Looking for Similar Patterns in Genomic Sequences*

Piero Montanari[1], Ilaria Bartolini[1], Paolo Ciaccia[1], Marco Patella[1], Stefano Ceri[2], and Marco Masseroli[2]

[1] DISI - Università di Bologna, Italy
[2] DEIB - Politecnico di Milano, Italy

Abstract. Genomics is opening many interesting practical and theoretical computational problems; one of them is the search for a collection of genomic regions at given distances from each other, i.e., a *pattern* of genomic regions. We designed and implemented an optimized pattern-search algorithm able to find efficiently, within a large set of genomic data, genomic region sequences which are similar to a given pattern and present its applicability to the problem of enhancer detection.

1 Introduction

Thanks to Next Generation Sequencing (NGS), huge repositories of genomic sequences are nowadays being collected by large consortia of research laboratories, e.g., ENCODE [2] and TCGA [9]. So far, bio-informatics has been challenged by NGS *primary analysis* (production of sequences in the form of short DNA segments, or "reads") and *secondary analysis* (alignment of reads to a reference genome and extraction of specific genomic features), yet the most important emerging problem is *tertiary analysis*, concerned with multi-sample processing, annotation and filtering of variants, and genome browser-driven exploratory analysis [7]. Tertiary analysis targets data produced by secondary analysis and is responsible of *sense making*, e.g., how genomic regions interact with each other.

The GenData 2020 project addresses this challenge by enabling queries and analysis of processed genomic data. Among the project's results are a *Genometric Data Model* (GDM), which encodes genomic data in terms of their regions and metadata, and a *Genometric Query Language* (GMQL) [4]. According to GDM, a genomic (DNA) *region* is a quadruple $\langle chr, left, right, strand \rangle$, in which *chr* represents the region chromosome, the *strand* (either + or −) can be missing, and the region includes all the DNA nucleotides whose position is between *left* and *right*. Moreover, typically a region is associated by secondary analysis with a *feature vector*, where each feature is extracted by suitable processing.

Several tertiary analysis problems consist of searching for *patterns of regions*, i.e., co-occurrences of certain region configurations, among which the *search for enhancers* (particular regions of the non-coding part of the genome playing the role of enhancing or repressing gene expressions) and the *investigation on 3D*

* This work is supported by the PRIN Project GenData 2020 (`http://www.bioinformatics.deib.polimi.it/gendata/`).

properties of the genome. Informally, a pattern is a collection of regions that are present on one or more *tracks*, where a track is an ordered sequence of regions produced by a specific NGS experiment. All regions of a pattern are placed within the same chromosome, but they need not to be contiguous on the original track.

In this paper we introduce an efficient *pattern-search algorithm* which provides biologists with the ability, once they identify an interesting genomic region pattern, to look for *similar* occurrences of such pattern in the whole genome.[3] Each result of our algorithm is a collection of regions with properties similar to the query pattern, in particular with (approximately) the same spatial configuration (*structural similarity*) and similar feature values (*region similarity*). As Figure 1 suggests, structural similarity ignores absolute coordinate values of regions, focusing on *inter-region distances*.

Fig. 1. Example of search result on target tracks with high structural similarity to the query pattern: inter-region distances of the pattern are (approximately) preserved in the regions of the result, both along the same target track and across different tracks.

2 The Base Problem

In the *base version* of our problem, the pattern to be searched is single-track, regions are reduced to points, and region features are not present. Formally, a track T is a strictly increasing sequence of N elements, $T = \langle t_1, \ldots, t_N \rangle$, where each $t_i \in \mathbb{N}^+$ and $t_i < t_{i+1}$. Given a "query" track $Q = \langle q_1, \ldots, q_M \rangle$ and a "target" track $T = \langle t_1, \ldots, t_N \rangle$, with $N \geq M$, a *matching of Q in T* is a strictly increasing function $f : [1, M] \to [1, N]$ that assigns to each element q_i of Q an element $t_{f(i)}$ of T. We refer to $(q_i, t_{f(i)})$ as a *matched pair*, and to $((i, f(i)))$ as a *matched index pair*. Notice that, although a matching only requires $N \geq M$, in practice it will be $N \gg M$ (typically $M \leq 10$, while $N = 10^3 \div 10^6$).

In order to search for patterns in the target track that are similar to the query track, we take a cost-based approach, where lower cost implies high similarity.

Definition 1 (Matching Cost). *Given tracks Q and T and a real value τ, the τ-matching cost of a matching f, $C_f(Q, T; \tau)$, is defined as:*

$$C_f(Q, T; \tau) = \sum_{i=1}^{M} \left(t_{f(i)} - q_i - \tau \right)^2 \tag{1}$$

[3] A stand-alone desktop application available at `http://www-db.disi.unibo.it/research/GenData/` and described in [5] enables biologists to define patterns of interest using the Integrated Genome Browser [6].

and the matching cost of f, $C_f(Q,T)$, is the minimum *τ-matching cost of f over all τ values, $C_f(Q,T) = \min_\tau C_f(Q,T;\tau)$.*

Let $\delta_{i,j} = t_j - q_i$ be the *absolute* offset between elements t_j and q_i. The τ parameter in Equation 1 allows us to consider *shifted* $(\delta_{i,j} - \tau)$ rather than absolute $(\delta_{i,j})$ offsets, and is used to translate Q so as to better match elements of T. For any given matching f of Q in T, the matching cost $C_f(Q,T)$ is obtained for $\tau = \sum_{i=1}^{M} \delta_{i,f(i)}/M$, i.e., the matching cost equals M times the variance of absolute offsets $\delta_{i,f(i)}$ of matched elements.

Given tracks Q and T, the best-matching problem (BMP) is to determine the matching f^* with minimum matching cost, i.e., $C_{f^*}(Q,T) \leq C_f(Q,T)\ \forall f$.

Example 1. Let $Q = \langle 1, 7, 10\rangle$ and $T = \langle 3, 5, 9, 11, 13, 14, 18,\ \ 21\rangle$. One of the possible matchings of Q in T is $f = \{((1,1)), ((2,3)), ((3,8))\}$ that assigns elements $(3, 9, 21)$ to corresponding elements of Q. Assuming $\tau = 7$, the τ-matching cost of f is 66, since $C_f(Q,T;7) = (3-1-7)^2 + (9-7-7)^2 + (21-10-7)^2 = 66$. The matching cost of f is obtained for $\tau = ((3-1) + (9-7) + (21-10))/3 = 5$: $C_f(Q,T) = 54$. The solution to the BMP is $f^* = \{((1,2)), ((2,4)), ((3,7))\}$ that matches Q to elements $(5, 11, 14)$ of T, which yields $\tau = 4$. The matching cost of f^* is $C_{f^*}(Q,T) = (5-1-4)^2 + (11-7-4)^2 + (14-10-4)^2 = 0$.

The best-matching problem, which amounts to finding a *minimum-variance matching*, is a specific case of quadratic assignment problem, which is known to be NP-hard [1]. Although this does not immediately lead to conclude that BMP is NP-hard as well, we strongly suspect this is the case, even because variance-minimization problems are reputed difficult to solve [8].

2.1 The Root-element Approach

An alternative cost definition that leads to a tractable version of the problem is to consider a fixed value of τ. In particular, we take $\tau \equiv \tau^r = t_{f(1)} - q_1$, i.e., a zero-cost for the first pair of matched elements of Q and T, which are therefore called *root (reference) elements*. The *root-element matching cost* $C_f^r(Q,T)$ is:

$$C_f^r(Q,T) = \sum_{i=1}^{M} \left(\delta_{i,f(i)} - \tau^r\right)^2 = \sum_{i=1}^{M} \left(\delta_{i,f(i)} - \delta_{1,f(1)}\right)^2 \tag{2}$$

and the root-element BMP (R-BMP) is finding the matching f^* with minimum root-element matching cost. Notice that, unlike BMP, in R-BMP the contribution of a matched pair $(q_i, t_{f(i)})$ to the overall cost is decoupled from that of the other pairs, since τ^r only depends on the root-elements and not on the whole matched elements. This is the key to develop an efficient dynamic programming (DP) algorithm, WDP-RBMP, where the 'W' stands for "windowed".

Lemma 1. *A matching f is optimal only if the (partial) matching $(f(1), f(2), \ldots, f(\ell))$, $\ell = 1, \ldots, M-1$, has minimum cost among all (partial) matchings*

$f' = (f'(1), f'(2), \ldots, f'(\ell))$ *such that* $f'(1) = f(1)$ *and* $f'(\ell) = f(\ell)$*, i.e.,* $\sum_{i=1}^{\ell}(\delta_{i,f(i)} - \delta_{1,f(1)})^2 \leq \sum_{i=1}^{\ell}(\delta_{i,f'(i)} - \delta_{1,f'(1)})^2$*. The condition is sufficient when also the last assignment,* $\ell = M$*, is considered.*

The intuition about the proof is that, for given "start" ($f(1)$) and "end" ($f(\ell)$) positions in T, any partial matching f' which also matches q_1 to $t_{f(1)}$ ($f'(1) = f(1)$) and q_ℓ to $t_{f(\ell)}$ ($f'(\ell) = f(\ell)$), yet has a partial cost higher than that of f cannot be completed to yield a matching with minimum cost.

Based on the above lemma, our WDP-RBMP algorithm starts by partitioning the problem into $(N - M + 1)$ subproblems, one for each possible value of $f(1)$ and, consequently, of τ^r.[4] Given $f(1)$, we apply the DP technique by constructing an $M \times N$ matrix $Z_{f(1)}$. The value of cell (i, j) of this matrix, also called the *cell cost*, $Z_{f(1)}(i, j)$, is computed as the minimum cost obtainable by matching the first $i - 1$ elements of Q in the first $j - 1$ elements of T and q_i with t_j, i.e.:

$$Z_{f(1)}(i, j) = \min_{h:h<j} \left\{ Z_{f(1)}(i - 1, h) \right\} + (\delta_{i,j} - \tau^r)^2 \tag{3}$$

Let h' denote the value of index h yielding the minimum in the above equation. For each cell (i, j) we also maintain a list, $ML_{f(1)}(i, j)$, of the indices of the matched elements of T, which is updated as $ML_{f(1)}(i, j) = ML_{f(1)}(i - 1, h') + (j)$, where '+' denotes list append. The procedure starts by filling a single cell, $(1, f(1))$, in the first row of the matrix, for which it is, by definition, $Z_{f(1)}(1, f(1)) = 0$ and $ML_{f(1)}(1, f(1)) = (f(1))$.

Theorem 1. *When cells of each matrix* $Z_{f(1)}$*,* $1 \leq f(1) \leq (N - M + 1)$*, are filled according to Equation 3, it is* $C^r_{f^*}(Q, T) = \min_{f(1)} \min_j \left\{ Z_{f(1)}(M, j) \right\}$.

Because of the constraint $f(i) < f(i + 1)$, several cells of matrix $Z_{f(1)}$ will remain *unfilled* (their cell cost is assumed to be ∞). Indeed, it is possible to considerably reduce the number of matrix cells to be filled, as follows.

The closest match of q_i in T ($i > 1$) is the element of T, with index $cm(i)$, for which $cost(i, j) \stackrel{\text{def}}{=} (\delta_{i,j} - \tau^r)^2$, that is, the cost of matching q_i and t_j, is minimized.[5] Since both T and Q are strictly increasing sequences, it is $cm(i) \leq cm(i+1)$. In absence of "conflicts" among the closest matches, i.e., when $cm(i) < cm(i + 1)$, the matching $f_{cm} = (f(1), cm(2), \ldots, cm(M))$ would be the best one for the given root element $t_{f(1)}$. However, this breaks down in case of conflicts ($cm(i) = cm(i + 1)$), since f_{cm} would *not* be a matching anymore.

The intuition behind the WDP-RBMP algorithm is that the optimal matching has to be "close" to f_{cm}, thus for each row i of $Z_{f(1)}$ only a *window* of cells of limited size around $cm(i)$ has to be considered.

Theorem 2. *In order to find the best possible matching* f^**, for each matrix* $Z_{f(1)}$ *the only cells to be filled are those in the* DP-window *defined as* $W =$

[4] It has to be $f(1) \leq N - M + 1$ in order to respect the constraint $f(i) < f(i + 1)$.

[5] We omit here the very particular case when the closest match is tied among two adjacent elements in T, which would unnecessarily lengthen the description.

$\{W_i : i > 1\}$, *where* $W_i = \{(i, j) : j \in [L_i, H_i]\}$, *and*

$$L_M = cm(M) \quad L_i = \min\{cm(i), L_{i+1} - 1\}\ (i = 2, \ldots, M-1)$$
$$H_2 = cm(2) \quad H_i = \max\{cm(i), H_{i-1} + 1\}\ (i = 3, \ldots, M)$$

Notice that, it is $L_i < L_{i+1}$ and $H_i < H_{i+1}$. Figure 2 shows a sample DP-window for the case $M = 7$ and the worst-case scenario in which all the $M - 1$ closest matches, $cm(i)$, are in conflict.

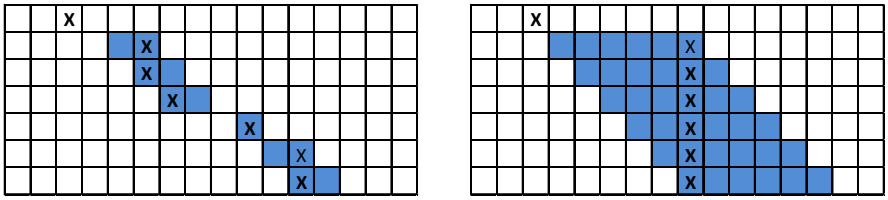

Fig. 2. Examples of DP-windows: closest matches are denoted with a **X**: A possible DP-window for the case $M = 7$ (left) and the worst-case scenario (right).

The complexity of the WDP-RBMP algorithm is $\mathcal{O}(MN(\log N + M))$, which is $\mathcal{O}(N \log N)$ when $M = o(N)$: for each of the $(N - M + 1)$ matrices we execute $M - 1$ binary searches to find the closest matches and then fill, in the worst case, $M - 1$ cells on each row. The cost of filling each cell is $\mathcal{O}(1)$, since for determining h', i.e., the value of index h which yields the minimum in Equation 3, it is sufficient to keep track of the minimum cell cost when filling the cells in a row.[6]

Example 2. Let $Q = \langle 8, 20, 22, 36\rangle$ and $T = \langle 10, 15, 17, 27, 35, 39, 45, 50, 62, 70\rangle$, and consider Z_2 (see Figure 3), one of the $(N - M + 1) = 7$ matrices generated by WDP-RBMP, for which it is $\tau^r = 15 - 8 = 7$. It is $cm(2) = 4$, since the closest match for $q_2 + \tau^r = 20 + 7$ is $t_4 = 27$. The same value is obtained for $cm(3)$, whereas $q_4 + \tau^r = 36 + 7 = 43$ yields $cm(4) = 7$. Based on Theorem 2 it is $W_2 = [3, 4]$, $W_3 = [4, 5]$, and $W_4 = [7, 7]$. To see how cells are filled consider cell $(3, 5)$, for which Equation 3 yields $Z_2(3, 5) = \min_{h:h<5}\{Z_2(2, h)\} + (\delta_{3,5} - \tau^r)^2 = 0 + ((35-22)-7)^2 = 36$. It is $h' = 4$, thus $ML_2(3, 5) = ML_2(2, 4) + (5) = (2, 4, 5)$. The best matching f for Z_2 is obtained in cell $(4, 7)$, that is, $f = ((\!(1, 2)\!), (\!(2, 4)\!), (\!(3, 5)\!), (\!(4, 7)\!))$ that assigns $(15, 27, 35, 45)$ to $(8, 20, 22, 36)$, with cost 40.

Z_2	**1**(10)	**2**(15)	**3**(17)	**4**(27)	**5**(35)	**6**(39)	**7**(45)	**8**(50)	**9**(62)	**10**(70)
1 (8)		0 (2)								
2 (20)			100 (2, 3)	**0** **(2, 4)**						
3 (22)				**104** **(2, 3, 4)**	36 (2, 4, 5)					
4 (36)							**40** **(2, 4, 5, 7)**			

Fig. 3. WDP-RBMP algorithm: the matrix Z_2 of Example 2. Closest matches, $cm(i)$, are in boldface.

[6] Here we apply the identity $\min\{a_1, \ldots, a_{n-1}, a_n\} = \min\{\min\{a_1, \ldots, a_{n-1}\}, a_n\}$.

3 Extending the Base Model

In this section we provide some intuition on how the base model can be extended so as to consider the following aspects:

Interval regions: In actual genomic applications elements t_i and q_j are intervals rather than points. Our solution is to reduce each interval to its *centroid*, which implies no changes to the WDP-RBMP algorithm. If regions' lengths are deemed to be relevant for the specific problem at hand, they can be modeled as region features (see below).

Multi-track patterns: When a query pattern is defined on NT different tracks, i.e., $\mathcal{Q} = (Q^1, \ldots, Q^{NT})$, where $Q^x = \langle q_1^x, \ldots, q_{M_x}^x \rangle$, each pattern track Q^x is searched in a different target track $T^x = \langle t_1^x, \ldots, t_{N_x}^x \rangle$. In order to give the same importance to all the tracks, we use as root-elements the couple $(q_1^y, t_{f^y(1)}^y)$ that introduces the minimal overall cost (defined as the sum of matching costs for each pattern track). WDP-RBMP can be extended by using NT matrices $Z_{f^y(1)}^x$, $x = 1, \ldots, NT$, when the root-element $t_{f^y(1)}^y$ is chosen from T^y. For each of the $N_y - M_y + 1$ possible values of the root-element, we fill matrix $Z_{f^y(1)}^y$ as in the single-track case, whereas for each other matrix $Z_{f^y(1)}^x$, $x \neq y$, we also fill the 1st row, since there is no root-element defined for T^x.

Negative matchings: Negative matching tracks are target tracks in which there must be no regions in the area of a result, thus they are used to limit the space of the solutions. The matrix columns of each element of a target track corresponding to regions in a negative track can therefore be dropped before starting the search process (such cells are given cost = ∞).

Partial matchings: The rationale behind partial matchings is that, in some cases, requiring to match all pattern elements might lead to a poor solution. In such cases, it might be preferable to "skip" one or more pattern elements, assigning them a cost $c(\perp)$. In order to apply WDP-RBMP to the partial matching case, a matrix $ZP_{f(1)}^x$ for each partial matching track TP^x is needed. The cost of cell (i, j) of $ZP_{f(1)}^x$ must now consider also the null ($\perp$) case, and matrix $ZP_{f(1)}^x$ needs to be extended with an additional "0" column, corresponding to an unmatched first element.

Region features: Region features are used to determine the *region distance* of each couple of matched regions, which becomes part of the cost function. Since the region distance is not monotone, it is not possible to limit the search only to a neighborhood of the closest matches of q_i elements.

Top-k queries: The Top-K version of R-BMP aims to discover the K matchings $F = \{f_1, \ldots, f_K\}$ with the smallest overall cost. We further require that resulting patterns have no regions in common, so as to increase the diversity of the result. This is obtained by comparing results produced by all the matrices and keeping the best K disjoint results; note that each matrix can produce at most one result, as all the matchings associated with a matrix share the same root element.

4 Experiments

We applied the WDP-RBMP algorithm to solve the relevant biological problem of *finding enhancer regions*. This involves the search for DNA regions outside of the genes, at a certain distance from a gene's transcription start site (TSS), and associated with the presence or absence of specific regions, where given molecules bound to the DNA, that can be measured through NGS ChIP-seq experiments.

In particular, biologists believe that an *active putative enhancer* (APE) region should be not closer than 20K bases to the closest TSS, and have presence of overlapping peaks of the H3K4me1 and H3K27ac signals, and absence of H3K4me3 signal peaks. Furthermore, an APE could optionally include overlapping peaks of the DHS, CTCF, P300, and/or Pol2 signals. Thus, the search for APE regions can be expressed as a multi-track, interval region matching problem where peak regions of the H3K4me1 and H3K27ac signals constitute two positive matching tracks, TSS regions and peak regions of the H3K4me3 signal constitute two negative matching tracks, and peak regions of the DHS, CTCF, P300 and Pol2 signals constitute four partial matching tracks, respectively.

For TSS regions we used public data from SwitchGear Genomics, provided by the UCSC annotation database. For all other signals, we considered ChIP-seq experiments on specimens of the K562 cell line (Chronic Myeloid Leukemia), which are publicly available in the ENCODE project repository; thus, we extracted all H3K4me1, H3K27ac, H3K4me3, DHS, CTCF, P300 and Pol2 samples of ChIP-seq peaks of the K562 cell line, we merged sample replicates, and created a single dataset with a single sample (track) for each signal, all samples with the same region features. The number of regions in each track is listed in Table 1.

The Top-100 results found by WDP-RBMP were visually inspected by an expert, who evaluated all of them correct. Automatic evaluation of all the 1,651 results found by WDP-RBMP (in less than one minute) is difficult, since there is neither consolidated knowledge of enhancers, nor a consensus on the computational method specifically designed for their discovery. Therefore, we used for comparison a different set of data, about *chromatin*[7] *state segmentation*, generated by the Broad Institute for a few cell lines, including K562, and made publicly available also in the ENCODE repository. These data, denoted as ENCODE HMM, describe a set of chromatin states of the genome using a Hidden Markov Model (HMM); in particular, each 200 base pair (i.e., nucleotide) interval is assigned to its most likely state under the model; one of such state is associated with enhancers.[8] Although ENCODE HMM data were obtained considering a different set of epigenomic signals (with only 3 out of 9 signals overlapping with the signals considered by us), we found a very good matching between the regions denoted as enhancers by ENCODE HMM and the regions determined by WDP-RBMP, as shown in Table 1. This evaluation demonstrates both the correctness of the pattern-matching method and its ability to find interesting regions.

[7] Chromatin is a complex of molecules consisting of DNA, proteins, and RNA.
[8] Detailed information about the method and model parameters can be found in [3].

Table 1. Considered tracks and number of genomic regions, left. Precision of WDP-RBMP wrt. ENCODE HMM dataset (EH), right.

sample	regions
TSS	131,780
H3K4me1	116,503
H3K27ac	45,796
H3K4me3	142,738
DHS	360,648
CTCF	318,982
P300	69,370
Pol2	177,900

top-K results	EH overlapping results	precision
10	9	90.00%
50	45	90.00%
100	86	86.00%
250	215	86.00%
500	415	83.00%
1,000	837	83.70%
(all) 1,651	1,411	85.46%

5 Conclusions

We presented an efficient method to find patterns in genomic region sequences; it has practical applications in revealing interesting and unknown regions of the genome, and therefore it is an important ingredient in supporting biological research. In our future work, we plan to use the method for biological research, in strong connection with biologists of the GenData 2020 project, by using experimental data produced at IEO or by connecting to public data sources.

References

1. R. E. Burkard et al. Assignment problems. SIAM, 2009.
2. ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57-74, 2012.
3. J. Ernst et al. Mapping and analysis of chromatin state dynamics in nine human cell types. *Nature*, 473(7345):43-49, 2011.
4. M. Masseroli et al. GenoMetric Query Language: A novel approach to large-scale genomic data management. *Bioinformatics*, 31(12):1881-1888, 2015.
5. P. Montanari et al. An IGB-based application for the search of patterns in a genomic sequence. *GenData 2020 Tech. Rep.*, May 2015.
6. J. W. Nicol et al. The Integrated Genome Browser: free software for distribution and exploration of genome-scale datasets. *Bioinformatics*, 25(20):2730-2731, 2009.
7. Paradigm4, Inc. Accelerating bioinformatics research with new software for Big Data to Knowledge (BD2K). Waltham, MA, 2015 (available at: `http://www.paradigm4.com/`).
8. M. Sniedovich. Solution strategies for variance minimization problems. *Computers & Mathematics with Appl.*, 21(2-3):49-56, 1991.
9. J. N. Weinstein et al. The cancer genome atlas pan-cancer analysis project. *Nat. Genet.*, 45(10):1113-1120, 2013.