OLOGRAM : Determining significance of total overlap length between genomic regions sets

Q Ferré, G. Charbonnier, N Sadouni, F Lopez, Y Kermezli, S. Spicuglia, C Capponi, B. Ghattas, D. Puthier

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Application Note

OXFORD

Genome analysis

OLOGRAM : Determining significance of total overlap length between genomic regions sets

Q. Ferré [1,2,3,†], G. Charbonnier [1,3,†], N. Sadouni [1,3], F. Lopez [1,3], Y. Kermezli [1,3,4], S. Spicuglia [1,3], C. Capponi [2], B. Ghattas [5], D. Puthier [1,3,*]

[1] Aix Marseille Univ, INSERM, UMR U1090, TAGC, Marseille, France, [2] Aix Marseille Univ, CNRS, UMR 7020, LIS, Qarma, Marseille, France, [3] Equipe Labellisée LIGUE contre le Cancer, [4] Tlemcen University, The Laboratory of Applied Molecular Biology and Immunology, Algeria, [5] Aix Marseille Univ, CNRS, UMR 7373, IMM, Marseille, France.

[*] To whom correspondence should be addressed. [†] These authors contributed equally.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Various bioinformatics analyses provide sets of genomic coordinates of interest. Whether two such sets possess a functional relation is a frequent question. This is often determined by interpreting the statistical significance of their overlaps. However, only few existing methods consider the lengths of the overlap, and they do not provide a resolutive p-value.

Results: Here, we introduce *OLOGRAM*, which performs overlap statistics between sets of genomic regions described in BEDs or GTF. It uses Monte Carlo simulation, taking into account both the distributions of region and inter-region lengths, to fit a negative binomial model of the total overlap length. Exclusion of user-defined genomic areas during the shuffling is supported.

Availability: This tool is available through the command line interface of the *pygtftk* toolkit. It has been tested on Linux and OSX and is available on Bioconda and from https://github.com/dputhier/pygtftk under the GNU GPL license.

Contact: denis.puthier@univ-amu.fr

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Current genomic analysis methods can localize a variety of sets of genomic regions, such as epigenomic features, resulting in a BED file giving their coordinates. To determine whether two such sets have a functional relationship, a typical approach is to look for significant co-localization by assessing the statistical significance of the amount of overlap between them (Haiminen *et al.*, 2008).

A comprehensive review of such methods is available through the *Coloc-stats* web interface (Simovski *et al.*, 2018), showing the biggest difference between them to be their null model. Many, such as *GREAT* (McLean *et al.*, 2010) or *CEAS* (Ji *et al.*, 2006) use a binomial test considering only the intersections of the peak centers with the query regions, while *BEDTOOLS fisher* (Quinlan and Hall, 2010) uses the number of intersecting "bins" (whose size depends on the input regions) to compute a hypergeometric test.

Generating an empirical null distribution by random shuffling of the regions within the sets is another possibility. For example, *pybedtools* incorporates a wrapper for this (Dale *et al.*, 2011) which was also used to tackle the N-fold overlap problem (Aszódi, 2012). For a more realistic null model, conservation of inter-segment length during the shuffling was first proposed by the *Genomic HyperBrowser* (Sandve *et al.*, 2010). However, the p-value they provide is only empirical and limited in its resolution by shuffling depth, itself limited by computation time.

Here we propose a new method, implemented in a tool named *OLOGRAM (OverLap Of Genomic Regions Analysis using Monte Carlo)*, to conveniently assess the significance of overlaps by fitting a Negative Binomial model on overlap statistics of interest via a Monte Carlo method.

2 Methods

2.1 Permutation and intersection computation

Let A and B be two sets of genomic regions with no overlaps within A nor B. For each subset $E_{A,k}$ (resp. $E_{B,k}$) of A (resp. B) only for chromosome k, let $L(E_{A,k})$ and $I(E_{A,k})$ be respectively the lists of regions' sizes and inter-regions distances (from end to start).

A shuffle is generated by performing independent random permutations of $L(E_{A,k})$ and $I(E_{A,k})$ for all chromosomes separately, and separately for A and B. This method differs from the classical *BEDTOOLS shuffle* which sets regions at random positions. The Genome HyperBrowser showed the relevance of this idea.

Our approach can also exclude regions from the shuffle by shuffling across a shorter, concatenated "sub-genome" generated by removing the excluded regions from both sets. This allows to compute enrichment relative to the genome minus excluded regions. For example, one can remove low mappability regions, or consider only accessible (i.e. DNAse I HyperSensitive) regions.

The tool then computes the regions' intersections between the i^{th} shuffle of A and the i^{th} of B, for all shuffles. This is done in RAM with a custom sweep-line (Shamos and Hoey, 1976) algorithm of $O(n)$ complexity to avoid disk I/O overhead. As intersections are only computed once per shuffle, the use of other algorithms such as Interval Trees with $O(n \log(n))$ complexity is not justified.

2.2 Discussion of statistical modeling

The null hypothesis (H_0) is that the regions of A are located independently of B. As such, we do not expect them to overlap more than expected by chance, if the regions were independently randomly placed on the genome.

Here, we propose a new statistical framework to model this problem. Under (H_0), for all regions A_i of A and B_j of B, consider the Bernoulli random variables $I_{i,j} = \mathbb{1}_{A_i \cap B_j \neq \varnothing}$.

They have very small probabilities $p_{i,j}$ (region sizes are typically small relative to chromosome size), that differ (each region has a different length, hence different intersection probability), and are dependent (the regions do not overlap).

Let N be the number of intersections and S the total number of overlapping nucleotides. Then $N = \sum_{i,j} I_{i,j}$ is a sum of dependant Bernoulli r.v. and can be modeled with a beta-binomial (Yu and Zelterman, 2008), itself modeled with a Negative Binomial. Unlike with *BEDTOOLS shuffle*, the dependency of the $I_{i,j}$ makes Poisson modeling unadapted.

Then consider $S = \sum_{i,j} \Lambda_{i,j}$ where $\Lambda_{i,j}$ is the length of the intersection between A_i and B_j. This sum has N nonzero terms, making it a Compound Negative Binomial. Furthermore, empirically $\Lambda_{i,j}$ will often follow a logarithmic distribution, so S can be approximated via a negative binomial (Omair *et al.*, 2018).

The assumptions taken here are confirmed in practice by a fitting test. Consequently, we reckon our model is plausible with N and S following negative binomial distributions of under (H_0) unknown parameters, approximated via this Monte Carlo approach. As such, we use them as test statistics: the p-value associated to their value in the true data is used to accept or reject the alternative hypothesis (H_1) that the regions of the query tend to overlap the reference.

3 Implementation

Our method is implemented as a plugin to *pygtftk* (Lopez *et al.*, 2019) and can be passed a GTF/BED stream or file (examples in documentation and Supplementary Data). Most of the code is written in Python 3, with performance-critical operations written in C++ and/or Cython (Behnel *et al.*, 2011). To preserve RAM, the total number of shuffles to be computed is divided into batches.

The tool will compute the overlap between the supplied BED region file and *(i)* any desired GTF feature, or *(ii)* features derived from GTF file attributes (e.g "gene_biotype"), or *(iii)* additional regions supplied as BEDs. It will output overlap statistics and the associated p-values.

The computing cost scales with the total number of lines in the reference and query files. A typical pairwise enrichment analysis of 10k regions against 10k takes 62 seconds on an 2,5 GHz Intel Core i7 processor. 200k against 200k takes 11 minutes.

3.1 Results

Suppl. Table 1 presents the applicability conditions and functionalities of various tools and approaches including *GREAT*, *CEAS*, *Bedtools Fisher*, *Genomic HyperBrowser* and *LOLA* (Sheffield and Bock, 2016).

An example of *OLOGRAM* output is available in *Suppl. Fig. 1*. We showcase interactions with *pygtftk* in *Suppl. Fig. 2*, and the importance of considering both S and N in *Suppl. Fig. 3*.

Using biological and artificial testing data, we found both S and N indeed follow a negative binomial distribution; this is shown in particular in *Suppl. Fig. 4* with the example of S on artificial data. A small total number of shuffles results in a noisy distribution, but whose two first moments (expectation, variance) remain similar than with a larger number of shuffles, making them sufficient to estimate the underlying distributions. We believe 200 shuffles (default parameter) to be an acceptable compromise between computing cost and precision of evaluation in most cases.

Fitting a distribution (as opposed to an empirical p-value) allows for better assessment of extreme overlaps presumably not encountered while shuffling. To confirm the goodness of fit, a fitting quality is given as $1 - V$ where V is Cramér's V score (Cramér, 1946) for the contingency table of observed vs. expected histogram bins. It works best when the individual probability of intersection is not too small, meaning the query and reference regions are not too small and/or scarce compared to each other.

We compare our tool to other existing approaches in *Suppl. Table 2*, showing that OLOGRAM can provide meaningful insights by being resolutive at low p-values. Discussion of those results can be found in *Suppl. Note 1*. The full code to reproduce the analyses presented is available at : https://github.com/dputhier/ologram_supp_mat, showcasing Snakemake integration.

4 Conclusion

We have implemented a method which allows to consider the information found in the number of overlapping base pairs, with a shuffling paradigm that conserves inter-region length, used to fit a negative binomial model. New features are being developed, including support for multiple overlaps between $n \geq 2$ sets.

Funding

Q.F, G.C., N.S., S.S. and D.P. were supported by recurrent funding from INSERM and Aix Marseille Univ and specific grants from A*MIDEX (A-M-AAP-EI-17-63-170228-17.32-SPICUGLIA-HLS), Institut National du Cancer (PLBIO018-031 INCA_12619) and Ligue contre le Cancer (Equipe Labellisée). Y.K. was supported by the Franco-Algerian partenariat Hubert Curien (PHC) Tassili (15MDU935)

References

Aszódi, A. (2012). MULTOVL: fast multiple overlaps of genomic regions. *Bioinformatics*, **28**(24), 3318–3319.

Behnel, S. *et al.* (2011). Cython: The best of both worlds. *Computing in Science Engineering*, **13**(2), 31 –39.

Cramér, H. (1946). *Mathematical Methods of Statistics*. Princeton University Press.

Dale, R. K. *et al.* (2011). Pybedtools: a flexible python library for manipulating genomic datasets and annotations. *Bioinformatics*.

Haiminen, N. *et al.* (2008). Determining significance of pairwise co-occurrences of events in bursty sequences. *BMC bioinformatics*, **9**, 336.

Ji, X. *et al.* (2006). CEAS: cis-regulatory element annotation system. *Nucleic Acids Research*, **34**, W551–W554.

Lopez, F. *et al.* (2019). Explore, edit and leverage genomic annotations using python GTF toolkit. *Bioinformatics*.

McLean, C. Y. *et al.* (2010). GREAT improves functional interpretation of cis-regulatory regions. *Nature Biotechnology*, **28**(5), 495–501.

Omair, M. A. *et al.* (2018). A bivariate model based on compound negative binomial distribution. *Revista Colombiana de Estadística*, **41**(1), 87–108.

Quinlan, A. R. and Hall, I. M. (2010). BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, **26**(6), 841–842.

Sandve, G. K. *et al.* (2010). The genomic HyperBrowser: inferential genomics at the sequence level. **11**(12), R121.

Shamos, M. I. and Hoey, D. (1976). Geometric intersection problems. In *17th Annual Symposium on Foundations of Computer Science (sfcs 1976)*, pages 208–215.

Sheffield, N. C. and Bock, C. (2016). LOLA: enrichment analysis for genomic region sets and regulatory elements in r and bioconductor. **32**(4), 587–589.

Simovski, B. *et al.* (2018). Coloc-stats: a unified web interface to perform colocalization analysis of genomic features. *Nucleic Acids Research*, **46**, W186–W193.

Yu, C. and Zelterman, D. (2008). Sums of exchangeable bernoulli random variables for family and litter frequency data. *Computational Statistics & Data Analysis*, **52**(3), 1636–1649.