



# Efficient Analysis of Annotation Colocalization Accounting for Genomic Contexts

Askar Gafurov<sup>1</sup>, Tomáš Vinař<sup>2</sup>, Paul Medvedev<sup>3,4,5</sup>, and Broňa Brejová<sup>1</sup>(✉)

<sup>1</sup> Department of Computer Science, Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava, Bratislava, Slovakia  
[brejova@fmph.uniba.sk](mailto:brejova@fmph.uniba.sk)

<sup>2</sup> Department of Applied Informatics, Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava, Bratislava, Slovakia

<sup>3</sup> Department of Computer Science and Engineering,  
The Pennsylvania State University, State College, USA

<sup>4</sup> Huck Institutes of the Life Sciences, The Pennsylvania State University,  
State College, USA

<sup>5</sup> Department of Biochemistry and Molecular Biology,  
The Pennsylvania State University, State College, USA

**Abstract.** An annotation is a set of genomic intervals sharing a particular function or property. Examples include genes, conserved elements, and epigenetic modifications. A common task is to compare two annotations to determine if one is enriched or depleted in the regions covered by the other. We study the problem of assigning statistical significance to such a comparison based on a null model representing two random unrelated annotations. To incorporate more background information into such analyses and avoid biased results, we propose a new null model based on a Markov chain which differentiates among several genomic contexts. These contexts can capture various confounding factors, such as GC content or sequencing gaps. We then develop a new algorithm for estimating p-values by computing the exact expectation and variance of the test statistic and then estimating the p-value using a normal approximation. Compared to the previous algorithm by Gafurov et al., the new algorithm provides three advances: (1) the running time is improved from quadratic to linear or quasi-linear, (2) the algorithm can handle two different test statistics, and (3) the algorithm can handle both simple and context-dependent Markov chain null models.

We demonstrate the efficiency and accuracy of our algorithm on synthetic and real data sets, including the recent human telomere-to-telomere assembly. In particular, our algorithm computed p-values for 450 pairs of human genome annotations using 24 threads in under three hours. The use of genomic contexts to correct for GC-bias also resulted in the reversal of some previously published findings.

**Availability.** The software is freely available at <https://github.com/fmfi-compbio/mcdp2> under the MIT licence. All data for repro-

ducibility are available at <https://github.com/fmfi-compbio/mcdp2-reproducibility>.

**Keywords:** genome annotation · colocalization · Markov chains

## 1 Introduction

Recent years have brought rapid growth in the number of different assays that can extract genome-scale functional information. This has led to growing collections of genome annotations; for example in the UCSC Genome browser, the GRCh38 human genome assembly currently features 136 different annotation tracks, many of which have multiple subtracks. In this work, we provide new models and algorithms for annotation colocalization analysis, where the goal is to determine if one input annotation is significantly colocalized with regions covered by another annotation. Such analyses may hint at possible connections between biological processes governing individual annotations (e.g. histone modification H3K4me3 sites are colocalized with promoter regions, and H3K4me3 indeed plays a role in gene transcription regulation [10]).

Mathematically, we view a genome annotation as a set of non-overlapping chromosomal intervals. Given two annotations, query  $Q$  and reference  $R$ , we consider two widely-used colocalization statistics. The *overlap statistic* is the number of intervals in  $R$  that intersect with at least one interval in  $Q$ . The *shared bases statistic* is the number of positions in the genome covered by both  $R$  and  $Q$ . However, even randomly generated annotations will share bases or have overlapping intervals by chance. In order to ascertain statistical significance of the observed statistic, its p-value needs to be computed under a suitable null hypothesis. Until very recently, all the methods [3, 6, 11, 17–19] were limited by having a null hypothesis that either does not properly model the data or its p-value computation does not scale to annotations of human-sized genomes.

Recently, Gafurov et al. [5] proposed an alternative null hypothesis in which the annotation is produced by a two-state Markov chain. The algorithm, called MCDP, was a substantial improvement in time and memory over previous approaches. However, it is quadratic in the number of reference intervals and still takes several hours for the human exon reference annotation. It thus remains time-prohibitive to compare many pairs of annotations against each other.

Another limitation of MCDP as well as other approaches is that two unrelated annotations may appear to be colocalized because they are each colocalized with another genomic feature [12]. For example, two annotations may appear colocalized simply due to their prevalence in high-GC regions, even though they are not related. More generally, different regions of the genome can be thought of as providing different background to the null model, and we think of these various backgrounds as partitioning the genome into *contexts*. Accounting for contexts in calculating p-values is important to limit false associations, yet this capability is limited or absent in existing tools.

In this paper, we propose a model and an algorithm to overcome these scalability and accuracy barriers. Our first contribution is a new algorithm MCDP2

for estimating p-values, which is linear in the number of reference intervals. To demonstrate the scalability of our algorithm, we considered 10 reference annotations, corresponding to different types of repeats in the human genome, and 45 query annotations, corresponding to epigenetic modifications in different cell lines. MCDP2 computed p-values for all 450 pairs using 24 threads in under 2 hours for the number of overlaps and 3 hours for the number of shared bases.

Our second contribution expands the modeling capability of the Markov chain null hypothesis so that it takes into account genomic context and thus captures various confounding factors influencing annotation colocalization. Unlike previous approaches [11], our model is able to handle annotation intervals that span class boundaries. We demonstrate the importance of modeling the genomic context by re-analyzing colocalization of copy number deletions with various gene classes [21] and find that adding a genome context in fact reverses some of the previous conclusions. In one striking example, the set of all exons appears enriched for overlap with copy number losses but enrichment turns into depletion after taking into account gaps and GC content. We also compare the colocalization of epigenetic marks with subtelomeric repeats on the new human telomere-to-telomere assembly [8], using a genome context to compare enrichment between two classes of repeats.

*Related Work.* Several null hypotheses for colocalization statistics have been considered prior to the Markov chain model [5]. Some lend themselves to fast and simple statistical tests (e.g. Fisher’s exact test) but do not capture relevant properties of the data. For example, one can assume that all positions in the query annotation are chosen uniformly at random [3, 19]. However, this does not capture either the integrity of intervals or their length distribution. A more faithful option is the permutational null hypothesis (also called gold null hypothesis [5]), which reshuffles the query intervals while maintaining their lengths [6, 11, 17]. Computing the exact p-values for the overlap statistic in this model is NP-hard [5], and the only known efficient algorithms are either inaccurate or impractical for human-sized annotations [18]. Sampling approaches can be used, but their accuracy is directly proportional to the number of samples, making it difficult to estimate small p-values. With these limitations, it was impossible to compute small p-values for human-sized genomes while having a null hypothesis that is faithful to the data.

Accounting for genomic contexts has also been considered but most previous approaches [3, 6, 15, 19] are only able to account for contexts which are completely inadmissible to annotations (e.g. assembly gaps, which are unassembled regions of the genome). A notable exception is GAT [11], which splits a genome into multiple contexts and analyzes colocalization in each context independently. However, this approach does not satisfactorily handle intervals that span context boundaries, which become prevalent when the contexts are short.

## 2 Methods

In this section, we define our context-aware Markov chain null model ( $\mathcal{H}_0^{\text{context}}$ ) and provide an overview of our algorithm for efficient estimation of p-values

under this model. We first present our results on a single chromosome; an extension to multiple chromosomes is discussed in Sect. 2.5.

We will denote the chromosome length as  $L$  and use 0-based coordinates. An *annotation* is a set of intervals contained in  $[0, L)$  so that each two intervals are disjoint and separated by at least one base. By  $|Q|$  we denote the number of intervals in annotation  $Q$ . We will represent an annotation either as a list of half-open intervals ordered from left to right  $Q = ([b_1, e_1), \dots, [b_{|Q|}, e_{|Q|})$ , or as a binary sequence  $Q = (Q_0, Q_1, \dots, Q_{L-1})$ , where  $Q_i$  is 1 if position  $i$  is covered by one of the intervals and 0 otherwise.

Let  $R$  and  $Q$  be two annotations, denoted as the reference and the query, respectively. In this setting, a test statistic is a function that measures the extent to which  $R$  and  $Q$  are colocalized. We will consider two concrete test statistics. One is the number of overlaps  $K(R, Q)$ , which is defined as the number of intervals in  $R$  that overlap some interval in  $Q$ . The other is the number of shared bases  $B(R, Q)$ , which is defined as the number of bases in the genome covered by both  $R$  and  $Q$ . Let  $F$  be the distribution of the query annotation under the null hypothesis of the query being generated independently of the reference annotation. Given some test statistic  $A(R, Q)$ , we are interested in computing the p-value measuring the statistical significance of enrichment of  $Q$  with respect to  $R$ , that is, probability  $\Pr_{Q' \sim F}[A(R, Q') \geq A(R, Q)]$ .

Our algorithm is based on the observation that the distribution of the test statistics is in most realistic scenarios well approximated by the normal distribution (see Sects. 3 and 4). Therefore, instead of computing the full probability mass function (PMF), we compute only its exact mean and variance and use them as the parameters of the normal distribution. This means that we calculate the p-value by first computing the *Z-score*, which is the number of standard deviations that  $A(R, Q)$  is above the expected value, under the null. Formally,

$$Z_A(R, Q) = \frac{A(R, Q) - \mathbb{E}_{Q' \sim F}[A(R, Q')]}{\sqrt{\text{Var}_{Q' \sim F}[A(R, Q')].}$$

Under the assumption that the statistic is normally distributed, the desired p-value is then simply  $1 - \Phi(Z_A(R, Q))$ , where  $\Phi$  is the cumulative distribution function of the standard normal distribution. Analogously, the p-value for the statistical significance of depletion is computed as  $\Phi(Z_A(R, Q))$ .

In Sect. 2.1, we describe our context-aware Markov chain model for generating random annotations and then use it to formally define the  $\mathcal{H}_0^{\text{context}}$  null model in Sect. 2.2. In Sects. 2.3 and 2.4, we outline our algorithm for computing the mean and variance of the overlap and shared bases test statistics. Finally, we describe how our model naturally extends to multiple chromosomes (Sect. 2.5).

## 2.1 A Generative Model

An annotation of a chromosome of length  $L$  can be generated by running a two-state Markov chain for  $L$  steps. The state at step  $i$  indicates whether the annotation includes position  $i$  on the chromosome. The lengths of the generated

intervals and of the gaps between them are known to be geometrically distributed in this model, and the transition probabilities of the Markov chain dictate the expected values of these two distributions [13]. The Markov chain generative model makes many properties easy to derive and fast to compute [5], and so we build upon it in this work.

We want to use such a generative model to test if a given query annotation  $Q$  behaves as if it was “randomly shuffled” on the chromosome. To this end, we set the parameters of the Markov chain so that the expected interval lengths and gaps between them match what is observed in the query  $Q$ . However, this does not allow us to incorporate background knowledge of the chromosome; i.e., some regions of the genome may be *a priori* more likely to generate an interval.

We therefore introduce the notion of genome contexts. Given a finite set of class labels  $\Lambda$ , a *genome context* is a mapping  $\phi : \{0, \dots, L - 1\} \rightarrow \Lambda$  of each position on the genome onto a class label (e.g.  $\Lambda = \{\text{gap}, \text{non-gap}\}$ ). This mapping partitions the genome into several segments with the same class assigned. We will refer to the positions where the class differs from the class label at the previous position as to *class boundaries*. We assume throughout the paper that a context is represented as a sequence of class boundary positions with the corresponding class labels, sorted in an increasing order by positions.

Our generative model allows each context class to have its own Markov chain, i.e. its own distribution of interval lengths and gaps. An annotation is then generated by iterating over the genome positions from left to right, and at each position  $i$  transitioning to the next state of the Markov chain according to the transition probabilities of the class at position  $i$  (see Fig. 1). A similar model was proposed by Burge and Karlin [1] for gene finding; their hidden Markov model uses different transition and emission probabilities based on GC content in the current window of the genome.

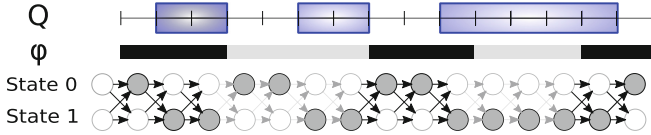
**Definition 1.** A context-aware Markov chain is a pair  $(\phi, \mathbf{T})$ , where  $\phi$  is a genome context and  $\mathbf{T} : \Lambda \rightarrow \mathbb{R}^{2 \times 2}$  is a mapping that provides a transition probability matrix for each context class. The context-aware Markov chain  $(\phi, \mathbf{T})$  generates a sequence of states  $(s_{-1}, s_0, \dots, s_{L-1}) \in \{0, 1\}^{L+1}$  with probability

$$\Pr[(S_{-1}, S_0, \dots, S_{L-1}) = (s_{-1}, s_0, \dots, s_{L-1})] = \bar{\pi}_{s_{-1}} \cdot \prod_{i=0}^{L-1} \mathbf{T}(\phi(i))_{s_{i-1}, s_i},$$

where  $\mathbf{T}(\phi(i))_{s, s'}$  is the probability of transition from state  $s$  to state  $s'$  in context class  $\phi(i)$ , and  $\bar{\pi}_s$  is the probability of state  $s$  in the stationary distribution of the Markov chain with transition probabilities  $\mathbf{T}(\phi(0))$ . Namely,

$$\bar{\pi} = \left( \frac{\mathbf{T}(\phi(0))_{1,0}}{1 - \mathbf{T}(\phi(0))_{0,0} + \mathbf{T}(\phi(0))_{1,0}} \quad \frac{1 - \mathbf{T}(\phi(0))_{0,0}}{1 - \mathbf{T}(\phi(0))_{0,0} + \mathbf{T}(\phi(0))_{1,0}} \right)$$

Note that we are indexing vectors and matrices starting from 0 in order to make the formulas more readable. The produced binary sequence of states  $(s_0, \dots, s_{L-1})$  can be viewed as an annotation of a genome of size  $L$ . State  $s_{-1}$



**Fig. 1.** An example of a query annotation  $Q = \{[1, 3], [5, 7], [9, 14]\}$  and the corresponding sequence of states of the context-aware Markov chain that induces the annotation. Genome context  $\phi$  is shown with black and gray colors corresponding to two distinct class labels. The same colors are also used on transition arrows between successive states of the Markov chain, as the transition probabilities depend on the genome context.

is added to the start for notational convenience, as we will often refer to the state preceding the start of an interval. The distribution of the random vector of generated states  $(S_{-1}, S_0, \dots, S_{L-1})$  will be denoted as  $C(\phi, \mathbf{T})$ . We will use the same notation to denote the distribution of the induced annotation.

## 2.2 The Context-Aware Markov Chain Null Model

The generative model above serves as a basis for our null model, which we call  $\mathcal{H}_0^{\text{context}}$ . Given a context  $\phi$  and a query annotation  $Q = (Q_0, \dots, Q_{L-1})$ , we first need to find the transition probabilities  $\mathbf{T}_Q$  that would maximize the probability of the context-aware Markov chain  $(\phi, \mathbf{T}_Q)$  generating  $Q$ . This is achieved through the standard approach of training Markov chains by counting transition frequencies [4]. In particular, for each class, we count the number of times each possible state transition occurs in  $Q$ . Formally,

$$n_{\lambda, s, s'} = 1 + \sum_{i=1}^{L-1} \mathbb{1}((\phi(i) = \lambda) \text{ and } (s = Q_{i-1}) \text{ and } (s' = Q_i)),$$

where  $\mathbb{1}$  is the indicator function which evaluates to 1 if the logical expression inside is true and 0 otherwise. A pseudocount 1 is added to avoid zero probabilities [4]. The transition matrix is then defined from these counts as

$$\mathbf{T}_Q(\lambda) = \begin{pmatrix} \frac{n_{\lambda, 0, 0}}{n_{\lambda, 0, 0} + n_{\lambda, 0, 1}} & \frac{n_{\lambda, 0, 1}}{n_{\lambda, 0, 0} + n_{\lambda, 0, 1}} \\ \frac{n_{\lambda, 1, 0}}{n_{\lambda, 1, 0} + n_{\lambda, 1, 1}} & \frac{n_{\lambda, 1, 1}}{n_{\lambda, 1, 0} + n_{\lambda, 1, 1}} \end{pmatrix}.$$

The mapping  $\mathbf{T}_Q$  is computable in time  $\mathcal{O}(|Q| + c)$  and space  $\mathcal{O}(|\Lambda|)$ , where  $c$  is the number of class boundaries. We can now formally define the context-aware Markov chain null hypothesis.

**Definition 2.** *The context-aware Markov chain null hypothesis ( $\mathcal{H}_0^{\text{context}}$ ) for query annotation  $Q$  and context  $\phi : \{0, \dots, L-1\} \rightarrow \Lambda$  is that the query annotation  $Q$  is generated by the context-aware Markov chain  $(\phi, \mathbf{T}_Q)$ .*

Note that under context  $\phi$  with a single class, the context-aware Markov chain null hypothesis  $\mathcal{H}_0^{\text{context}}$  reduces to the Markov chain null hypothesis by Gafurov et al. [5], with a small difference that the initial state distribution  $\vec{\pi}$  is set to the stationary distribution at position -1 instead of always starting in state 0 at position 0.

When there is just one context class,  $\mathcal{H}_0^{\text{context}}$  can be viewed as an approximation to the permutational null, i.e. shuffling the query intervals around in a random fashion [5]. In the case of multiple classes,  $\mathcal{H}_0^{\text{context}}$  can be thought of as an approximation to shuffling the query intervals around separately within each class. However,  $\mathcal{H}_0^{\text{context}}$  also transparently handles intervals spanning one or even multiple class boundaries.

### 2.3 Computing the Mean and Variance of the Overlap and Shared Bases Test Statistics

Here, we state our main algorithmic result: fast computation of the mean and variance of  $K(R, Q)$  and  $B(R, Q)$  statistics under  $\mathcal{H}_0^{\text{context}}$ . The mean and variance are then used to compute the p-values using the normal approximation.

**Theorem 1.** *Let  $R$  and  $Q$  be two annotations and let  $\phi$  be a genome context with  $c$  class boundaries. Let  $A$  be either the number of overlaps test statistic  $K$  or the number of shared bases test statistic  $B$ . It is possible to compute the mean  $\mathbb{E}_{Q' \sim C(\phi, \mathbf{T}_Q)}[A(R, Q')]$  and variance  $\text{Var}_{Q' \sim C(\phi, \mathbf{T}_Q)}[A(R, Q')]$  in space  $\mathcal{O}(|R|+c)$  and in time*

- $\mathcal{O}(|R| + |Q| + c)$  when  $A$  is the overlap test statistic,
- $\mathcal{O}(|Q| + (|R| + c) \log t)$  when  $A$  is the shared bases test statistic; here,  $t$  is the length of the longest stretch of positions within a single reference interval with the same context class in  $R$ .

Under the assumption that the test statistic is approximately normally distributed, this algorithm can be used to obtain the full probability mass function of its distribution under the null, i.e. values  $\Pr_{Q' \sim C(\phi, \mathbf{T}_Q)}[A(R, Q') = x]$  for all values of  $x$ . Note that the previous MCDP algorithm for this problem only works with a single context, only works for the overlap statistic  $K$ , and runs in  $\mathcal{O}(|R|^2 + |Q|)$  time [5]. However, it makes no assumption about normality.

In the next section, we provide a high-level description of the algorithm for computing expectation and variance. The details and proof of correctness were omitted due to space constraints.

### 2.4 Mean and Variance of Any Separable Statistic

We call a statistic *separable* if it can be expressed as a sum of contributions from each reference interval and each contribution depends only on the part of the query annotation inside this reference interval. For example, the contribution of each reference interval in the overlap statistic  $K$  is 1 or 0, depending on whether

there is an overlap with some query interval. In the shared bases statistic  $B$ , the contribution is the number of bases that a particular reference interval shares with the query intervals.

Thanks to linearity of expectation, the expected value of any separable statistic can be computed for every reference interval separately and then summed together. The simplest case is the shared bases statistic  $B$ , which can be expressed as the sum of indicator variables for each base covered by  $R$ , and in the case of single-class null model, the expectation can be computed simply as the number of bases covered by  $R$  multiplied by the stationary probability of state 1 of the Markov chain. Context-aware models complicate the situation, as each base of the genome has its unique marginal distribution over states, depending on the sequence of class labels preceding it.

Computing variance is more complicated, as the values of the statistic in individual intervals of  $R$  are dependent, and therefore the overall variance is not a simple sum of individual variances. However, in a sequence of Markov chain states  $(S_0, \dots, S_{L-1})$ , states  $S_i$  and  $S_j$  are conditionally independent given  $S_k = s$  for  $i \leq k \leq j$ . Therefore, if random variable  $X$  is a function of  $S_i, \dots, S_{k-1}$  and random variable  $Y$  is a function of  $S_k, \dots, S_{j-1}$ , then  $\text{Var}[X + Y \mid S_k = s] = \text{Var}[X \mid S_k = s] + \text{Var}[Y \mid S_k = s]$ .

Our algorithm computes conditional variance in individual intervals of  $R$  conditioning on states at both interval boundaries, and then combines them using this formula. In order to remove conditioning on the boundary states, we use the law of total variance, which for binary variable  $S_k$  can be written as  $\text{Var}(X) = \sum_{s=0}^1 \text{Var}[X \mid S_k = s] \Pr[S_k = s] + \sum_{s=0}^1 E[X \mid S_k = s]^2 \Pr[S_k = s](1 - \Pr[S_k = s]) - 2 \prod_{s=0}^1 E[X \mid S_k = s]^2 \Pr[S_k = s]$ .

The key data structure in our algorithm is a  $\mathcal{O}(1)$ -sized vector called a *two-sided plumbus* defined below. It contains the quantities that we need to compute for every interval of  $R$ , conditioning on states at interval boundaries. In the definition, function  $v$  expresses the contribution of a reference interval to the separable test statistic.

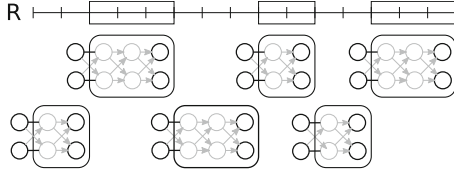
**Definition 3.** Let  $(\phi, \mathbf{T})$  be a context-aware Markov chain with state sequence  $S_{-1}, S_0, \dots, S_{|L|-1}$ , let  $[i, j)$  be a subinterval of  $[0, L)$ , and let  $v$  be a function on a binary sequence of length  $j - i$ . We define the two-sided plumbus for interval  $[i, j)$  as the collection of values

$$\begin{aligned} \mu_v(i, j \mid x, y) &:= \mathbb{E}[v(S_i, \dots, S_{j-1}) \mid S_{i-1} = x, S_{j-1} = y] \\ \sigma_v^2(i, j \mid x, y) &:= \text{Var}[v(S_i, \dots, S_{j-1}) \mid S_{i-1} = x, S_{j-1} = y] \\ \Psi(x \xrightarrow{i-1, j-1} y) &:= \Pr[S_{j-1} = y \mid S_{i-1} = x] \end{aligned}$$

for all combinations of  $(x, y)$  in  $\{0, 1\}^2$ .

The two-sided plumbuses computed for individual intervals of  $R$  and gaps between them are then combined to plumbuses for successively longer intervals, until we cover the whole chromosome and obtain the overall variance and expected value of the statistic of interest (see Fig. 2).





**Fig. 2.** Two-sided plumbuses for a reference annotation  $R$ . The plumbuses in the first row correspond to the reference intervals, and the plumbuses in the second row correspond to the gaps between the intervals. We highlight in black the boundary states on which we condition the values in each plumbus. Note that the conditional means  $\mu_v$  and variances  $\sigma_v^2$  in the gap plumbuses are constant zeroes, since gaps do not contribute to the total test statistic.

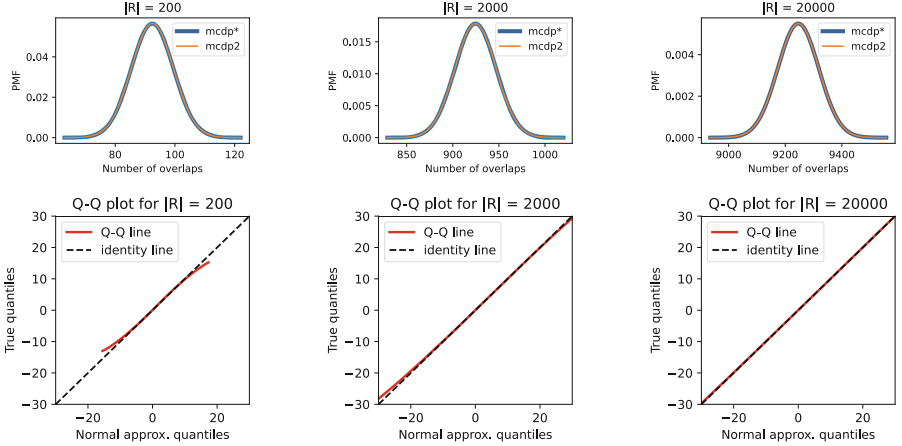
In the algorithm, we compute  $\Pr[S_j = y \mid S_i = x]$  and  $\Pr[S_i = S_{i+1} = \dots = S_j = 0]$  in constant time, provided that interval  $[i, j]$  is labeled by the same context class. This leads to a linear-time algorithm for  $K(R, Q)$  statistic. For  $B(R, Q)$  statistic, we split an interval of  $R$  into subintervals of size 1, compute plumbuses for them, and combine them in a similar manner, as we combine plumbuses in the overall algorithm. However, within a single context class, corresponding plumbus depends only on the interval length, and thus we can compute plumbuses for interval sizes which are powers of two and combine them to obtain a plumbus for any interval length within a single context in logarithmic time.

## 2.5 Multiple Chromosomes

Both our model and our algorithm can be extended to genomes with multiple chromosomes in a straightforward way. We assume that the query annotation is generated independently for each chromosome. The training of the context-aware Markov chain is accomplished simply by counting transition frequencies on all chromosomes. The test statistic for the whole genome is defined as the sum of test statistic values for the individual chromosomes. This, in turn, allows to compute the mean and variance of the total statistic by summing the means and variances, respectively, for the individual chromosomes. Note that this simple computation works for the variance thanks to the chromosome independence assumption. Therefore, the time and space complexity of our algorithm remains the same for the case of multiple chromosomes.

## 3 Experiments

*The Normal Distribution Yields an Accurate  $p$ -value Approximation.* Our MCDP2 algorithm computes the exact expectation and variance of the null distribution and uses them to approximate the null distribution by the normal distribution. Here, we first compare the accuracy of this approximation for the  $K(R, Q)$  statistic with the exact distribution computed by the previous MCDP algorithm [5]. The comparison was performed on synthetic data sets with genome



**Fig. 3.** The comparison of the exact probability mass functions (PMFs) for the number of overlaps  $K$  statistic (MCDP\*) with its normal approximation (MCDP2) on synthetic data sets. Each column represents a different number of reference intervals ( $|R| \in \{200, 2000, 20000\}$ ). The bottom plots show the quantile-quantile plot (Q-Q plot) between the two distributions.

length  $L = 10^8$  bp, query annotations with 50 000 randomly generated intervals of length 500 bp each and reference annotations with up to 20 000 intervals of length 500 bp each. To understand the influence of the number of reference intervals on the accuracy, we vary  $|R|$  from 200 to 20 000.

Figure 3 shows that the exact PMF in general agrees well with the normal approximation. The approximation approach allows to estimate even very low p-values accurately with the growing number of reference intervals. However, for  $|R| = 200$  the differences in the extreme tail of the distribution lead to overly conservative p-values. Therefore, for small values of  $|R|$  we recommend the use of the exact MCDP algorithm, which is not time-prohibitive. The new MCDP2 tool includes a reimplement of the exact computation of the PMF and its extension to multiple context classes, and we use it in these experiments under the label MCDP\*. We have also conducted similar experiments for the shared bases statistic  $B(R, Q)$  with similar results (results not shown due to space constraints).

*MCDP2 is Fast and Memory Efficient.* The speed of our algorithm enables us to apply our tools to large-scale comparisons, such as the data from a recent study of ENCODE epigenetic modification enrichment for different repeat types in the human genome [8], employing the Telomere-to-Telomere (T2T) human genome assembly [14]. We use a context with two classes, one corresponding to all repeats and one to the rest of the genome, leading to over 4 million class boundaries. We use one of the 10 repeat types as the reference and one of the 45 available combinations of an epigenetic modification and a cell line as the

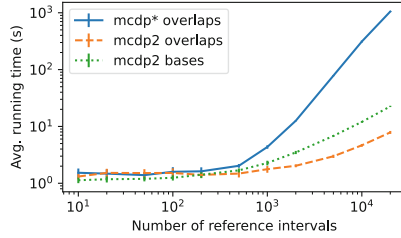
**Table 1.** Data set sizes and average running times for comparison of repeat types ( $R$ ) with epigenetic modifications ( $Q$ ), using all repeats as context. Averages are computed across 45 different query annotations, each representing a specific epigenetic modification in a specific cell line. Note that the running time grows with  $|R| + c$ , and in this experiment,  $c$  is large even for inputs with small  $|R|$ .

Repeat type	$ R $	time (s)	
		$K(R, Q)$	$B(R, Q)$
RNA	11 139	220	160
Other	8 835	220	157
Unknown	11 229	226	159
Satellite	47 041	229	202
Low-complexity	102 521	244	223
DNA	505 896	343	588
LTR	660 823	387	769
Simple	708 565	398	637
LINE	1 440 792	578	1 477
SINE	1 672 984	640	1 637

query. Using 24 CPU threads, MCDP2 computed p-values for all 450 pairs in approx. 2 hours (wall clock) for the number of overlaps and approx. 3 hours for the number of shared bases, using at most 4.2 Gb (2.3 Gb) memory per comparison for overlaps (shared bases) statistic (see Table 1).

We compare the running time of MCDP2 to the quadratic-time MCDP\* algorithm on the synthetic data sets used for Fig. 3, with 20 pairs of  $R$  and  $Q$  generated for each setting (Fig. 4). Note that MCDP\* is faster than the original MCDP implementation [5], thanks to more extensive use of `numpy` library and reimplementing part of the algorithm in C++. For the overlaps statistic, the MCDP\* needs more than 1 000 seconds for 20 000 reference intervals, while our new approach MCDP2 only takes approximately 8 seconds on the same inputs. Computation for the number of shared bases is slightly slower (23 seconds for  $|R| = 20\,000$ ), which is consistent with its quasi-linear time complexity (in contrast to purely linear for the number of overlaps).

*Genome Contexts Enable More Detailed Analysis of Colocalization of Copy Number Loss with Different Gene Groups.* To illustrate the power of our context-aware null model, we have reanalyzed the colocalization of exons of different gene groups with copy number loss regions, originally performed by Zarrei et al. [21]. Figure 5 shows the Z-scores for both  $K$  and  $B$  test statistics and for three types of contexts. The first context function only uses a single class. The second context function creates two classes by masking regions that are assembly gaps; this is motivated by the fact that both copy number losses and exons are annotated exclusively outside the gaps and, therefore, may appear colocalized



**Fig. 4.** Average running time on synthetic data for overlaps (MCDP2, MCDP\*) statistic and shared bases (MCDP2) statistic. The vertical bars represent the standard deviation over 20 samples. Both axes are in log scale. The calculations were performed on a single thread on Intel(R) Xeon(R) Gold 6248R CPU.

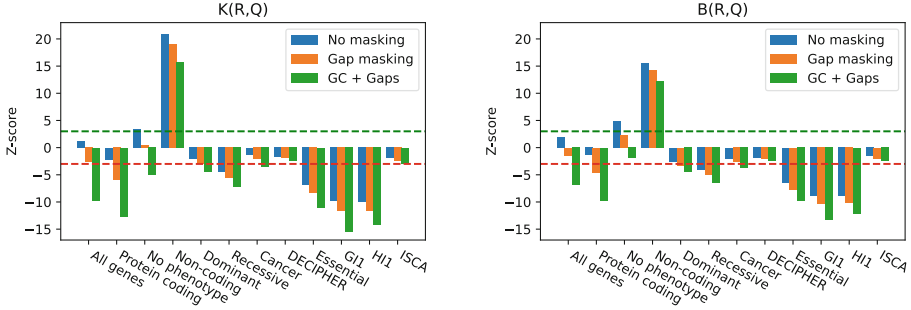
even if they were independent (previously also studied by Domanska et al. [2]). The third context function uses six classes: one for gaps and the other five for discretization of the GC content in 1 kbp windows; this is motivated by the fact that GC content is known to be a significant confounding factor in many genomic analyses [1, 7].

Figure 5 illustrates the importance of having a class in the context dedicated to gaps. In one jarring scenario, the set of all exons is enriched for overlaps with copy number losses (also observed by Zarrei et al.), but after accounting for gaps, the exons become depleted. More generally, across all studied gene groups, the Z-score decreases when the gaps are taken into account. This is expected as neither exons nor copy number losses occur in gaps, and thus ignoring gaps in the analysis may create spurious enrichments or lower the degree of observed depletion compared with analysis that takes gaps into account.

The GC-aware context also proves crucial for an accurate analysis. For example, the depletion of all exons for overlaps with copy number losses becomes much more pronounced in the GC-aware context. In another striking example, genes with no known phenotype appear enriched for overlap with losses (also in agreement with Zarrei et al.) when using the gap-aware context, but enrichment turns into slight depletion after taking GC content into account.

Other observations from Fig. 5 are generally consistent with biological expectations. Protein-coding genes are only slightly depleted in the single-class context but become significantly depleted when using gap-aware or GC-aware contexts. This depletion is consistent with the expectation that protein-coding exons are mostly evolutionarily conserved. Interestingly, the set of non-coding genes is strongly enriched for copy number losses under all three context functions, and the enrichment was also observed by Zarrei et al.

*Differential Analysis of Non-telomeric and Telomeric TAR Elements.* Completion of the previously inaccessible parts of the human genome [14] has allowed Gershman et al. [8] to study telomere-associated repeats (TARs) and their colocalization with epigenetic modifications. While TARs located in sub-telomeric regions are presumed to be important for telomere length regulation,

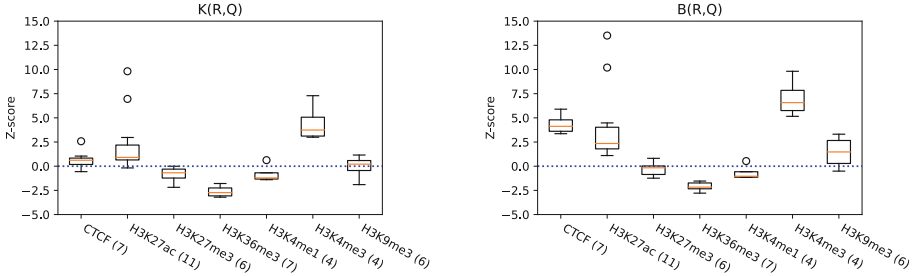


**Fig. 5.** Z-scores for colocalization of exons of various gene groups ( $R$ , x-axis) with copy number losses ( $Q$ ) under three different null models: single-class context, gap-aware context, and GC-aware context. Left: overlap statistic  $K$ ; Right: shared bases statistic  $B$ . The green and red dashed lines stand for Z-score  $+3$  and  $-3$  respectively, corresponding to p-value 0.00135 for enrichment/depletion. (Color figure online)

TAR copies have also been dispersed to other parts of the genome. Differences between these two groups may further clarify mechanisms and functions of TARs in subtelomeric regions. While Gershman et al. observe differences in enrichment of some epigenetic marks, they do not assign statistical significance to their findings.

We adapted our context-aware Markov chain model to perform such differential analysis of enrichment between two annotations. In general, consider two references  $R_1 \subseteq R_2$ . In our case,  $R_1$  are non-telomeric TARs,  $R_2$  are all TARs, and  $Q$  are regions with a particular epigenetic mark. One could compare the enrichment p-value of  $Q$  in  $R_1$  with the enrichment p-value of  $Q$  in  $R_2$ ; however, this is not statistically sound [9, 20]. Instead, we create a context  $\phi_{\text{rel}}$  with two class labels {outside, inside}, where positions covered by  $R_2$  are labeled “inside” and all other positions are labeled “outside.” We then use a test statistic to measure the significance of enrichment of  $Q$  in reference  $R_1$  with context  $\phi_{\text{rel}}$ . This context ensures that within  $R_1$  the null model uses the parameters estimated from intervals of  $Q$  that overlap  $R_2$ , thus comparing colocalization of  $Q$  in  $R_1$  relative to colocalization of  $Q$  in the whole  $R_2$ . Note that the query intervals can occur also outside of  $R_2$ , and their properties are summarized in the parameters of the Markov chain for the “outside” class. These outside areas then influence the distribution of the test statistic under the null only by influencing the initial state distribution at the start of each interval of  $R_1$ .

Figure 6 shows the result of this analysis. Similarly to Gershman et al. [8] we observe relative enrichment of activating marks  $H3K27ac$  and  $H3K4me3$  in non-telomeric TARs using both  $K$  and  $B$  statistics. We can also see enrichment of  $CTCF$ , which is significant only under the shared bases statistic, perhaps due to the small number of intervals in  $R_1$ . Gershman et al. were not able to observe relative enrichment for  $CTCF$  on non-telomeric TARs, although they do observe that  $CTCF$  is strongly enriched in both TAR classes compared to



**Fig. 6.** Relative enrichment of telomere-associated repeats (TARs) located further than 20 kbp from chromosome ends ( $R_1$ ) with epigenetic modifications ( $Q$ ) in comparison to all TARs ( $R_2$ ), using both the number of overlaps statistic  $K$  (left) and the number of shared bases statistic  $B$  (right). The numbers in the parentheses denote the number of cell lines available for each modification.

the background. This highlights usefulness of our context model in scenarios requiring relative analysis of two reference annotations.

## 4 Discussion

We have introduced a novel model for annotation colocalization analysis, which uses genomic contexts to capture confounding factors that may lead to false significance results. Taking advantage of the Markovian properties of our model, we have provided a general framework to compute the exact mean and variance of a broad class of colocalization test statistics (which we named *separable*). Using this framework, we were able to obtain linear and quasi-linear algorithms to compute the Z-scores for the number of overlaps and the number of shared bases. We have then proposed to convert the exact Z-score to approximate p-values using the normal distribution.

Our algorithm computes the Z-score in  $\mathcal{O}(|Q| + |R| + c)$  time for the overlap number statistic and in  $\mathcal{O}(|Q| + (|R| + c) \log t)$  time for the shared bases statistic, where  $|Q|$  and  $|R|$  are the number of intervals in the query and reference, respectively,  $c$  is the number of context class switches, and  $t$  is an upper bound on the reference interval length. This is in contrast to the previous best algorithm, which did not account for genome contexts and took  $\mathcal{O}(|R|^2 + |Q|)$  time to compute the probability mass function of the p-values [5].

In our experiments, we have demonstrated that our algorithm is sufficiently fast to allow large-scale studies comparing many pairs of annotations with large reference sets and frequent context class boundaries. We have reanalyzed data sets from two large-scale studies [8, 21], and thanks to our new context-aware model, we were able to further illuminate the nature of colocalizations discovered in these works, in some cases reversing previously published findings.

We have experimentally shown that the normal approximation of the distribution of the number of overlaps under  $\mathcal{H}_0^{\text{context}}$  yields accurate p-values, and

the approximation gets tighter with increasing number of reference annotation intervals. This behaviour intuitively follows from the representation of a separable statistic as a sum of contributions for individual reference intervals. If those contributions were independent, their sum would converge to a normal distribution with a growing number of reference intervals under the classical central limit theorem. Though the contributions of individual intervals are dependent in our case, the fact that the dependencies stem from a Markov chain makes it possible that the sum converges under some extensions of the central limit theorem. In future, we hope to characterize sufficient conditions for such convergence. Additionally, we would like to explore the possibility of providing lower and upper bounds on the precision of the p-value estimation, possibly by applying the Stein's method [16].

On a more practical side, in our future research, we would like to explore the possibility of using quantitative contexts, with numeric values such as GC content, epigenetic mark density, sequence conservation etc. Some work in this direction has already been done, particularly in HyperBrowser [17]. In MCDP2 this could be achieved for example by parameterizing the weights of the underlying Markov chains with the context value at each position. The challenge would be to keep the running time efficient for large genomes.

Another challenge is to provide statistical significance for statistics comparing colocalization of query  $Q$  with respect to two different references  $R_1$  and  $R_2$ , such as  $B(R_1, Q)/B(R_2, Q)$ . This may in some situations be preferable to our approach of comparing such colocalization through contexts, which we used for the analysis of TAR elements.

**Funding Information.** This material is based upon work supported by the National Science Foundation under Grant No. DBI-2138585. Research reported in this publication was supported by the National Institute Of General Medical Sciences of the National Institutes of Health under Award Number R01GM146462. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. This work was also supported by a grant from the European Union Horizon 2020 research and innovation program No. 872539 (PANGAIA); and grants from the Slovak Research and Development Agency APVV-22-0144 and the Scientific Grant Agency VEGA 1/0538/22.

## References

1. Burge, C., Karlin, S.: Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.* **268**(1), 78–94 (1997)
2. Domanska, D., Kanduri, C., Simovski, B., Sandve, G.K.: Mind the gaps: over-looking inaccessible regions confounds statistical testing in genome analysis. *BMC Bioinf.* **19**(1), 481 (2018). <https://doi.org/10.1186/s12859-018-2438-1>
3. Dozmorov, M.G., Cara, L.R., Giles, C.B., Wren, J.D.: GenomeRunner web server: regulatory similarity and differences define the functional impact of SNP sets. *Bioinformatics* **32**(15), 2256–2263 (2016)

4. Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G.: *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge (1998). <https://doi.org/10.1017/CBO9780511790492>
5. Gafurov, A., Brejová, B., Medvedev, P.: Markov chains improve the significance computation of overlapping genome annotations. *Bioinformatics* **38**(Supplement-1), i203–i211 (2022). <https://doi.org/10.1093/bioinformatics/btac255>
6. Gel, B., Diez-Villanueva, A., Serra, E., Buschbeck, M., Peinado, M.A., Malinverni, R.: regioneR: an R/Bioconductor package for the association analysis of genomic regions based on permutation tests. *Bioinformatics* **32**(2), 289–291 (2016)
7. Gelfman, S., Ast, G.: When epigenetics meets alternative splicing: the roles of DNA methylation and GC architecture. *Epigenomics* **5**(4), 351–353 (2013)
8. Gershman, A., et al.: Epigenetic patterns in a complete human genome. *Science* **376**(6588) (2022). <https://doi.org/10.1126/science.abj5089>
9. Goodman, S.: A dirty dozen: twelve p-value misconceptions. *Semin. Hematol.* **45**(3), 135–140 (2008). <https://doi.org/10.1053/j.seminhematol.2008.04.003>
10. Guenther, M.G., Levine, S.S., Boyer, L.A., Jaenisch, R., Young, R.A.: A chromatin landmark and transcription initiation at most promoters in human cells. *Cell* **130**(1), 77–88 (2007). <https://doi.org/10.1016/j.cell.2007.05.042>
11. Heger, A., Webber, C., Goodson, M., Ponting, C.P., Lunter, G.: GAT: a simulation framework for testing the association of genomic intervals. *Bioinformatics* **29**(16), 2046–2048 (2013). <https://doi.org/10.1093/bioinformatics/btt343>
12. Kanduri, C., Bock, C., Gundersen, S., Hovig, E., Sandve, G.K.: Colocalization analyses of genomic elements: approaches, recommendations and challenges. *Bioinformatics* **35**(9), 1615–1624 (2019). <https://doi.org/10.1093/bioinformatics/bty835>
13. Koller, D., Friedman, N.: *Probabilistic Graphical Models: Principles and Techniques*. MIT press, Cambridge (2009)
14. Nurk, S., et al.: The complete sequence of a human genome. *Science* **376**(6588), 44–53 (2022). <https://doi.org/10.1126/science.abj6987>
15. Quinlan, A.R., Hall, I.M.: BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* **26**(6), 841–842 (2010). <https://doi.org/10.1093/bioinformatics/btq033>
16. Ross, N.: Fundamentals of Stein’s method. *Probab. Surv.* **8**(1), 210–293 (2011). <https://doi.org/10.1214/11-PS182>
17. Sandve, G.K., et al.: The genomic hyperbrowser: inferential genomics at the sequence level. *Genome Biol.* **11**(12) (2010). <https://doi.org/10.1186/gb-2010-11-12-r121>
18. Sarmashghi, S., Bafna, V.: Computing the statistical significance of overlap between genome annotations with ISTAT. *Cell Syst.* **8**(6), 523–529 (2019). <https://doi.org/10.1016/j.cels.2019.05.006>
19. Sheffield, N.C., Bock, C.: LOLA: enrichment analysis for genomic region sets and regulatory elements in R and Bioconductor. *Bioinformatics* **32**(4), 587–589 (2016). <https://doi.org/10.1093/bioinformatics/btv612>
20. Sullivan, G.M., Feinn, R.: Using effect size-or why the P value is not enough. *J. Grad. Med. Educ.* **4**(3), 279–282 (2012)
21. Zarrei, M., MacDonald, J.R., Merico, D., Scherer, S.W.: A copy number variation map of the human genome. *Nat. Rev. Genet.* **16**(3), 172–183 (2015). <https://doi.org/10.1038/nrg3871>