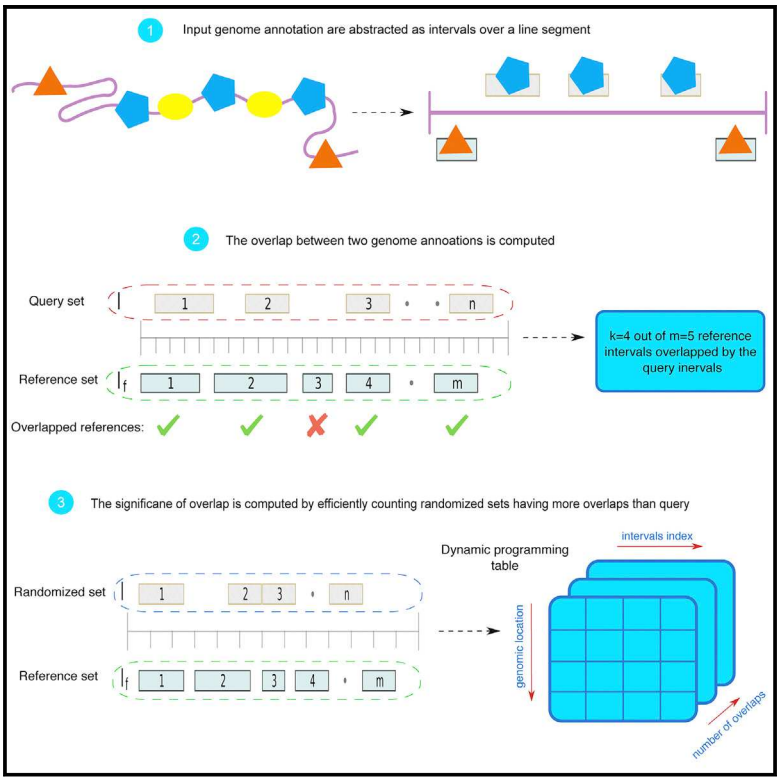
Cell Systems

Computing the Statistical Significance of Overlap between Genome Annotations with ISTAT

Graphical Abstract



Highlights

- The overlap between genome annotations is often used to study biological association
- We describe a tool for computing the statistical significance of annotations' overlap
- Our method corrects p values reported in previous experiments by orders of magnitude

Authors

Shahab Sarmashghi, Vineet Bafna

Correspondence

vbafna@cs.ucsd.edu

In Brief

Annotating the genome by demarcating coordinates of gene structures, sequences associated with methylation marks, etc., is a fundamental problem in biology. Annotated regions can be abstracted as intervals on a line, and the overlap between sets of intervals is often used to establish correlation between annotations and obtain biological insights. Computing the statistical significance of overlap between annotations is a relatively unexplored problem, often done using permutation tests and assumptions on the null distribution. We describe a tool for efficiently computing the significance of the overlap between two sets of intervals using a dynamic programming approach. The tool corrects the p values reported in previous experiments by orders of magnitude.

Sarmashghi & Bafna, 2019, Cell Systems *8*, 523–529
June 26, 2019
https://doi.org/10.1016/j.cels.2019.05.006

CellPress

Cell Systems
Focus on RECOMB
Report

Computing the Statistical Significance of Overlap between Genome Annotations with iSTAT

Shahab Sarmashghi[1] and Vineet Bafna[2,3,*]
[1]Department of Electrical and Computer Engineering, University of California, San Diego, San Diego, La Jolla, CA 92093, USA
[2]Department of Computer Science & Engineering, University of California, San Diego, San Diego, La Jolla, CA 92093, USA
[3]Lead Contact
*Correspondence: vbafna@cs.ucsd.edu
https://doi.org/10.1016/j.cels.2019.05.006

SUMMARY

Genome annotation remains a fundamental effort in modern biology. With reducing costs and new forms of sequencing technologies, annotations specific to tissue type and experimental conditions are continually being generated (e.g., histone methylation marks). Computing the statistical significance of overlap between two different annotations is key to many biological findings but has not been systematically addressed previously. We formalize the problem as follows: let I and I_f each describe a collection of n and m intervals of a genome with particular annotation. Under the null hypothesis that genomic intervals in I are randomly arranged with respect to I_f, what is the significance of k of m intervals of I_f intersecting with intervals in I? We describe a tool iSTAT that implements a combinatorial algorithm to accurately compute p values. We applied iSTAT to simulated and real datasets to obtain precise estimates and contrasted them against previous results using permutation or parametric tests.

INTRODUCTION

Annotating the genome is a central problem in biology. Subsequent to the sequencing and assembly of the human genome and the development of deep sequencing technologies, researchers have developed a number of technologies aimed at identifying functional regions on the genome. Examples of annotation include repeat elements (Jurka, 2000), protein-coding genes (Venter et al., 2001), non-coding RNA (Bartel, 2009), regulatory regions (ENCODE Project Consortium, 2012), sites with specific epigenetic modifications (ENCODE Project Consortium, 2007), transcription start sites (ENCODE Project Consortium, 2012), ribosome initiation sites (Ingolia et al., 2009, 2012), and regions relating to genome structure, such as the regions with a change in copy number and other variation (Pinkel et al., 1998; Feuk et al., 2006). With reducing costs and new forms of sequencing technologies, annotations specific to tissue type and experimental conditions are continually being generated.

In all of these examples, we implicitly represent the genome as a line segment and an "annotation" as a collection of non-overlapping intervals on that line. Excessive overlap of the intervals in a pair of annotations is indicative of a biological association and is widely used to support hypotheses asserting biological principles. While studying the function of epigenetic modifications on the genome, Guenther et al. (2007) observed that about 3/4 of all known promoter regions overlapped with intervals highly enriched for the methylation of lysine 4 on histone H3 (H3K4me3) including in genes without any detected transcript. Assuming that the presence of histone H3K4me3 was correlated with transcription initiation, they hypothesized that transcription initiation occurs in all genes but was followed by transcriptional elongation only in active genes. In another example, Zarrei et al. (2015) computed the association of the copy number variable (CNV) genomic regions against each of the multiple annotations such as protein-coding and non-coding genes, cancer genes, lincRNAs, promoters, etc., to assess the variability of different functional regions of the genome. Wu et al. (2003) studied viral integration in the human genome finding that a large fraction of HIV-1 and MLV integrations in H9 and HeLa cells lay within the start and end of transcription of a gene. In contrast, while 16.8% of the MLV integrations landed ±1Kb from a CpG island, only 2.1% of HIV-1 integrations landed near a CpG island. They computed p values using permutation tests to assert the significance of these differential associations.

In experiments related to genome annotations, such questions are ubiquitous, and they all distill down to the underlying statistical question of significantly overlapping intervals. Hence, it has been a standard practice to compute a p value using the null distribution of overlaps against randomly located intervals. Random annotations can be generated by randomizing the position of intervals while preserving the coherence of each region and provide exact answers when the space of all possible random samples can be enumerated. However, in many real-life examples including the above studies, the sample space is enormous, and naive sampling-based methods cannot achieve adequate resolution to distinguish between rare events in feasible running times. On the other hand, while parametric tests used in the literature are computationally efficient, they oversimplify the problem by casting intervals as points and ignore the dimension of annotated regions on the genome, which often results in artificially low p values, thereby inflating apparent significance.

In this paper, we introduce a tool, iSTAT, which can enumerate over the space of all randomized samples in order to find the exact null distribution, under the assumption that the order of intervals is preserved when randomizing their position (see Box 1).

Box 1. Primer

Genome annotation, referring to the assignment of function to specific regions, remains a foundational effort of modern biology. With reducing costs and new forms of sequencing technologies, annotations specific to tissue type and experimental conditions are continually being generated (e.g., histone methylation marks, regions with high gene expression, and genomic copy number,). Furthermore, excessive overlap of the intervals in a pair of annotations is indicative of a biological association and is widely used as a basis for new biological insight, including examples such as the overlap of histone methylation sites and promoter activity, targeted insertion of viral sequences into the human genome, and others. Computing the statistical significance of the overlap between two different annotations is key to these experiments. However, the problem has not been systematically addressed previously. To the best of our knowledge, the p value computation for sets of overlapping intervals has been limited either to permutation tests that do not scale to computation of small p values or simple parametric tests such as hypergeometric or binomial tests that are based on simplifying assumptions about the length and structure of intervals. Our paper, however, formulates a null model where the size of intervals and their relative arrangement are considered when the significance of overlap is evaluated.

We formalize the problem as follows: let I and I_f each describe a collection of n and m intervals on a line segment of finite length. Under the null hypothesis that intervals in I are randomly arranged w.r.t I_f, what is the significance of k of the m intervals of I_f intersecting with some interval in I? We describe a tool ISTAT that implements a combinatorial algorithm to accurately compute p values and also describe trade-offs that make the computation fast without losing accuracy. ISTAT first computes k, the number of intervals in I_f that have overlap with any interval in I. The significance of the overlap between reference and query intervals is measured by sampling a random set of intervals, I_r, where the positions of query intervals are randomized along the genomic region while retaining the total number of intervals and their individual lengths same as the intervals in I. To reduce the combinatorial complexity of the problem, we also assume that the order of intervals in I are preserved when sampling for I_r, but we show through our simulations that the impact of this additional assumption is negligible for a typical enrichment problem. The p value is computed by counting the fraction of times when k or more overlaps occur between I_f and I_r. We applied ISTAT to simulated and real datasets to obtain precise estimates. In many cases, the ISTAT estimates provided a significant correction to previous results obtained using permutation tests or parametric tests.

Using simulated data, we show that the impact of our assumption on p value calculation is limited. ISTAT also provides a fast approximate solution based on Poisson binomial (PB) distribution, and using simulated data, we characterize its performance in approximating the generic null distribution. Moreover, we demonstrate the result of applying our methods to four examples of interval overlap problem from previously published studies and compare ISTAT results with the p values reported in those studies.

RESULTS

We used the following notation throughout the paper. Let I_f denote a "reference" collection of m intervals and I denote a "query" collection of n intervals (Figure 1). Each interval is denoted by a pair of indices (u_1, u_2) with $0 \leq u_1 < u_2 \leq g$, where g denotes the length of the genomic region of interest, for example, a chromosome. ISTAT first computes k, the number of intervals in I_f, which have overlap with any interval in I. The significance (p value) of the overlap between reference and query intervals is measured by sampling a random set of intervals, I_r, where the positions of query intervals are randomized along the genomic region while retaining the total number of intervals and their individual lengths same as the intervals in I (Figure 1). To reduce the combinatorial complexity of the problem, we also assume that the order of intervals in I are preserved when sampling for I_r, but we show through our simulations that the impact of this additional assumption is negligible for a typical enrichment problem. The p value is naturally defined as the probability of observing k or more overlaps between I_f and I_r.

It is not feasible to count random sets one by one as the space of all possible random intervals expands exponentially with the number of intervals. Instead, ISTAT uses a dynamic programming (DP) algorithm. For each $0 \leq k \leq m$, we recursively compute the number of all distinct random sets resulting in k overlaps and calculate the p value from the cumulative distribution of the overlap statistics (STAR Methods). In practice, to make the computation efficient for large genomes with large numbers of intervals, we use a practical interval "scaling" option by considering the natural partitioning of the genome into intervals and the gaps amid them and scale each interval and gap in I and I_f by a fraction ν. The running time of ISTAT p value computation is $\mathcal{O}(ng\nu m)$, and the memory usage scales as $\mathcal{O}(g\nu m)$, where both can be controlled by choosing a proper scaling factor $\nu < 1$.

In ISTAT, we provide an even more efficient option to approximate the p value. Specifically, assuming that intervals in I_f are overlapped (by I_r intervals) independently from each other, we can show that the overlap statistics k follows a PB distribution (STAR Methods). We characterize the impact of independence assumption on the accuracy of computed p values by defining a parameter η as a measure of "spread" of intervals in I_f (STAR Methods) and investigating the approximation for different values of η. We provide empirical bounds on η to guide the user on how closely PB approximates the distribution of overlap statistics, especially when annotations include a large number of intervals.

Performance of Simulated Data

We simulated intervals in a randomly generated chromosome to test the performance of ISTAT. To study the impact of scaling and fixed-order assumption on the DP algorithm, we chose g = 200Mbp, and the two sets of intervals I and I_f with $n = m =$ 100 intervals. The intervals in I and I_f were generated with random lengths l_i and x_j distributed uniformly over [1Kbp, 10Kbp]. The

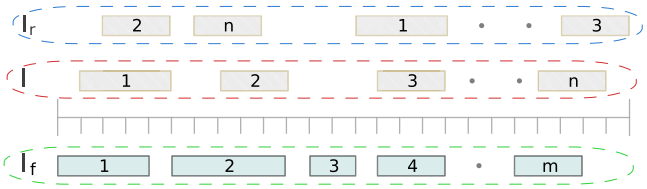


Figure 1. A Schematic of Interval Overlap Problem

I_f denotes the reference collection of intervals and I represents the query collection. The randomized set I_r is generated by relocating the intervals in I such that all possible non-overlapping random sets are equiprobable.

intervals in I_f were placed uniformly at random along the chromosome, while ensuring no overlap between them. We benchmarked iStat speed across a wide range of values for n, m, and g. We also simulated intervals in I_f distributed non-uniformly to study how their positional distribution impacted the quality of PB approximation.

The Impact of Scaling on p Value

The iStat algorithm has substantial demands on memory and time. To allow it to work on the human genome, we scaled down the intervals and the gaps between them by a fraction ν. To test the impact of scaling, we considered the example of a chromosome described above, with g = 200Mbp and $n = m$ = 100. The impact on DP p values due to scaling with $\nu \in \{1, 10^{-1}, 10^{-2}, 10^{-3}\}$ is shown in Figure 2A. As can be observed, scaling preserves the p values tightly. To further investigate robustness of DP p value computation to the scaling, we also considered an adversarial example where I and I_f contain intervals smaller than ν^{-1}. For that purpose, the interval lengths were selected from a uniform distribution over [100bp, 4Kbp]. Thus, when we applied scaling factor $\nu = 10^{-3}$, approximately one-fourth of intervals were smaller than the resolution ν^{-1} and become unit intervals. Nevertheless, p values obtained with $\nu = 10^{-3}$ tightly followed finer-scale p values (Figure 2B), validating the use of scaling to make the computation efficient.

Effect of Order on p Value

To test the effect of fixed order on p value, we used a scaling factor $\nu = 10^{-3}$ and applied the iStat DP method to 100 random instances of simulated intervals described before, each with a random permutation of I. In Figure 2C, we plotted the mean p value for all k, as well as the standard error of the mean. We observe that the standard error was distributed tightly around the mean (at least an order of magnitude smaller than the mean for all k), while its ratio to the mean increased slightly for smaller p values. The mean p value range from $0.4320 \pm 2.279 \cdot 10^{-4}$ for $k = 1$ to $1.017 \cdot 10^{-269} \pm 6.246 \cdot 10^{-271}$ for $k = 100$. The results suggest that fixing the order in DP algorithm to compute the p value is an acceptable compromise for many real datasets.

Running Time

Using a desktop PC with Intel Core i7-6700K CPU and 32 GB DDR4 RAM, the running time of our DP algorithm (in a logarithmic scale) versus the number of query intervals is plotted in Figure 2D for a number of scaling factors. The running time scales almost linearly with the number of query intervals n. It also scales linearly with the number of reference intervals m (Figure 2E), the size of chromosome g (Figure 2F), and when larger scaling factors are used.

Poisson Binomial Approximation

To study the accuracy of using PB for the distribution of overlap statistics, we simulated different cases by changing the number of query and reference intervals as well as the spread of reference intervals over the genome. Although the closeness of PB approximation is a complicated function of the distribution of intervals and its exact characterization is hard, the parameter η, defined as the ratio of spread of reference intervals to the total length of genomic region (STAR Methods), proved to be relevant, yet simple to calculate. To test the role of η in p value estimation, we compared the p values of the PB method against the DP method for different values of η (Figure 3). Relative to the DP, the PB approximation underestimates p values when $\eta = 0.005445$ (Figure 3A) and overestimates for $\eta = 0.6197$ (Figure 3C). However, this over-estimation is not as pronounced as the underestimation in the case of clumping and reduces with large n (Figures 3E and 3F). As a rule of thumb, we suggest using DP (with the largest computationally feasible scaling ν) when $\eta < 0.06$, to avoid inflating the significance of overlap. For the case of multiple chromosomes, the minimum η among all chromosomes is recommended as a conservative choice.

Enrichment Analysis on Real Data

To test our methods on interval data from previously published studies, we applied iStat to four examples from the literature and compared the results with the reported p values. The first example comes from Deshpande et al. (2018), relating to matching of focal copy number changes in tumor genomes. The second dataset is from Zarrei et al. (2015), where a map of CNV in the human genome is provided, and different genomic elements are investigated for the presence or absence of CNVs. We also ran iStat on an example from an epigenetics context (Guenther et al., 2007), where the promoters are found to be enriched for H3K4 methylation. The last example was extracted from an effort to systematically annotate the genome by the means of characterizing chromatin states (Ernst and Kellis, 2010).

TCGA-CNV Enrichment in Extra-chromosomal DNA

Focal copy number amplification (CNA) is central to the pathology of many cancers (Davoli et al., 2017; Verhaak et al., 2019), but its mechanistic origin is not well understood. Recent results suggest that CNA can often be attributed to formation of independently replicating extra-chromosomal DNA (ecDNA) elements (Turner et al., 2017). This could be tested by measuring the significance of the association of ecDNA regions obtained from tumor-derived cell lines (I) against CNAs identified from array-CGH data (denoted as I_f) from tumor genomes (TCGA: The Cancer Genome Atlas Program - National Cancer Institute, n.d.).

The number of intervals in query and reference sets were not large, with $n = 116$ and $m = 101$, so we did not scale the intervals, obtaining the p value $8.679 \cdot 10^{-6}$ at the observed overlap $\mathcal{K} = 54$. For comparison, we got the same p value after scaling with $\nu = 10^{-1}$. As expected from $\eta = 0.001$, the PB approximation ($p - \text{value} = 2.642 \cdot 10^{-10}$) inflated the significance of the

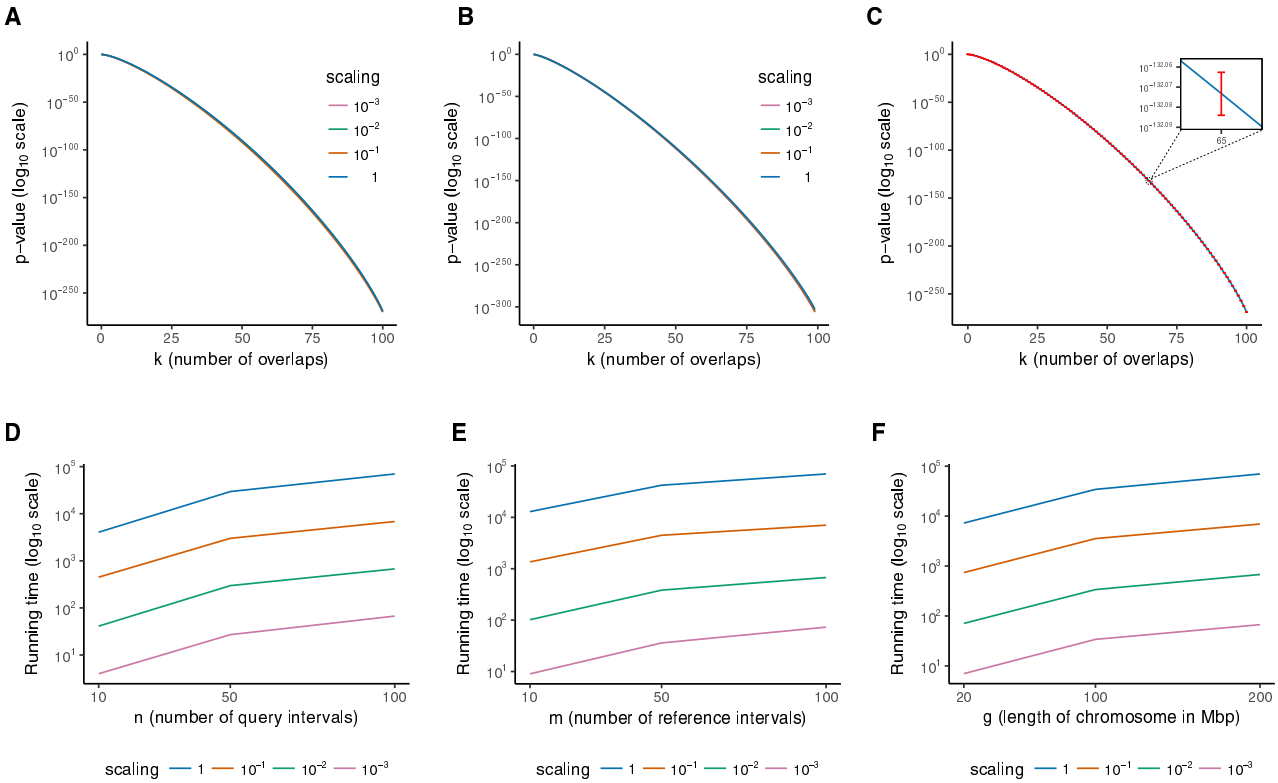


Figure 2. Testing DP Algorithm on Simulated Data

Impact of scaling parameter ν on the DP p value when (A) $l_i, x_j \sim \mathcal{U}[1\text{Kbp}, 10\text{Kbp}]$, and (B) $l_i, x_j \sim \mathcal{U}[100\text{bp}, 4\text{Kbp}]$. (C) Impact of ordering on the DP p value, with $\nu = 10^{-3}$. The mean of 100 p value computations for random orderings is plotted, and the error bars represent the standard error of the mean. Running time (in s) of the DP algorithm: (D) as a function of n, with $m = 100$ and $g = 200$Mbp. (E) as a function of m, with $n = 100$, $g = 200$Mbp. (F) as a function of g, with $n = m = 100$.

association (Figure 4A). Overall, the ISTAT DP results were useful in validating proposed mechanisms for the origin of focal CNA in cancer.

Non-coding Genes Enrichment in CNVs

Zarrei et al. (2015) tested the overlap between $n = 3132$ regions of copy number gains (I) against the location of $m = 9058$ non-coding genes (I_f). Using a permutation test, they reported a p value of 0.0001, showing the limited resolution of permutation tests. In the supplementary data, they used a binomial distribution to report another estimate of $p-\text{value} = 2.32 \cdot 10^{-54}$, pointing to the difficulty of getting an accurate estimate.

Using the scaling factor $\nu = 10^{-2}$, with $\mathcal{K} = 987$ of intervals in I_f overlapped, we computed $p-\text{value} = 5.216 \cdot 10^{-18}$, confirming high enrichment of non-coding genes in CNV gains. After applying an order of magnitude smaller scaling factor $\nu = 10^{-3}$, we obtained a very similar estimate of $p-\text{value} = 2.532 \cdot 10^{-18}$ providing confidence in our estimates using $\nu = 10^{-2}$ (Figure 4B). The results also indicated that $\sim 10^{18}$ randomized samples would have been needed to get an accurate estimate using permutation tests.

For this data, we computed $\eta = 0.024$ suggesting that the PB estimates would inflate the p value. Indeed, the PB approximation gave an estimate of $1.370 \cdot 10^{-52}$, indirectly explaining how the binomial distribution used by the authors also resulted in a smaller p value and inflated the significance.

Enrichment of H3K4me3 in Promoters

Guenther et al. (2007) found that 74% of all annotated promoters were enriched for H3K4 tri-methylation, concluding that a large fraction of genes with no detected transcript have promoter-proximal nucleosomes enriched for H3K4me3 modification. To evaluate the statistical significance of this observation, we took the set of regions highly enriched for H3K4me3 in ES cells as the query set (data provided as supplementary information in their paper [Guenther et al., 2007]). However, they did not provide coordinates for the promoters. Therefore, for the reference intervals, we used the collection of all promoters (−5.5Kbp to 2.5 Kbp relative to TSS-transcription initiation site-of all RefSeq genes) as the reference set of intervals. Although with the I_f that we used we did not get the same ratio of overlap as reported in the paper but still the p value was quite significant. At the observed overlap of $\mathcal{K} = 2642$ out of $m = 24889$ reference intervals, the PB p value was $1.775 \cdot 10^{-76}$, while the DP p value with $\nu = 10^{-2}$ is $2.734 \cdot 10^{-82}$. For this example, $\eta = 0.1$ so PB approximation gave a conservative p values as expected (Figure 4C).

Enrichment of Promoters in Promotor-Associated Chromatin States

In a study by Ernst and Kellis (2010), among 51 identified chromatin states, states 1 to 11 were referred to as promoter-associated states because of high enrichment for promoter regions. We tried to compute the p value of enrichment by

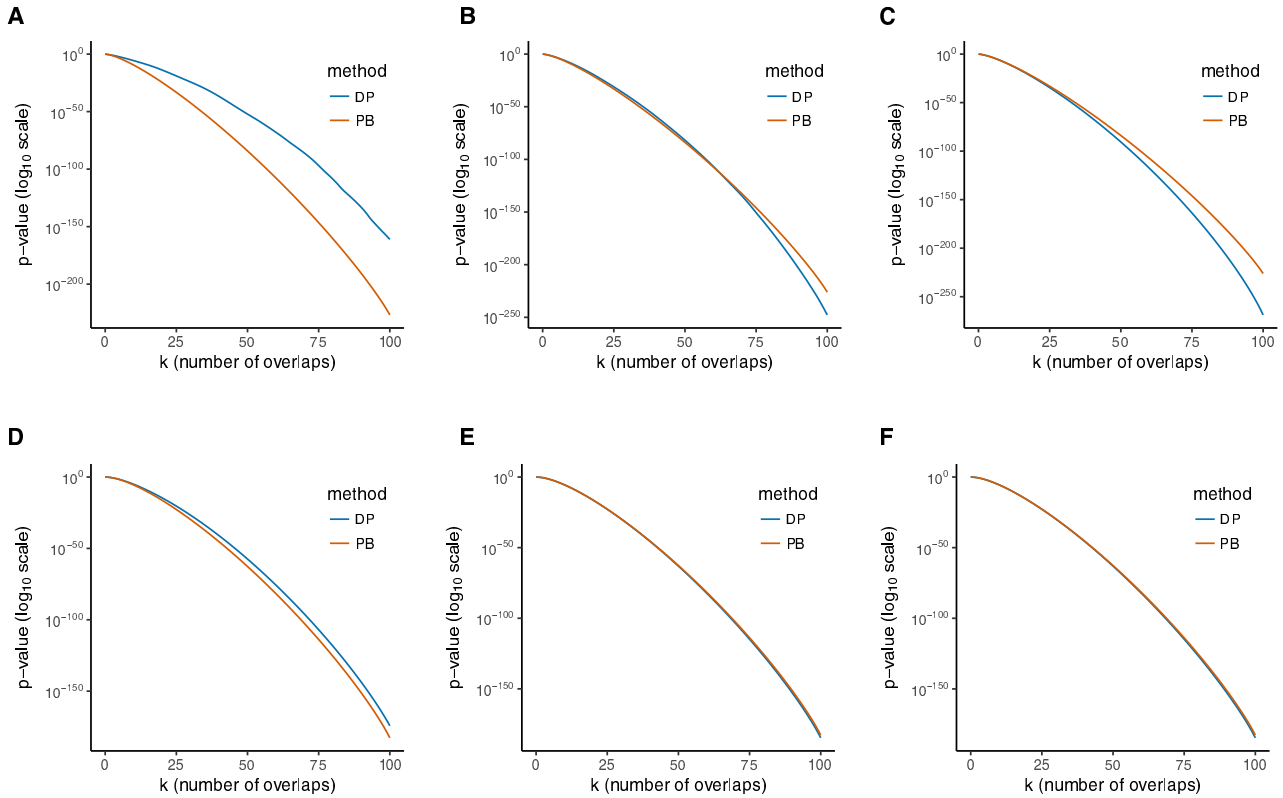


Figure 3. Testing PB Approximation on Simulated Datasets

All simulations are run with $g = 200$ Mbp and $m = 100$. For (A)–(C), we set $n = 100$ and $l_i, x_j \sim \mathcal{U}[1\text{Kbp}, 10\text{Kbp}]$, simulating different η values: (A) $\eta = 0.0054$, (B) $\eta = 0.053$, and (C) $\eta = 0.62$. For (D)–(F) we set $n = 1000$ and $l_i, x_j \sim \mathcal{U}[1\text{Kbp}, 2\text{Kbp}]$, with η: (D) $\eta = 0.0079$, (E) $\eta = 0.062$, and (F) $\eta = 0.68$.

considering the set of all promoter regions (within 2 Kbp of RefSeq TSS) as the query set I and 200-bp intervals identified with state 9 as the reference set I_f. From $m = 4995$ intervals in I_f, $\mathcal{K} = 344$ are overlapped by the query intervals. The $p-$value $= 1.588 \cdot 10^{-8}$ (using the scaling factor $\nu = 10^{-2}$) shows that it would be very unlikely to observe such overlap only by chance, yet it is much less significant than the p value reported by the authors ($\leq 10^{-200}$), computed using the hypergeometric distribution. As $\eta = 0.01$, PB approximation expectedly gives smaller p value ($1.082 \cdot 10^{-13}$; Figure 4D).

DISCUSSION

Our results explore the statistics of interval overlaps. The question is quite natural in the post-genomic era where annotating the genome for function, structure, and variation and identifying correlated annotations are key problems. While scientists have used many different ways to compute the significance of overlap between two sets of intervals, their computations often do not explicitly state the assumptions on the null model or accurately compute the p values given specific assumptions.

To the best of our knowledge, the p value computation for sets of overlapping intervals has been limited to either permutation tests, which do not scale to computation of small p values, or simple parametric tests such as hypergeometric or binomial tests, which are based on simplifying assumptions about the length and structure of intervals. Our paper, however, formulates a null model where the size of intervals and their relative arrangement are considered when the significance of overlap is evaluated. We explicitly state the assumptions that we have made in our proposed model and assess the impact of our assumptions thorough the experiments on simulated and real datasets. Computation of exact p values may be necessary in some cases. For example, p values can be used to compare the significance of two "competing" annotations with different numbers of intervals (n) and intersections (k). We develop a framework that makes exact computation of p values possible, even for very small p values.

The proposed DP method is able to compute very small p values by efficiently counting the number of possible random rearrangements of intervals resulting in a specific amount of overlap. Although we assume that the order of intervals is not changed and it may be possible to construct adversarial examples where changing the order has a material impact on p values, but our simulation of typical examples of interval data show that the resulting change in p values is not significant. Our experiments on simulated and real datasets also suggest that to improve the speed and memory usage, we can employ reasonable scaling factors and still obtain accurate p values.

The PB approximation is very efficient to compute. However, our results suggest that for typical values found in real-life examples, the independence assumption is too strong and might

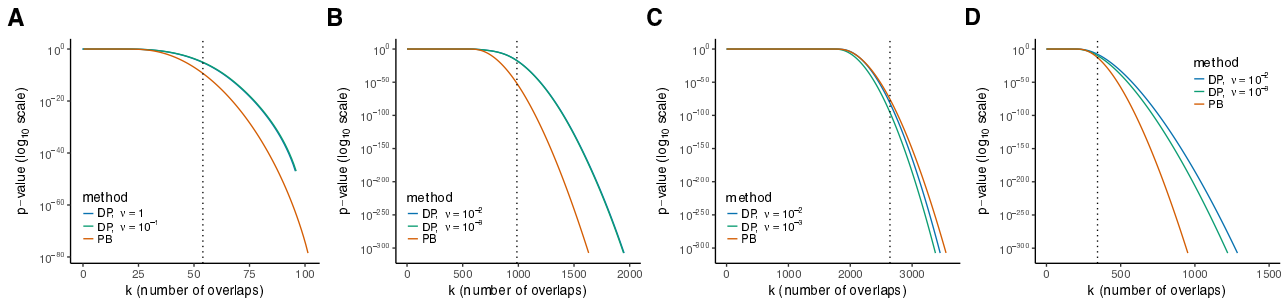


Figure 4. Enrichment Analysis on 4 Biological Datasets from Published Studies

The p value curve is plotted for a range of overlap statistics k, computed using two different scaling factors (shown in blue and green). The result of PB approximation is also shown in orange. The dashed line shows the observed overlap $\mathcal{K}$ for each dataset. For datasets (A) and (B), the p values computed using both scaling factors are almost identical.

(A) TCGA-CNV enrichment in extra-chromosomal amplified oncogenes; $\mathcal{K} = 54$.

(B) Non-coding genes enrichment in CNVs; $\mathcal{K} = 987$.

(C) Enrichment of H3K4me3 in promoters; $\mathcal{K} = 2642$.

(D) Enrichment of promoters in promoter-associated chromatin states; $\mathcal{K} = 344$.

result in underestimated p values or the false reporting of some overlap as being significant. Nevertheless, we have introduced parameter η, which can be readily computed from the data before running the DP method, to estimate the accuracy of the PB method compared to the DP algorithm results. Future work should look into more systematic characterization of PB approximation.

Throughout our experiments, we let the intervals be uniformly distributed over the whole extent of the chromosomes. However, one might be interested in a non-uniform distribution of intervals under the null model, to account for confounding variables such G/C content, sequence context, or intergenic-genic region. Our methods can be used in such cases by confining the problem to the specific regions of interest. Hence, only intervals falling into such regions are considered, and g would be the total length of the segments that intervals are allowed to be distributed there. Moreover, we considered the overlap of two intervals as a binary event and defined the statistic based on the number of overlapping intervals. However, the DP method can be modified to compute the p value when the overlap statistic is defined based on the total amount of shared base pairs instead. Thus, we provide this as an option in ISTAT software and give the user the flexibility of choosing the appropriate measure of overlap for their specific application.

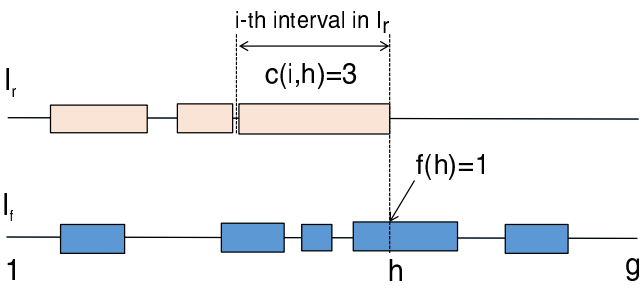


Figure 5. Illustrating the Basics of the Dynamic Programming Algorithm

How the functions $c(i, h)$ and $f(h)$ are evaluated. In this example, the i-th interval in I_r, which ends at h, intersects 3 intervals in I_f, so $c(i, h) = 3$. Also, there is an interval in I_f spanning h, so $f(h) = 1$.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- CONTACT FOR REAGENT AND RESOURCE SHARING
- METHOD DETAILS
  - Problem Formulation
  - Dynamic Programming Algorithm
  - Time Complexity
  - Multiple Chromosomes
  - Poisson Binomial Approximation
- DATA AND SOFTWARE AVAILABILITY

ACKNOWLEDGMENTS

We thank the editor and reviewers for their constructive comments. S.S. and V.B. were supported by grants from the NSF (IIS-1815485 and DBI-1458557) and the NIH (GM114362).

AUTHOR CONTRIBUTIONS

S.S. and V.B. conceived the idea and developed the algorithms. S.S. implemented the software and performed the experiments. Both authors contributed to the analysis of results and writing of the manuscript.

DECLARATION OF INTERESTS

V.B. is a co-founder and has an equity interest in Pretzel Therapeutics, Inc. and Digital Proteomics, LLC, and receives income from Digital Proteomics. The terms of this arrangement have been reviewed and approved by the University of California, San Diego, in accordance with its conflict-of-interest policies. Pretzel Therapeutics and Digital Proteomics were not involved in the research presented here.

Received: April 2, 2019
Accepted: May 14, 2019
Published: June 12, 2019

REFERENCES

Bartel, D.P. (2009). MicroRNAs: target recognition and regulatory functions. Cell *136*, 215–233.

Davoli, T., Uno, H., Wooten, E.C., and Elledge, S.J. (2017). Tumor aneuploidy correlates with markers of immune evasion and with reduced response to immunotherapy. Science *355*, eaaf8399.

Deshpande, V., Luebeck, J., Bakhtiari, M., Nguyen, N.-P.D., Turner, K.M., Schwab, R., Carter, H., Mischel, P.S., and Bafna, V. (2018). Reconstructing and characterizing focal amplifications in cancer using AmpliconArchitect. bioRxiv.

ENCODE Project Consortium (2007). Identification and analysis of functional elements in 1% of the human genome by ENCODE pilot project. Nature *447*, 799–816.

ENCODE Project Consortium (2012). An integrated encyclopedia of DNA elements in the human genome. Nature *489*, 57–74.

Ernst, J., and Kellis, M. (2010). Discovery and characterization of chromatin states for systematic annotation of the human genome. Nat. Biotechnol. *28*, 817–825.

Feuk, L., Carson, A.R., and Scherer, S.W. (2006). Structural variation in the human genome. Nat. Rev. Genet. *7*, 85–97.

Guenther, M.G., Levine, S.S., Boyer, L.A., Jaenisch, R., and Young, R.A. (2007). A chromatin landmark and transcription initiation at most promoters in human cells. Cell *130*, 77–88.

Hong, Y. (2013). On computing the distribution function for the poisson binomial distribution. Comp. Stat. Data Anal. *59*, 41–51.

Ingolia, N.T., Brar, G.A., Rouskin, S., McGeachy, A.M., and Weissman, J.S. (2012). The ribosome profiling strategy for monitoring translation in vivo by deep sequencing of ribosome-protected mRNA fragments. Nat. Protoc. *7*, 1534–1550.

Ingolia, N.T., Ghaemmaghami, S., Newman, J.R., and Weissman, J.S. (2009). Genome-wide analysis in vivo of translation with nucleotide resolution using ribosome profiling. Science *324*, 218–223.

Jurka, J. (2000). Repbase update: a database and an electronic journal of repetitive elements. Trends in Genet. *16*, 418–420.

The Cancer Genome Atlas Program - National Cancer Institute (n.d.). https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga.

Pinkel, D., Segraves, R., Sudar, D., Clark, S., Poole, I., Kowbel, D., Collins, C., Kuo, W.L., Chen, C., Zhai, Y., et al. (1998). High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. Nat. Genet. *20*, 207–211.

Turner, K.M., Deshpande, V., Beyter, D., Koga, T., Rusert, J., Lee, C., Li, B., Arden, K., Ren, B., Nathanson, D.A., et al. (2017). Extrachromosomal oncogene amplification drives tumour evolution and genetic heterogeneity. Nature *543*, 122–125.

Venter, J.C., Adams, M.D., Myers, E.W., Li, P.W., Mural, R.J., Sutton, G.G., Smith, H.O., Yandell, M., Evans, C.A., Holt, R.A., et al. (2001). The sequence of the human genome. Science *291*, 1304–1351.

Verhaak, R.G.W., Bafna, V., and Mischel, P.S. (2019). Extrachromosomal oncogene amplification in tumour pathogenesis and evolution. Nat. Rev. Cancer *19*, 1.

Wang, Y.H. (1993). On the number of successes in independent trials. Stat. Sinica, 295–312.

Wu, X., Li, Y., Crise, B., and Burgess, S.M. (2003). Transcription start regions in the human genome are favored targets for MLV integration. Science *300*, 1749–1751.

Zarrei, M., MacDonald, J.R., Merico, D., and Scherer, S.W. (2015). A copy number variation map of the human genome. Nat. Rev. Genet. *16*, 172–183.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited Data		
TCGA-CNV enrichment in extra-chromosomal DNA	Turner et al. (2017)	Supplemental information
Non-coding genes enrichment in CNVs	Zarrei et al. (2015)	Table S9
Enrichment of H3K4me3 in promoters	Guenther et al. (2007)	Table S2
Enrichment of promoters in promoter-associated chromatin states	Ernst and Kellis (2010)	http://compbio.mit.edu/ChromatinStates/
Software and Algorithms		
iSTAT	This paper	https://github.com/shahab-sarmashghi/Skmer

CONTACT FOR REAGENT AND RESOURCE SHARING

Further information and requests for resources should be directed to and will be fulfilled by the Lead Contact, Vineet Bafna (vbafna@cs.ucsd.edu).

METHOD DETAILS

We use the space-counted, zero-start convention for the genomic coordinates. Namely, we count the space between bases starting from 0 (the one before the first base) up to g (the one after the last base), where g denotes the length of the genomic region of interest. We use 'i' to index the intervals in query set I, which has total number of n intervals, and designate 'j' to index the intervals in reference set I_f, which consists of m intervals in total. The length of i-th query interval and j-th reference interval are represented by l_i and x_j, respectively. Two intervals (u_1, u_2) and (v_1, v_2) overlap iff they share common nucleotide(s). A collection of intervals is *non-overlapping* if no pair of intervals in the collection overlap.

Problem Formulation

Let $I_f \subseteq I$ denote the subset of intervals in I_f that are *hit* (overlap with intervals in I). Suppose $|I_f \subseteq I| = k$. We measure the significance (p value) of this observation by sampling a random set of intervals I_r with the following properties

- $|I_r| = |I|$. I_r has exactly n elements.
- Intervals in I_r have the same lengths as the intervals in I.
- The location of intervals in I_r are drawn from a distribution (implicitly) such that all possible random sets are equally likely.

Let I_r be drawn according to the process above, then p value is defined as

$$P - \text{value}(k) = \Pr(|I_f \subseteq I_r| \geq k).$$

While the computational complexity of the problem is not known, we can argue that it is hard. Clearly, the number of possible random sets is very large; ranging from $\binom{g+n-\sum_i l_i}{n}$ when all l_i's are identical, to $\binom{g+n-\sum_i l_i}{n} n!$ when all l_i are distinct. For typical values of $g = 2 \cdot 10^8$ (length of a chromosome), $n = 100$ (number of annotated regions), and $\sum l_i = 10^6$ (total length of regions covered by an annotation), counting all possibilities naively to compute $\Pr(|I_f \subseteq I| \geq k)$ is computationally intractable. Thus, we impose the restriction that the intervals in I_r must retain the same order as the intervals in I (i.e, if interval B starts after interval A in I, same should happen in I_r), and present a dynamic programming (DP) algorithm to compute the number of distinct random sets with $|I_f \subseteq I_r| = k$, for all k. In practice, to apply the algorithm to large genomes with abundant annotation, we use a practical interval 'scaling' scheme by considering the natural partitioning of the genome into intervals and the gaps amidst them, and scale each interval and gap in I and I_f by a fraction ν. Ideally, we want to have $\nu = 1$, but large problems require smaller fractions to make the computation feasible from both running time and memory usage aspects. Nevertheless, we show that the algorithm still yields a close approximation of p value.

Dynamic Programming Algorithm

For interval i in I_r, genomic location h, $(1 \leq h \leq g)$, $0 \leq k \leq m$, $a \in 0, 1$, let $N(i, h, k, a)$ denote the number of arrangements of the first i intervals in I_r such that (see Figure 5):

- The i-th interval ends exactly at location h.
- k intervals in I_f are hit by the first i intervals in I_r.
- $a = 0$ if the interval from I_f that spans h (if any) has not been counted earlier; $a = 1$ otherwise.

We also define $N_1(i, h, k, a)$ identically to $N(i, h, k, a)$ with the exception that the i-th interval ends at or before location h. Note that if the j-th interval in I_f spans h, it is counted as a hit, but may have already been counted by some other interval in I_r. Although a separate function can be defined to store that information, we use a as an indicator in dynamic programming for the sake of brevity. In order to compute $N_1(i, h, k, a)$, we must define some auxiliary functions. Let $c(i, h)$ denote the number of intervals in I_f which intersect with $(h - l_i, h)$ in I_r. While evaluating $c(i, h)$, (j_1, j_2) in I_f is counted as an intersecting interval with $(h - l_i, h)$ if $j_1 < h$ and $j_2 > h - l_i$. We also define binary function $f : (0, g] \rightarrow \{0, 1\}$, where $f(h) = 1$ if some interval in I_f spans h, meaning that it starts before h and ends after it, and $f(h) = 0$ otherwise (Figure 5). For the simplicity of exposition, it is assumed that a single nucleotide overlap between two intervals from I_r and I_f is sufficient to count the reference interval as intersected. However, we can be more strict by accepting only the overlaps which include z or more base pairs (units). In that case, we just need to generalize the definitions of $c(i, h)$ and $f(h)$. For $c(i, h)$, the intersection conditions should change to $j_1 \leq h - z$ and $j_2 \geq h - l_i + z$, which can be compressed into a single condition $\min\{j_2, h\} - \max\{j_1, h - l_i\} \geq z$. Also, $f(h) = 1$ if $j_1 \leq h - z$ and $j_2 \geq h + z$.

To explain the recurrences, note that $N_1(i, h, k, a)$ can be computed by adding cases where the i-th interval ends exactly at h, and cases where the i-th interval ends strictly before h. To compute $N(i, h, k, a)$ we need to consider all arrangements where the first $i - 1$ intervals in I_r ends before the start of the i-th interval at $h - l_i$.

$$N_1(i,h,k,a) = \begin{cases} N(i,h,k,a) & h = 1 \\ N(i,h,k,a) + N_1(i,h-1,k,\min\{a, f(h-1)\}) & \text{Otherwise} \end{cases} \quad \text{(Equation 1)}$$

$$N(i,h,k,a) = \begin{cases} 0 & h < \sum_{x=1}^{i} l_x \text{ or } k < c(i,h) - a \\ 1 & i = 1 \text{ and } k = c(i,h) - a \\ N_1(i-1, h-l_i, k-c(i,h)+a, f(h-l_i)) & \text{Otherwise} \end{cases}$$

where

$$1 \leq i \leq n, \quad 1 \leq h \leq g, \quad 0 \leq k \leq m, \quad a \in \{0, 1\}.$$

The *DP* p value $(\Pr(|I_f \subseteq I_r| \geq k))$ can be computed using the ratio

$$P - \text{value}(k) = \frac{\sum_{\kappa=k}^{m} N_1(n, g, \kappa, 0)}{\sum_{\kappa=0}^{m} N_1(n, g, \kappa, 0)}.$$

Recall that the total number of configurations is

$$\sum_{\kappa=0}^{m} N_1(n, g, \kappa, 0) = \binom{g - \sum_{i=1}^{n} l_i + n}{n}.$$

which can be very large and surpass the upper limit of ordinary data types. Therefore, we perform all calculations using a logarithmic scale. The multiplication and division can be done trivially in the logarithmic scale. For the addition and subtraction, we use the following simple math trick. Let $a = \log A$ and $b = \log B$. Then, $c = \log(A \pm B)$ is calculated without explicitly converting a and b to their intractably large counterparts, A and B, using

$$c = \begin{cases} a + \log(1 \pm \exp(b - a)) & \text{if } b > a \\ b + \log(1 \pm \exp(a - b)) & \text{if } a > b \end{cases}$$

As a matter of fact, this trick is useful when A and B are both large, but the ratio $\frac{A}{B} = \exp(a - b)$ is computable, which is the case in the recurrence relation given by Equation 1.

Time Complexity

The number of iterations to complete the table of values for $N_1(i, h, k, a)$ is $\mathcal{O}(ngm)$. The functions $c(i, h)$ and $f(h)$ can be pre-computed (using a modified version of binary search algorithm), so each iteration is computed in a constant time. Therefore, the total time complexity is $\mathcal{O}(ngm)$ which is pseudo-polynomial because the input size is $\mathcal{O}((n + m)\log g)$. The running time can be reduced to $\mathcal{O}(ng\nu m)$ by scaling the genome using scaling factor $\nu < 1$. We also use a number of tricks to improve the speed of computations, including lowering memory usage from $\mathcal{O}(ngm)$ to $\mathcal{O}(gm)$. We should note that this time complexity is achieved under the assumption

that the order of intervals in I_r is same as I. In Results, we show that choosing different orders does not significantly change the p value.

Multiple Chromosomes

In many cases of interest, the intervals reported are on multiple chromosomes, with a non-uniform distribution across chromosomes. Therefore, the appropriate random interval set I'_r may only allow permutation of interval positions within the chromosome it is originally assigned to. For this alternative null model, the DP algorithm is applied to each chromosome to enumerate rearrangements of intervals within each chromosome, and then the results are combined to compute the overall p value. Specifically, consider Q chromosomes. For an arbitrary chromosome $q, 1 \leq q \leq Q$, let $I_q \subseteq I$ and $I_{f,q} \subseteq I_f$ denote the subsets of intervals paced on q, containing n_q and m_q intervals, respectively. Similarly, we can define $I_{r,q}$ to be a random reordering of I_q on chromosome q. Let $N_q(k_q)$ denote the number of configurations of intervals in $I_{r,q}$ s.t. $|I_{f,q} \subseteq I_{r,q}| = k_q$. Using dynamic programming on each of Q chromosomes, we can obtain $N_q(k_q)$ $1 \leq q \leq Q$, $0 \leq k_q \leq m_q$. For $k \in [0, m]$ we define the p value to be

$$P-\text{value}(k) = \Pr\left(\sum_{q=1}^{Q} k_q \geq k\right).$$

With the equiprobability assumption and using simple arguments based on multiplication principle to count the number of desired configurations, we can compute the p value as

$$P-\text{value}(k) = \frac{\sum_{(k_1,k_2,\ldots,k_Q)\in T_k} \prod_{q=1}^{Q} N_q(k_q)}{\sum_{(k_1,k_2,\ldots,k_Q)\in T_0} \prod_{q=1}^{Q} N_q(k_q)},$$

where T_k is the set of all Q-tuples $(k_1, k_2, \ldots, k_Q)$ such that $\sum_{q=1}^{Q} k_q \geq k$. While the denominator can be easily computed via the following identity

$$\sum_{(k_1,k_2,\ldots,k_Q)\in T_0} \prod_{q=1}^{Q} N_q(k_q) = \prod_{q=1}^{Q} \sum_{k_q=0}^{m_q} N_q(k_q),$$

it is not efficient to iterate over T_k to compute the numerator for each k. Instead, we use a simple recursive procedure to compute it. Let $M(q, k)$ be the number of configurations that the first q chromosomes have k intersections. The p value can be expressed in terms of $M(q, k)$ as

$$P-\text{value}(k) = \frac{\sum_{\kappa=k}^{m} M(Q,\kappa)}{\sum_{\kappa=0}^{m} M(Q,\kappa)}.$$

The following recurrence relation lets us to efficiently compute the p value for all $k \in [0, m]$

$$M(q,k) = \sum_{l=0}^{\min\{k,m_q\}} M(q-1, k-l) N_q(l)$$

$$M(q,0) = \prod_{u=1}^{q} N_u(0), \quad M(1,k) = N_1(k)$$

where the time complexity is $\mathcal{O}(Qm^2)$. Nevertheless, in almost all practical cases, the total time complexity of calculating the p value is dominated by the complexity of applying DP algorithm to each chromosome to compute all $N_q(k_q)$. As DP algorithm on each chromosome is done independently, we can take advantage of parallel computing and the total running time would be $\mathcal{O}\left(\max_q \{n_q g_q m_q\}\right)$.

Poisson Binomial Approximation

For the case that annotations contain too many intervals such that the processing resources to run DP algorithm cannot be afforded, we provide an approximation which is reasonably close under certain condition. For simplicity, we remove the non-overlapping assumption on I_r. Thus, I_r is a randomly located collection of n intervals of lengths $l_1, l_2, l_3, \ldots, l_n$ with arbitrary order. Let E_{ij} denote the event that the j-th interval in I_f is intersected by the i-th interval in I_r. Then,

$$p_{ij} := \Pr(E_{ij}) = \frac{l_i + x_j - 1}{g}$$

As before, we assumed that a single nucleotide overlap is sufficient. For the more strict overlap condition of at least z base pairs overlap, p_{ij} is given by

$$p_{ij} = \begin{cases} 0 & \text{if } z > \min\{x_j, l_i\} \\ \dfrac{l_i + x_j - 2z + 1}{g} & \text{Otherwise} \end{cases}$$

Now, let $\overline{E}_{ij}$ be the event that the i-th interval in I_r does not intersect the j-th interval in I_f. In the absence of the non-overlapping assumption on I_r, the events $\overline{E}_{ij}, i = 1, 2, \ldots, n$, are independent, and the probability of their intersection is given by the product of individual probabilities. Therefore, the probability of $E_j = \cup_{i=1}^{n} E_{ij}$, which is the event where interval $j \in I_f$ is hit by I_r, can be calculated as

$$P_j := \Pr(E_j) = \Pr\left(\cup_{i=1}^{n} E_{ij}\right) = 1 - \Pr\left(\cap_{i=1}^{n} \overline{E}_{ij}\right) = 1 - \prod_{i=1}^{n} \Pr(\overline{E}_{ij}) = 1 - \prod_{i=1}^{n} (1 - \Pr(E_{ij})). \quad \text{(Equation 2)}$$

Now consider the binary indicator variable X_j, where $X_j = 1$ iff event E_j occurs. We have m Bernoulli experiments with success probabilities $P_1, P_2, \ldots, P_m$, and we are interested in computing $\Pr\left(\sum_j X_j = k\right)$. In general, there are dependencies between E_j's for different values of j. However, under certain condition where intervals are not too close or far away, we can approximately assume independence between different intervals. The sum of m independent Bernoulli trials with different success probabilities is a Poisson binomial (PB) distribution (Wang, 1993).

$$\Pr\left(\sum_{j=1}^{m} X_j = k\right) = \sum_{A \in F_k} \prod_{u \in A} P_u \prod_{v \in A^c} (1 - P_v) \quad \text{(Equation 3)}$$

where F_k is the set of all subsets of {1,2,…,m} with k elements. Equation 3 allows us to compute the p value as

$$P - \text{value}(k) = \Pr\left(\sum_{j=1}^{m} X_j \geq k\right).$$

We cannot directly use Equation 3 by enumerating over all elements in F_k, but use a recursive approach to compute it, following Hong (2013). It is reproduced here for completeness. Let $\pi_{k,j} = \Pr(\sum_{u=1}^{j} X_u = k)$ denote the probability of getting k hits in the first j intervals in I_f. Our goal is to compute $\Pr(\sum_{u=1}^{m} X_u = k) = \pi_{k,m}$. All values $\pi_{k,j}$ can be computed in $\mathcal{O}(m^2)$ time using

$$\pi_{k,j} = P_j \pi_{k-1,j-1} + (1 - P_j)\pi_{k,j-1}, \quad 0 \leq k \leq m, \ 0 \leq j \leq m \text{ with the boundary conditions} \quad \text{(Equation 4)}$$
$$\pi_{-1,j} = \pi_{j+1,j} = 0, j = 0, 1, \ldots, m \text{ and } \pi_{0,0} = 1. \text{ Other FFT based methods are also applicable (Hong, 2013).}$$

With the above PB approximation, we assume that the event of an interval in I_f being hit is independent of other intervals being hit, greatly reducing the computational complexity of the problem. To understand the impact of this assumption, we introduce parameter η. Recall that $P_j = \Pr(E_j) = \Pr(X_j = 1)$ is the probability that interval j (length x_j) in I_f is hit by some interval in I_r. Let d_j denote the distance of interval j from interval $j - 1$. Define $\Delta := (m - 1) \cdot \textbf{median}\{d_j \mid j = 2, 3, \ldots, m\}$, and $\eta := \frac{\Delta}{g}$. Parameter η is a measure of the spread of intervals in I_f. For $\eta \ll 1$, and j' sufficiently close to j, we expect to have

$$\Pr(X_j = 1 \mid X_{j'} = 1) > \Pr(X_j = 1).$$

In other words, if intervals in I_f are clumped, then $E_j, E_{j'}$ are not statistically independent but positively correlated, and we will underestimate the true p value. For larger values of η, and j,j' sufficiently distant,

$$Pr(X_j = 1 \mid X_{j'} = 1) < Pr(X_j = 1),$$

The negative correlation leads to an over-estimation of the p value. To better recognize this effect, imagine an extreme case where $n < m$ and due to the size and spread of intervals in I_f, at most n intervals in I_f can be hit. Therefore, $p - \text{value}(n + 1) = \Pr\left(\sum_j X_j > n\right) = 0$. The independence assumption in PB computation, though, will lead to a non-zero value (over-estimate) for $p - \text{value}\ (n + 1)$.

DATA AND SOFTWARE AVAILABILITY

The iSTAT software is made publicly available on https://github.com/shahab-sarmashghi/ISTAT.git