i.e., the segments are $S_i = [s_{i-1} + 1, s_i] =: (s_{i-1}..s_i)$. By slight abuse of notation, we write $\mathbf{S} = (s_0, s_1, \ldots, s_m)$, i.e., we now specify a segmentation in terms of its breakpoints. Moreover, we write $s \in \mathbf{S}$ to mean that s represents a breakpoint in the segmentation $\mathbf{S}$.

The *"boundary movers distance"* was introduced in Reference [24,33] as

$$D_B(\mathbf{S}|\mathbf{C}) := \sum_{s \in \mathbf{S}} \min_{c \in \mathbf{C}} d(s, c)\,, \tag{18}$$

where $d(\,.\,,\,.\,)$ is some distance function between the positions s and c on $[1, \ldots, n]$. The dissimilarity measure D_B is not symmetric and satisfies $D_B(\mathbf{S}|\mathbf{C}) = 0$ whenever $\mathbf{C}$ is a refinement of $\mathbf{S}$. The segmentation aggregation problem that minimizes $\sum_q w_q D_B(\mathbf{S_q}|\mathbf{C})$, therefore, is solved by the union segmentation $\mathbf{C} = \hat{\mathbf{S}}$, while $\sum_q w_q D_B(\mathbf{C}|\mathbf{S_q}|)$ is minimized by the indiscrete segmentation $\{[1, n]\}$. As noted in Reference [24,33], these measures, thus, are only useful with additional constraints on the number or size of allowed segments.

The symmetrized version of D_B, however, has attractive properties for our purposes, as we shall see: Clearly, $\min_{c \in \mathbf{C}} d(s, c) = \min\{d(s, c'), d(s, c'')\}$, where c' and c'' delimit the segment of $\mathbf{C}$ within which s resides. If $s = c'$ or $s = c''$, the contribution vanishes; hence, we can write

$$D_B(\mathbf{S}|\mathbf{C}) := \sum_{(c'\ldots c'') \in \mathcal{C}} \sum_{s \in (c'\ldots c'')} \min\{d(s, c'), d(s, c'')\}. \tag{19}$$

This term individually penalizes a segment (c', c'') of $\mathbf{C}$ for containing boundary points of $\mathbf{S}$ in its interior. On the other hand, we can rewrite $D_B(\mathbf{C}|\mathbf{S})$ in terms of segments of $\mathbf{C}$ by simply splitting the contribution of each boundary between the two adjacent segments:

$$D_B(\mathbf{C}|\mathbf{S}) = \sum_{(c', c'') \in \mathbf{C}} \frac{1}{2} \left(\min_{s \in \mathbf{S}} d(s, c') + \min_{s \in \mathbf{S}} d(s, c'') \right). \tag{20}$$

Here, we have used that the lower bounds of the first segments and the upper bounds of the last segments necessarily coincide (they are the boundary of the interval on which our segmentations live); therefore, they do not contribute to the distance. Again, the minima are only taken over two alternative breakpoints of $\mathbf{S}$ for each given value of c' or c'', namely those delimiting the segments of $\mathbf{S}$ harboring the breakpoints c' and c'' of the consensus

segment. It is not difficult to see that $D(\mathbf{S},\mathbf{C}) = D_B(\mathbf{S}|\mathbf{C}) + D_B(\mathbf{C}|\mathbf{S})$ vanishes only if $\mathbf{S} = \mathbf{C}$. Furthermore, $D_B(\mathbf{S}_q|\mathbf{C}) + D_B(\mathbf{C}|\mathbf{S_q})$ can be written as a sum of contributions

$$\Delta_q((c'...c'')) := \sum_{\substack{s\in S_q \\ s\in(c'...c'')}} \min\{d(s,c'),d(s,c'')\} + \frac{1}{2}\min_{s\in\mathbf{S}_q} d(s,c') + \frac{1}{2}\min_{s\in\mathbf{S}_q} d(s,c'') \tag{21}$$

for each of the segments $(c',c'') \in \mathbf{C}$ and each segment of the input segmentations $\mathbf{S}_q$. Clearly, $\Delta((c'...c'')) = \sum_q w_q\Delta_q((c'...c''))$ depends only on the input segmentations $\mathbf{S}_q$ and the boundary breakpoints c' and c'', i.e., an individual segment in the consensus $\mathbf{C}$. The segmentation aggregation problem with the symmetrized boundary movers distance, therefore, can again be solved by dynamic programming recursion Equation (8). A more in-depth analysis of this distance function is the subject of ongoing research.

3. Computational Results

3.1. Implementation

The consensus segmentation algorithm is available as an *R* package *consseg*, where the dynamic programing recursion is implemented in *C++* via *Rcpp* ($\geq$0.12.18) and *RcppXPtrUtils* ($\geq$0.1.1) to allow user-defined potential functions. A CRAN package accompanying this contribution will be made available. The development version is available at https://github.com/Bierinformatik/consseg (accessed on 1 December 2020).

The input segmentations are converted into an index returning for each position k the minimum position $\min B_q$ and the maximum position $\max B_q$ of the segment B_q containing k. With their help, $\delta_<(k)$ is obtained by adding $w_q\mathfrak{e}(B_q)$ to $\delta_<(k-1)$ if $k = \max B_q$. Analogously, $\delta_\leq(k)$ is computed by adding $w_q\mathfrak{e}(B_q)$ to $\delta_\leq(k)$ whenever $k = \min B_q$. The terms $\delta^{\cap}_<(k)$ and $\delta^{\cap}_>(k)$ are evaluated as defined in Equation (10). These computations are interleaved with the evaluation of F_k, Equation (8). Since the expensive part in the algorithm is the evaluation of the segment cost $\Delta[j+1,k]$, we avoid their recomputation in the backtracking step by storing instead the values $J[k]$ of the segment boundary j that realizes the minimum in Equation (8) for position k. Thus, the last segment of the optimal segmentation on $[1,k]$ is $[J[k]+1,k]$. Backtracking then proceeds on $[1,J[k]]$. The segment boundaries of the optimal segmentation are, therefore, obtained as $j_{i+1} = J[j_i]$, starting from $j_0 = n$ and continuing until $j_k = 0$ is reached. It is worth noting that the fast, incremental Equation (12) is prone to rounding errors for large n and very fast-growing potentials $\mathfrak{e}$ due to the computation of difference $\delta_<(k) - \delta_\leq(j)$ of two sums.

3.2. Consensus Segmentation of Yeast Transcriptome Data

To demonstrate the usefulness of consensus segmentations, we explored the yeast transcriptome time series mentioned in the introduction. We computed the consensus of segmentations obtained with widely different parameter choices [11]. We found that the consensus segmentation appears to produce a robust representation of the transcriptome and seems to fit better to the current annotation of the yeast genome than any particular choice of segmentation parameters. The example in Figure 3 also shows that distinct non-coding components, such as SRG1, are readily detectable. Short segments with very low coverage are likely gaps between transcriptome units without relevant RNA products. Consistently detectable elements, even if lowly expressed, on the other hand, may be of interest for closer inspection.

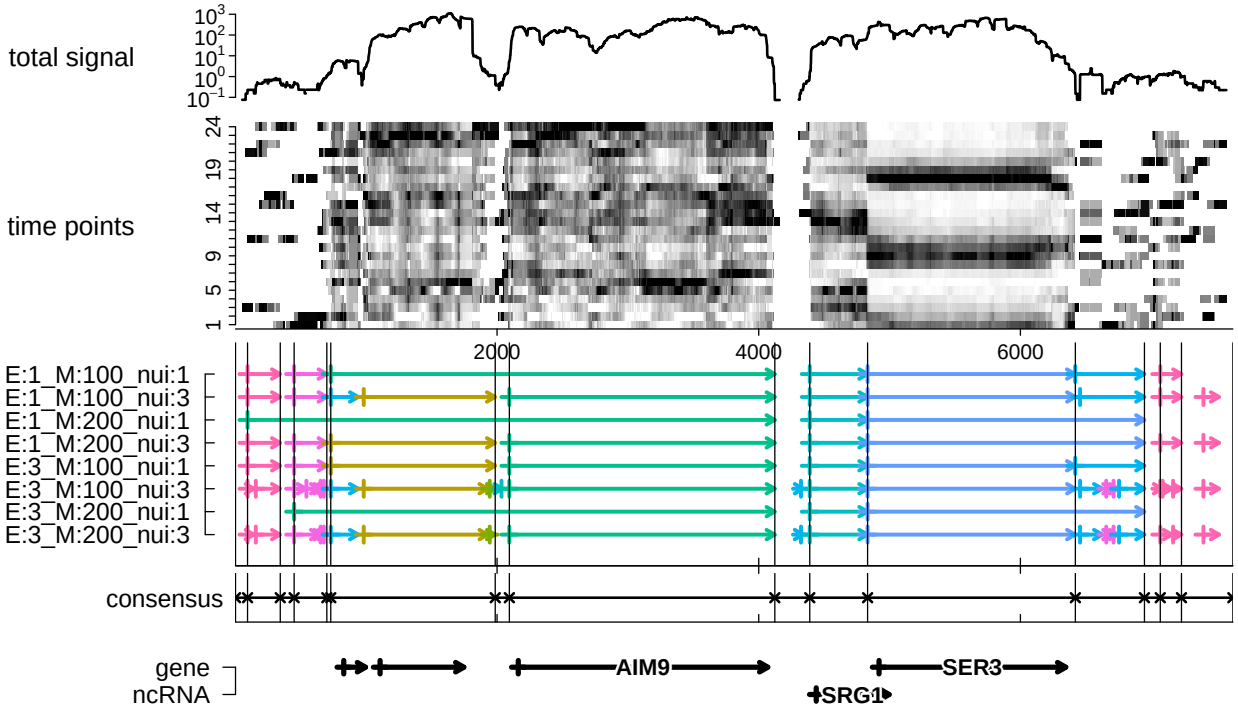


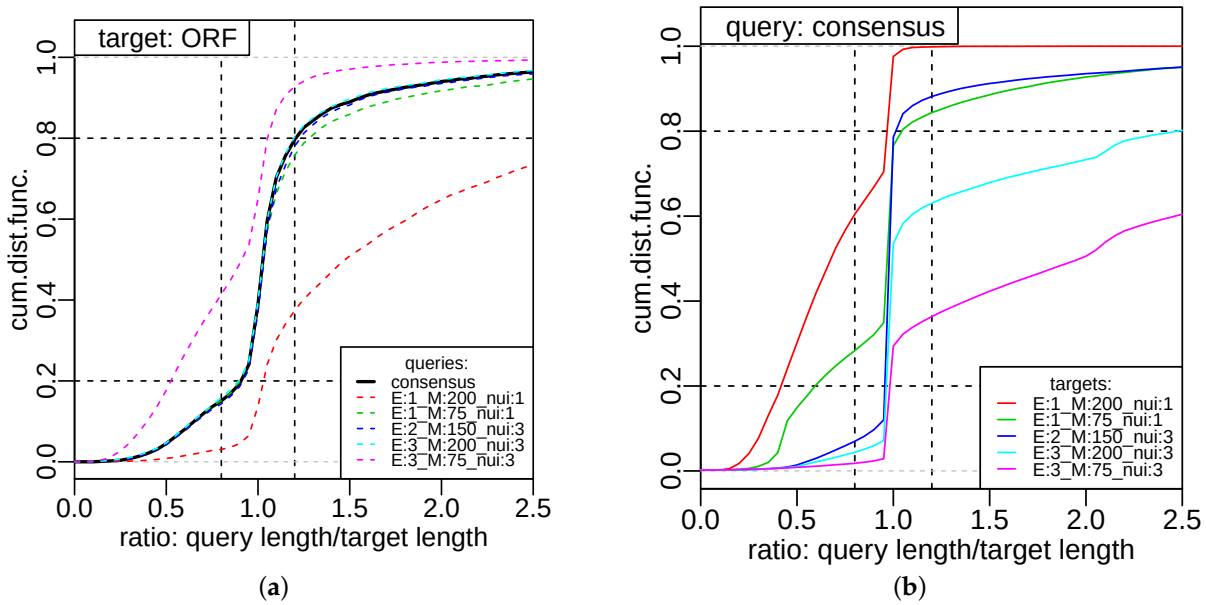
Figure 3. Alternative segmentations of the yeast transcriptome data shown in Figure 1 (here, the coverage time-series is shown as gray-values and the logarithms of the total coverage). Below, we show eight alternative segmentations computed with segmenTier [11] with different parameter settings. The consensus segments, computed for potential $\mathfrak{e}(z) = z^2/2$, match very well with the expectations from visual inspection of the data and from the annotation of yeast the genome (bottom). SRG1 is a non-coding RNA that represses the adjacent SER3 gene by transcriptional interference [35].

In order to evaluate the usefulness of consensus segmentations in a more quantitative manner, we quantify the overlap of segments with annotated coding sequences. To this end, we determine for each CDS C the segment $B(C)$ with the largest Jaccard index and then record the ratio $r(C)$ of segment length and annotation length. In symbols:

$$r(C) := |B(C)|/|C| \quad \text{with} \quad B(C) := \arg\max_{B \in \mathsf{S}} \frac{|B \cap C|}{|B \cup C|}. \tag{22}$$

The cumulative distribution function $\mathrm{cdf}(r)$ computed over a large set of known transcripts C quantifies the congruence between segmentation and annotation. As a reference we use here the transcripts harboring coding sequences annotated in Reference [36]. A perfect overlap between a consensus segment and the annotated transcript is indicated by $r = 1$, $r < 1$ indicates a segment that is shorter and $r > 1$ a segment that is longer then the annotated transcript. Figure 4a shows the cumulative distribution function $\mathrm{cdf}(r)$ for the 5171 CDSs of *S. cerevisae* IFO 0233 for five segmentations with widely different parameters computed with `segmenTier` [11]. The red curve is the consensus over these segmentations. It shows that the consensus segmentation is a robust method: computed from a small sample of distinct segmentations, some of which do not perform particularly well, it performs at least as well as the best individual segmentation obtained from an extensive search of the parameter space in Reference [11]. Discrepancies between annotation and consensus are not only limitations on the segmentation approach but also derive from inaccuracies of the annotation, processing of transcripts, and the complexity of the yeast transcriptome, which harbors abundant overlapping and polycistronic transcripts [37]. The consensus performs as well as the best individual segmentation (according to the benchmark in Reference [11]). Figure 4b shows that the individual segmentations share between about 30% and 70% of the segments with the consensus (corresponding to the height of the vertical jump at $r = 1$), i.e., the consensus does not simply recapitulate any individual input segmentation.

We, therefore, advocate to use the consensus segmentation as a robust and essentially parameter insensitive method for transcriptome analysis in compact genomes.



(**a**) (**b**)

Figure 4. Quantitative evaluation of the consensus of genome-wide transcriptome segmentations of RNA-seq data from *S. cerevisae* from ref. [11]. (**a**) Cumulative distribution function of the length ratios r between overlapping segments and previously annotated ORF transcripts [36]. A ratio of $r = 1$ indicates a good match. The consensus (black solid line) of five representative segmentations (colored dashed lines) by `segmenTier` with widely different parameter settings (as indicated in Figure 2d of Reference [11]) is at least on par with the best individual segmentation. (**b**) Overlap of the consensus with the five different input segmentations. The individual segmentations share between about 30% and 70% of their segments with the consensus (vertical jump at $r = 1$). The consensus was computed with $\mathfrak{e}(z) = z^2/2$.

The consensus segmentation of the transcriptome of *S. cerevisae* IFO 0233 comprises 74,091 segments. After filtering for spacers using the input segmentations [11] and very short segments that most likely correspond to small overlaps and noise in the RNA-seq data, we retained 32,480 segments. Figure 5 shows the distribution of median coverage. Not surprisingly, segments overlapping known protein-coding sequences (CDS) show higher expression levels than other segments. Many segments overlap various types of long non-coding RNAs, such as CUTs and SUTs [38,39]. We also observe many segments with substantial expression levels that so far have remained unannotated, providing a large pool of candidates for novel ncRNAs. Transcriptome segmentation is only the first step towards an accurate and reliable genome annotation. The subsequent processing of the segmentation data, however, is beyond the scope of this present contribution and will be addressed in forthcoming work.

3.3. Consensus Segmentations of Growth Curves

The usefulness of consensus segmentations is by no means limited to transcriptome data or segmentations of genomes. We, therefore, include here also a very different application. The growth of a population of bacteria over time can be quantified by measuring the apparent absorption, usually referred to as OD (optical density), in a spectrophotometer. Growth curves typically show distinct growth regimes: an initial time lag that precedes a phase of exponential growth, which is followed by a deceleration phase that finally settles into saturation, see, e.g., Reference [40]. These can be separated by approximating the time course of $\log OD$ by a sequence of line segments, i.e., as a continuous, piece-wise linear function. The corresponding approximation problem is again a segmentation problem that can be solved by dynamic programming [28]. Empirically, one observes that resulting segmentations are quite sensitive to small difference in the growth curves. We show here

that the consensus segmentation is a convenient way to extract a robust estimate for the duration of the different phases.

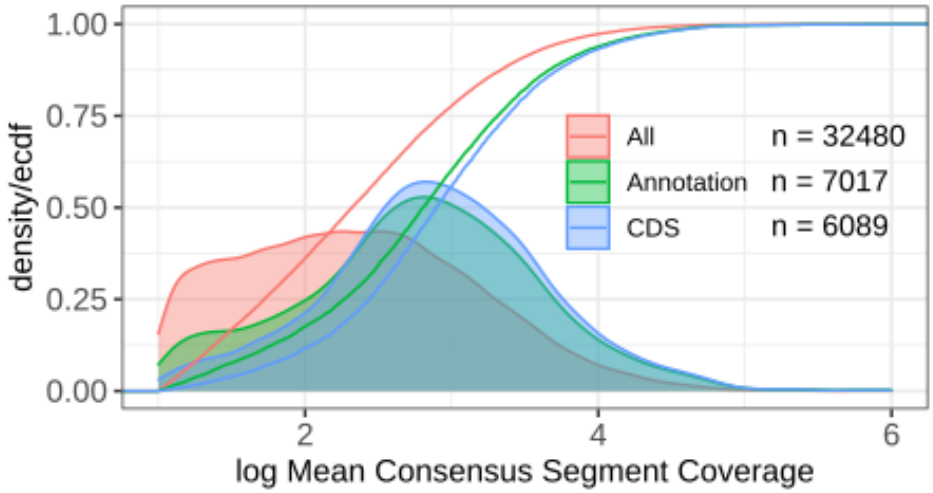


Figure 5. Distribution of RNA expression across the consensus segmentation of *S. cerevisae* IFO 0233. We distinguish segments that overlap a coding sequence (CDS) or another annotation item (Annotation) present in the current genome annotation taken from SGD (Saccharomyces Genome Database), and unannotated segments (All). An overlap of at least 30% with an annotation item was required. Densities are normalized to 1 for each class. Cumulative distributions are superimposed.

In Figure 6, we compare the growth curves of four *Escherichia coli* cultures grown in minimal glucose medium at 37 °C. The R package `dpseg` [41] was used to segment the individual growth curves. This package also uses a dynamic programming approach. Instead of fixing the number of segments as in Reference [28], it uses a penalty parameter to adjust the resolution of the segmentation. Considering the consensus of the individual segmentations and the variation of the breakpoints across the replicates provides a more realistic view of that data compared to the segmentation of an averaged growth curve.

3.4. Refinement Conjecture

Theorem 2 states that the union segmentation $\hat{\mathbf{S}}$ refines the consensus segmentation $\mathbf{C}$ for the class of boundedly convex consensus functions. Numerical simulations strongly suggest that this is true also for many potentials with increasing positive curvature, despite the failure of the proof technique for the general case. We simulated 10 segmentations of length 50 and a maximum of 10 segments, using the base-R *sample* function to randomly choose breakpoints in a given range. Figure 7 shows consensus segmentations for six potential functions from negentropy to exponential.

We found that the consensus segmentation only contained breakpoints that are present in at least one of the input segmentation. This suggests:

Conjecture 1. $\hat{\mathbf{S}}$ *appears to refine* $\mathbf{C}$ *for all superadditive potentials and possibly even for all convex potentials.*

This "Refinement Conjecture" is of considerable practical use. If true, (assumed as a heuristic), it reduces the computational effort to $O(s^2)$ where s is the number of break points in the input segmentations. We further observed a trend for faster-growing potentials to yield more and, thus, shorter consensus segments. This is consistent with the fact that the bound $\sqrt[1+\alpha]{2}|B|$ on the length of the consensus intervals in the argument leading up Corollary 5 decreases with the exponent α in polynomial potentials.

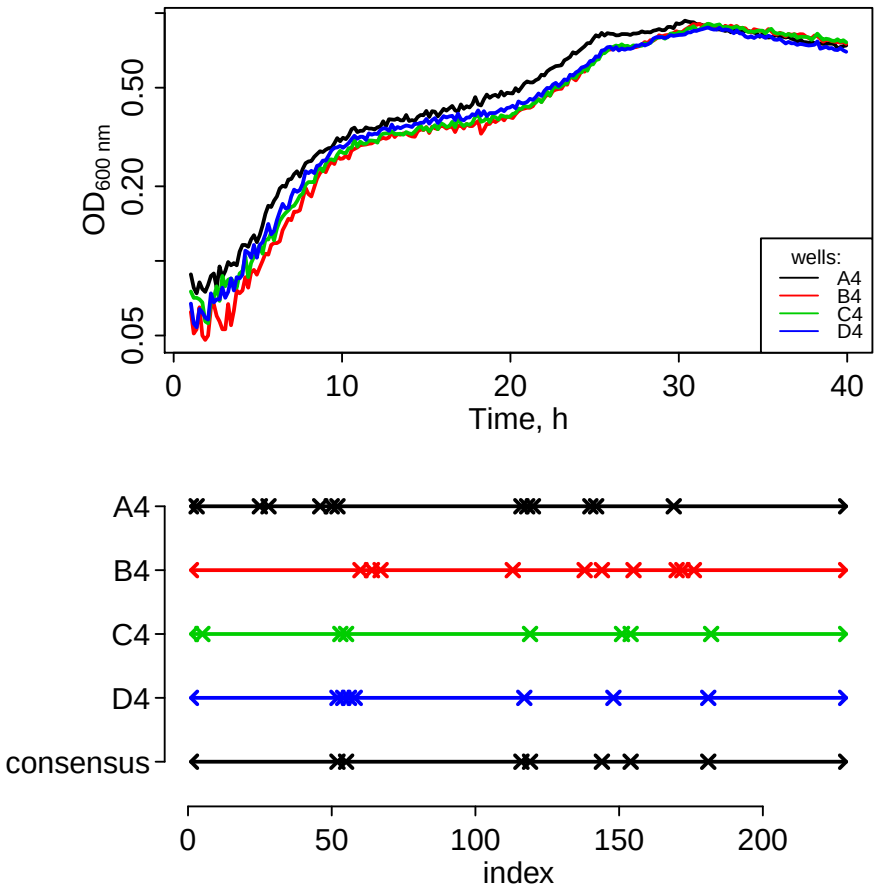


Figure 6. Four *Escherichia coli* cultures were grown at identical conditions (four replicates in a larger experiment) in M9 medium with 0.2% glucose at 37 °C in a BMG Clariostar platereader and the optical density at 600 nm, $\ln(\mathrm{OD}_{600\,\mathrm{nm}}$ was measured every 10 minutes. The growth curves of each of the four replicates were segmented into intervals with constant slope by the `dpseg` algorithm with the default jump penalty parameter $P = 0$ [41].

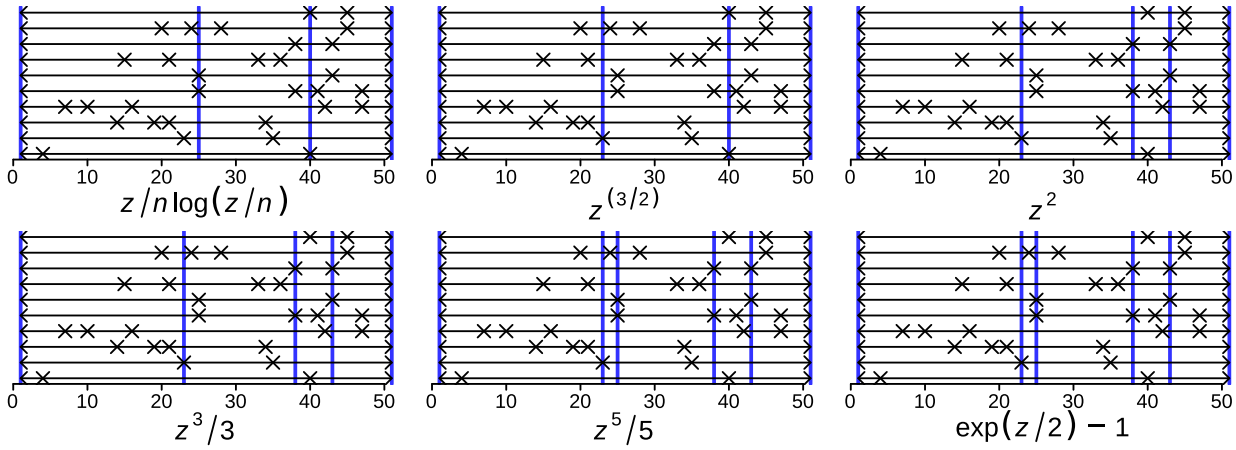


Figure 7. Consensus segmentation (shown by blue vertical lines) for a collection of 10 random segmentations with equal weights for six different potential functions $\mathfrak{e}(z)$. Note that only breakpoints of the input segmentations (marked by $\times$ appear in the consensus segmentation.

4. Concluding Remarks

In this work, we have extended and generalized previous work [24,33] on the segmentation aggregation problem. We showed that for the class of boundedly convex potential functions, including negentropies and powers $z^{1+\alpha}$ with $0 < \alpha \leq 1$, all consensus breakpoints are breakpoints in at least one of the contributing segmentations. Furthermore, we showed that for all superadditive potentials, consensus segments cannot be longer than twice the longest input segment. This bound allows a further reduction of the computational effort.

Consensus segmentations as described here pertain to two major application scenarios: (i) Reconciliation of segmentations of multi-dimensional data, comprising, e.g., independent measurements, such as biological or technical replicates, or measurements of different quantities, e.g., different histone modifications. (ii) Reconciliation of segmentations of the same data set produced with different similarity measures. In principle, it is also possible to compute the consensus segmentation of different segmentations produced, e.g., with randomized algorithms or different heuristics using the same similarity measures. A major advantage of consensus segmentations is that they can be computed without specific information about the data underlying the input segmentations. Such knowledge is not needed because the segmentation aggregation problem depends only on the distance function D as a "parameter". Empirically, we found that variations of the distance functions have only very moderate consequence on the consensus segmentation.

In simulations, we found strong support for the Refinement Conjecture. This provides support for approaches that utilize a dynamic programming segmentation method to select be best segmentation from the union of segmentations that are computed with different heuristics. Such a scheme has been proposed in Reference [42]. In this manner, one can potentially achieve a substantial gain in computational efficiency compared to the full dynamic programming segmentation. The `C-KS` approach [12] also restricts itself to the union segmentation.

We considered two very different application scenarios. In applications to transcriptome data consensus segmentations have the potential to substantially improve annotations. A particular strength of the consensus approach is, by highlighting differences that are consistent between data tracks, the ability to identify processing-related boundaries. This is of particular interest in organisms with operons, poly-cistronic primary transcripts, or no expression-free gaps between genes. In all these cases, it becomes difficult and often impossible to distinguish transcriptional units from patterns of mapped RNA-seq reads alone. Here, we have used data from yeast strain IFO 0233, which has been used previously to illustrate transcriptome segmentation in Reference [11]. We have seen that the consensus segmentation provides a robust prediction of transcriptomic units from a moderate number of individual segmentations with very different parameters. We obtained thousands of segments that may correspond to the non-coding transcripts in *S. cerevisae* IFO 0233. Since the present contribution is intended to describe the method of consensus segmentation and its mathematical justification, we will report elsewhere on a comprehensive analysis of the IFO 0233 transcriptome.

Consensus formation is also of use to aggregate data from biological replicates. As an example, we showed that consensus segmentations of growth curves can be used to robustly determine distinct growth phases.

The consensus segmentation methods incorporate weights that refer to input segmentations. This feature can be used, for instance, to weight individual transcriptome data by coverage. In the case of growth functions, weights may be chosen to decrease with average measurement error, quantified, e.g., as average deviation from the linear fit. It would also be of interest to associate weights with individual segments. This can certainly be done in the context of the Boundary Mover's distance. Whether this can be also be achieved in the potential-based approach, and to what extent the mathematics results of this contribution will remain intact, however, is a question for future research.

We observed that consensus segmentations are quite robust w.r.t. to the choice of the potential on real data, while we observed a trend towards shorter consensus segments with increasing α for the power potentials $e(z) = z^{\alpha-1}$ on i.i.d. random data. Conceptually, consensus segmentations based on the comparison of segments are designed to handle essentially arbitrary heterogeneity along the time or genome coordinate, while breakpoint-centered approaches, such as `C-KS`, need to rely on statistical regularities of true breakpoints. In order to assess the utility of different potentials $\mathfrak{e}(\,.\,)$ and dissimilarity measures $D(\,.\,,\,.\,)$, and to compare the segment-centered dynamic programming approach with breakpoint-centered alternatives, a principled way of benchmarking consensus segmentation methods will be necessary. This will require, in particular, the development of a simulator for correlated segmentations that mimic characteristics of different types of underlying data. At present, such tools are not available.

Our analysis of consensus segmentations suggests several avenues for future research. From a theoretical point of view, the most immediate open problem is the Refinement Conjecture, and the characterization of the potential function and—more generally—the dissimilarity measures for which the union segmentation refines the consensus segmentation. This is also of practical relevance since the recursions can be restricted to breakpoints of the union segmentation in such cases; see Equation (15). In addition, a more detailed understanding of the consensus segmentation would be useful. For instance, are there potentials or dissimilarities that guarantee a breakpoint in the consensus within every interval that contains a breakpoint in every input segmentation, or in the (weighted) majority of the input segmentations? Such results could immediately be used to limit the scope of j in Equation (8). More generally, one may ask whether the idea of consensus segmentation can give rise to useful ways of measuring the local accuracy or reliability of consensus and/or input break points.

Author Contributions: All authors jointly contributed the conceptualization of the study, the theoretical results, the interpretation of the results, and the writing of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the German Federal Ministry for Education and Research (BMBF 031A538B as part of de.NBI and BMBF 031L0164C, RNAProNet, to P.F.S.), the *Deutsche Forschungsgemeinschaft* (DFG proj. nos. AX 84/4-1 and STA 850/30-1), and the Lebanese Association for Scientific Research (LASeR). H.S. receives a *Landesgraduiertenstipendium* of the Free State of Saxony.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The RNA-seq data for the example in Figures 1 and 3 (SRG1 ncRNA and SER3 protein coding gene) is available via the *segmenTier* R package (object `tsd` can be loaded with `data(primseg436)`) and the bacterial growth curves in Figure 6 via the *dpseg* package (`oddata` is available after loading the package). The genome-wide RNA-seq data will be made available at a public repository with a full report on the data (in progress, work by RM, PFS and DBM). The segmentations of the yeast IFO 0233 transcriptomes are available as Supplemental File in csv format.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the scientific content of this work.

Abbreviations

CDS	Coding sequence
CUT	Cryptic unstable transcript
ncRNA	non-coding RNA
SGD	Saccharomyces Genome Database
SUT	Stable uncharacterized transcripts
ORF	Open reading frame
i.i.d.	independent and identically distributed

Glossary of mathematical symbols:

$A, B, \ldots$	Segments in a segmentation
$[i, j]$	Interval of from to i to j (inclusive)
$\mathbf{C}$	Consensus segmentation (of a set of segmentations)
$\mathbf{S_q}$	(Input) segmentation
w_q	Weight of an input segmentation
$\hat{\mathbf{S}}$	Union segmentation (of a set of segmentations)
$D(.,.)$	Distance (dissimilarity) between segmentations
$D_B(.\|.)$	Boundary movers distance
F_k	Score of a partial segmentation on $[1, k]$
$\mathfrak{e}(.)$	Potential of an interval
$\Delta[i, j]$	Score on a consensus interval
$\delta_<(i)$	Score contribution of segments ending no later than i
$\delta_\leq(i)$	Score contribution of segments beginning no later than i
$\delta_<^\cap(i)$	Score contribution of r.h.s. part a segment spanning i
$\delta_>^\cap(i)$	Score contribution of l.h.s. part a segment spanning i
$\delta^*(i, j)$	score correction for segments beginning before i and ending after j

References

1. Pirooznia, M.; Goes, F.S.; Zandi, P.P. Whole-genome CNV analysis: advances in computational approaches. *Front. Genet.* **2015**, *6*, 138. [CrossRef]
2. Yen, A.; Kellis, M. Systematic chromatin state comparison of epigenomes associated with diverse properties including sex and tissue type. *Nat. Commun.* **2015**, *6*, 7973. [CrossRef]
3. Zeller, G.; Henz, S.R.; Laubinger, S.; Weigel, D.G.R. Transcript Normalization and Segmentation of Tiling Array Data. *Pac. Symp. Biocomput.* **2008**, *13*, 527–538.
4. Hardcastle, T.J.; Kelly, K.A.; Baulcombe, D.C. Identifying small interfering RNA loci from high-throughput sequencing data. *Bioinformatics* **2012**, *28*, 457–463. [CrossRef] [PubMed]
5. Bischler, T.; Kopf, M.; Voß, B. Transcript mapping based on dRNA-seq data. *BMC Bioinform.* **2014**, *15*, 122. [CrossRef] [PubMed]
6. David, L.; Huber, W.; Granovskaia, M.; Toedling, J.; Palm, C.J.; Bofkin, L.; Jones, T.; Davis, R.W.; Steinmetz, L.M. A high-resolution map of transcription in the yeast genome. *Proc. Natl. Acad. Sci. USA* **2006**, *103*, 5320–5325. [CrossRef]
7. Danford, T.; Dowell, R.; Agarwala, S.; Grisafi, P.; Fink, G.; Gifford, D. Discovering regulatory overlapping RNA transcripts. *J. Comput. Biol.* **2011**, *18*, 295–303. [CrossRef] [PubMed]
8. Braun, J.V.; Müller, H.G. Statistical methods for DNA sequence segmentation. *Stat. Sci.* **1998**, *13*, 142–162. [CrossRef]
9. Elhaik, E.; Graur, D.; Josić, K. Comparative Testing of DNA Segmentation Algorithms Using Benchmark Simulations. *Mol. Biol. Evol.* **2010**, *27*, 1015–1024. [CrossRef]
10. Girimurugan, S.B.; Liu, Y.; Lung, P.Y.; Vera, D.L.; Dennis, J.H.; Bass, H.W.; Zhang, J. iSeg: An efficient algorithm for segmentation of genomic and epigenomic data. *BMC Bioinform.* **2018**, *19*, 131. [CrossRef]
11. Machné, R.; Murray, D.B.; Stadler, P.F. Similarity-Based Segmentation of Multi-Dimensional Signals. *Sci. Rep.* **2017**, *7*, 12355. [CrossRef] [PubMed]
12. Toloşi, L.; Theißen, J.; Halachev, K.; Hero, B.; Berthold, F.; Lengauer, T. A method for finding consensus breakpoints in the cancer genome from copy number data. *Bioinformatics* **2013**, *29*, 1793–1800. [CrossRef] [PubMed]
13. Segal, M.R.; Wiemels, J.L. Clustering of Translocation Breakpoints. *J. Am. Stat. Assoc.* **2002**, *97*, 66–76. [CrossRef]
14. Rand, W.M. Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **1971**, *66*, 846–850. [CrossRef]
15. Fowlkes, E.B.; Mallows, C.L. A method for comparing two hierarchical clusterings. *J. Am. Stat. Assoc.* **1983**, *78*, 553–569. [CrossRef]
16. Ben-Hur, A.; Elisseeff, A.; Guyon, I. A stability based method for discovering structure in clustered data. *Pac. Symp. Biocomput.* **2002**, *7*, 6–17.
17. Hubert, L.; Arabie, P. Comparing partitions. *J. Classif.* **1985**, *2*, 193–218. [CrossRef]
18. Mirkin, B. *Mathematical Classification and Clustering*; Kluwer Academic Press: Dordrecht, The Netherlands, 1996.
19. Van Dongen, S. *Performance Criteria for Graph Clustering and Markov Cluster Experiments*; Technical Report; Centrum voor Wiskunde en Informatica: Amsterdam, The Netherlands, 2000.
20. Mirkin, B.G. On the Problem of Reconciling Partitions. In *Quantitative Sociology: International Perspectives on Mathematical and Statistical Modeling*; Blalock, H.M., Aganbegian, A., Borodkin, F.M., Boudon, R., Capecchi, V., Eds.; Academic Press: New York, NY, USA, 1975; pp. 441–449.
21. Barthélemy, J.; Leclerc, B. The median procedure for partitions. In *Partitioning Data Sets*; Cox, I., Hansen, P., Julesz, B., Eds.; American Mathematical Society: Providence, RI, USA, 1995; Volume 19, pp. 3–34. [CrossRef]
22. Křivánek, M.; Morávek, J. NP-hard problems in hierarchical-tree clustering. *Acta Inform.* **1986**, *23*, 311–323. [CrossRef]
23. Wakabayashi, Y. The complexity of computing medians of relations. *Resenhas IME-USP* **1998**, *3*, 323–349.

24. Mielikäinen, T.; Terzi, E.; Tsaparas, P. Aggregating Time Partitions. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; Eliassi-Rad, T., Ungar, L., Craven, M., Gunopulos, D., Eds.; Association for Computing Machinery: New York, NY, USA, 2006; pp. 347–356. [CrossRef]
25. Meilă, M. Comparing Clusterings: An Axiomatic View. In *Machine Learning, Proceedings of the Twenty-Second International Conference*; De Raedt, L., Wrobel, S., Eds.; Association for Computing Machinery: New York, 2005; pp. 577–584. [CrossRef]
26. Meilă, M. Comparing clusterings by the variation of information. In *Learning Theory and Kernel Machines*; Schölkopf, B., Warmuth, M.K., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2777, pp. 173–187. [CrossRef]
27. Vingron, M.; Sibbald, P.R. Weighting in sequence space: A comparison of methods in terms of generlized sequences. *Proc. Natl. Acad. Sci. USA* **1993**, *90*, 8777–8781. [CrossRef] [PubMed]
28. Bellman, R. On the approximation of curves by line segments using dynamic programming. *Commun. ACM* **1961**, *4*, 284–286. [CrossRef]
29. Bai, J.; Perron, P. Computation and analysis of multiple structural change models. *J. Appl. Econom.* **2002**, *18*, 1–22. [CrossRef]
30. Fragkou, P.; Petridis, V.; Kehagias, A. A Dynamic Programming Algorithm for Linear Text Segmentation. *J. Intell. Inf. Syst.* **2004**, *23*, 179–197. [CrossRef]
31. Picard, F.; Robin, S.; Lavielle, M.; Vaisse, C.; Daudin, J. A statistical approach for CGH microarray data analysis. *BMC Bioinform.* **2005**, *6*, 27. [CrossRef]
32. Huber, W.; Toedling, J.; Steinmetz, L.M. Transcript mapping with high-density oligonucleotide tiling arrays. *Bioinformatics* **2006**, *22*, 1963–1970. [CrossRef] [PubMed]
33. Terzi, E. Problems and Algorithms for Sequence Segmentations. Ph.D. Thesis, Department of Computer Science Series of Publications A Report A-2006-5, University of Helsinki, Helsinki, Finland, 2006.
34. Haiminen, N.H.; Mannila, H.; Terzi, E. Comparing segmentations by applying randomization techniques. *BMC Bioinform.* **2007**, *8*, 171. [CrossRef] [PubMed]
35. Martens, J.A.; Laprade, L.; Winston, F. Intergenic transcription is required to repress the *Saccharomyces cerevisiae* SER3 gene. *Nature* **2004**, *429*, 571–574. [CrossRef]
36. Xu, Z.; Wei, W.; Gagneur, J.; Perocchi, F.; Clauder-Munster, S.; Camblong, J.; Guffanti, E.; Stutz, F.; Huber, W.; Steinmetz, L.M. Bidirectional promoters generate pervasive transcription in yeast. *Nature* **2009**, *457*, 1033–1037. [CrossRef]
37. Pelechano, V.; Wei, W.; Steinmetz, L.M. Extensive transcriptional heterogeneity revealed by isoform profiling. *Nature* **2013**, *497*, 127–131. [CrossRef] [PubMed]
38. Parker, S.; Fraczek, M.G.; Wu, J.; Shamsah, S.; Manousaki, A.; Dungrattanalert, K.; de Almeida, R.A.; Invernizzi, E.; Burgis, T.; Omara, W.; et al. Large-scale profiling of noncoding RNA function in yeast. *PLoS Genet.* **2018**, *14*, e1007253. [CrossRef]
39. Till, P.; Mach, R.L.; Mach-Aigner, A.R. A current view on long noncoding RNAs in yeast and filamentous fungi. *Appl. Microbiol. Biotech.* **2018**, *102*, 7319–7331. [CrossRef] [PubMed]
40. Hall, B.G.; Acar, H.; Nandipati, A.; Barlow, M. Growth Rates Made Easy. *Mol. Biol. Evol.* **2014**, *31*, 232–238. [CrossRef] [PubMed]
41. Machné, R.; Stadler, P.F. `dpseg`: Piecewise Linear Segmentation by Dynamic Programming. R Package Version 0.1.2. 2020. Available online: https://cran.r-project.org/web/packages/dpseg/ (accessed on 1 December 2020).
42. Pierre-Jean, M.; Rigaill, G.; Neuvial, P. Performance evaluation of DNA copy number segmentation methods. *Brief. Bioinform.* **2015**, *16*, 600–615. [CrossRef] [PubMed]