Application of the Burrows-Wheeler Transform for Searching for Approximate Tandem Repeats

Agnieszka Danek[1], Rafał Pokrzywa[1],
Izabela Makałowska[2], and Andrzej Polański[1]

[1] Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
agnieszka.danek@polsl.pl
[2] Laboratory of Bioinformatics, Faculty of Biology, Adam Mickiewicz University,
Umultowska 89, 61-614 Poznań, Poland

Abstract. Tandem repeats (TRs) are contiguous copies of repeating patterns, which may be either exact or approximate. Approximate tandem repeats (ATRs) in a genomic sequences are adjacent copies of a repeating pattern of nucleotides, where similarity is defined by a suitable measure. Both TRs and ATRs are used in forensic analysis, DNA mapping, testing for inherited diseases and many evolutionary studies. All their functions and roles are not well defined and remains a subject of ongoing investigation. However, growing biological databases together with tools to look for such repeats may lead to better understanding of their behavior. This paper presents our method for searching for ATRs defined on the basis of the model of substitution mutations and its comparison to two other tools. The capabilities and limitations of methods are analyzed and results obtained with each tool are investigated.

Keywords: approximate tandem repeats, Burrows-Wheeler transform, suffix array, Hamming distance.

1 Introduction

Tandem repeats (TRs) are consecutive, repeating patterns in genomic sequences. TRs belong to the most important loci in genomes due to their abundance in DNA sequences and to their role both in evolution and in molecular mechanism of functioning of organisms. Evolution of tandem repeats loci is governed by a mechanism called slippage mutation, e.g. [1], which due to its high intensity belongs to the major factors of genomic dynamics. A very important issue is the dynamics of interaction between slippage and point mutation, which is still an area of an intensive research [2], [3], [4]. As for functional roles of TRs in cellular mechanisms there is a lot of evidence proving linkage of TRs to important molecular processes in cells. TRs play important roles in the gene expression and transcription regulations [5]. They are also widely used as markers for DNA mapping and DNA fingerprinting [7]. It is well known that when TRs are occurring in increased, abnormal number, they cause a series of inherited diseases [6] (i.e. trinucleotide repeat disorders).

Critically important element in the research on TRs is development of tools for their efficient and accurate identification. Locations of TRs in genomes can be detected by using appropriately designed experimental techniques [8] based on DNA amplification methodologies. However, in the era of very high power of direct sequencing technology, methodologies of discovering TRs by using text mining techniques are coming to the first place. These methodologies allow for efficient and massive detection of both patterns and locations of TRs in genomic sequences. In the aspect of discovery by using text mining, TRs should be divided into two groups, exact tandem repeats (ETRs) and approximate tandem repeats (ATRs). For both groups of TRs many detection methods were published, some are overviewed in the recent survey papers [9] [10] [11] [12]. Detection tools based on text mining can be divided into two classes. The first class includes algorithms suitable for searching for ATRs based on introducing mathematical models or transformations to represent and measure repeatability of DNA sequences. These algorithms use models and methods such as autocorrelation distance between sequences, transforming nucleotide symbols to numerical values and then using frequency domain analyses, HMM models [13] [14]. The second class of methods involves combinatorial text searches through genomic databases [15] [16] [17] [18]. As result of intensive studies on TRs and developing methods of their efficient identification several growing-in-size biological databases of TRs have been developed [19] [20] [21]. These databases support many researches in molecular biology and evolution.

As observed in comparisons presented in the literature [9] [12], detection methods for TRs still need development and refinement due to their limitations and differences seen between various approaches. This observation is particularly important for ATRs, due to the fact that different approaches not only differ in algorithmic aspects but also often use different measures of similarity between motifs, which makes the results more difficult to compare. In this paper we present a new method for combinatorial, exhaustive detection of approximate tandem repeats based on application of the Burrows-Wheeler Transform, called BWatrs [15], [22]. We also show a study devoted to comparisons of different algorithms for discovery of ATRs. We compare three tools for discovery of ATRs, mreps published in [17], Tandem Repeat Finder published in [23] and our tool BWatrs. We compare tools for searching for ATRs by using quantitative performance measures suitable for evaluating combinatorial text search engines, number the detected pattern stratified with respect to motif length. Our research is based on developing an algorithm dedicated to analysis and comparison of lists of results returned by different ATRs search tools. To the best of our knowledge this study is the first one devoted exclusively to comparisons of combinatorial text mining algorithms for discovery of ATRs and presenting results of extensive browsing of returned lists of ATRs, including one-to-one identities of detected motifs and numbers of unique motives specific to each tool. The difficulty of task addressed in our study stems from inconsistent definitions of ATR among different papers and varying approaches to look for ATRs. In order to pursue scheduled research we had to set standards in parameter choices for different tools.

In the following sections, three evaluated tools are presented with the focus on our program. Next the methodology of comparison and performed experiment are described. Finally, the results of the comparisons are given together with some conclusions.

2 Evaluated Tools

In our research we evaluated three different tools designed to look for the approximate tandem repeats (ATRs) in the genomic sequences. In the succeeding subsections each of them is presented. The approaches are briefly described along with all parameters that can be tuned. Additionally, a more detailed explanation of our algorithm is provided.

2.1 Burrows-Wheeler Approximate Tandem Repeats Searcher

Burrows-Wheeler Approximate Tandem Repeats Searcher (BWatrs) is our approach to find ATRs. Its prior version was presented in [22]. It is a development of a method for searching for exact tandem repeats described in [24], [15]. To be aware of kind of ATR looked for by our method, it is necessary to present some related terminology. First, we define a measure of dissimilarity between two strings and the successive definitions help to understand a type of ATRs searched.

Definition 1. Given two strings A and B of equal length, $h(A, B)$ is a Hamming distance between A and B, that is a minimal number of substitution needed to be done in string A to transform it to string B.

Definition 2. A *K-mismatch double ATR with period p* is a string consisting of two consecutive strings S_1 and S_2, both of length p, that $h(S_1, S_2) \leq K$.

Definition 3. A *K-mismatch ATR with period p* is any string of length $n \geq 2p$ for which every substring of length $2p$ is a *K-mismatch double ATR with period p*. The ratio n/p is called an exponent.

Definition 4. A *maximal K-mismatch ATR with period p* is a string which is a *K-mismatch ATR with period p* and cannot be further extended to the left or to the right to still meet the definition of the *K-mismatch ATR with period p*.

Assuming we are given a string S over the alphabet Σ, range of acceptable periods $< p_1, p_2 >$ and maximal number of errors K, we are interested in finding all *maximal K-mismatch approximate tandem repeats with period p*, where $p \in < p_1, p_2 >$, within the string S. In our future consideration when we refer to ATR searched by the BWatrs we mean this kind of repeat.

Parameters. The BWatrs takes as an input a sequence S and several parameters determining the type of ATRs that are searched: minimum and maximum period (*minPeriod* and *maxPeriod*), minimum exponent *minExp*, minimum total length *minTotal*, maximum number of errors K and maximum percentage of errors *Kprc* between adjacent repeats, minimum total score *minScore* and a flag enabling/disabling marking *mark*.

The meaning of the *minPeriod*, *maxPeriod* and *K* parameters is directly connected with the Def. 4 of the searched ATR. The *minExp* and *minTotal* are straightforward and represent minimum acceptable exponent and total length of the ATR. The *Kprc* gives the possibility to define what percentage of the consecutive repeats (each *K-mismatch double ATR*) can be mismatched. In case of every period, the more restrictive of *K* and *Kprc* is taken into account. The *minScore* parameter defines the minimum total score (*totalScore*) of the ATR, which is calculated according to idea presented in [17], as:

$$totalScore = 1 - \frac{\#errors}{totalLength - period} \tag{1}$$

where $\#errors$ is a sum of all errors (mismatches between the successive motifs) in the ATR. If after an error, the nucleotide return to its previous state, it is count as only one error. The *totalScore* was introduced to control the level of similarity between the whole ATR (not only between the adjacent motifs) and to determine the best period of the found ATR. Finally, the *mark* flag determine whether regions with ATRs already found should be marked to exclude them from future analysis.

Algorithm. At the beginning the input string *S* over the alphabet *Σ* is converted according to the Burrows-Wheeler transform (BWT) [25]. A special character # is appended to *S*, to indicate the end of the string. Then shift rotations of *S*# are made to obtain all suffixes of the input string. Next, all rotations are sorted alphabetically. Last column of such an array of suffixes is a BWT of *S*.

To find ATRs, three auxiliary arrays of length |*S*#| are also constructed. The mapping array *Map* determines at which position in the first column the character is located and, simultaneously, which character in the BWT precedes the current character. The *Pos* array determines the original positions of the suffixes in *S*. The *Prm* array, is an inverse of the *Pos* array (*Prm(Pos(i))* = *i*).

First step of the presented algorithm is finding candidates for *K-mismatch double ATRs*. Converted input string, together with the auxiliary arrays, allows to make use of the alphabetically sorted array of input string suffixes, without the need of storing the whole suffix array structure. The *Map* array is used together with an auxiliary array *C* of length |*Σ*| and the function *occ* to determine the number of occurrences of a certain pattern in the input string *S* according to the algorithm presented by Ferragina and Manzini [26]. The value *C* [*ch*] is the total number of occurrences of all characters preceding character *ch* in the BWT string. The function $occ\,(ch, 1, x)$ reports the number of occurrences of character *ch* in the BWT string from 1 to x in a constant time. The procedure starts with an empty pattern *P*, *startPos = 0*, *endPos* = |*S*| and recursively appends each character *ch* from the considered alphabet in front of *P*. This approach uses the results from the previous iteration to calculate a range of positions for a longer pattern (*ch* + *P*): *newStartPos = C[ch] + occ(ch, 1, prevStartPos)*, *newEndPos = C[ch] + occ(ch, 1, prevEndPos)*. If there is only one occurrence of the current pattern, the recursion is stopped. This way it is possible to go recursively through all groups of repeats found.

Observation. *Two strings of length p with the Hamming distance k between them have always a common, matching substring of length d at corresponding positions, such that:*

$$d \geq \left\lfloor \frac{p}{k+1} \right\rfloor \tag{2}$$

The above observation, previously made in a similar form by Kurtz et al. [18], allows to determine how far away from each other a certain pair of repeats should be positioned to be considered a candidate for a *K-mismatch double ATR*. Therefore, a group of repeats of length d will be used to find candidates for *K-mismatch double ATR* only for periods p satisfying the equation (2) for all $k \leq K$ (or some $K' < K$, if K is restricted by *Kprc*) and conforming to input parameters *minPeriod* and *maxPeriod*. A pair of repeats from the considered group of repeats, with indexes i and j in the suffix array structure, can be a part of a *k-mismatch double ATR* with calculated period p if a difference between their positions in the input string is equal to p, that is $Pos(i) - Pos(j) = p$. To find all such pairs of repeats it is enough to check for each repeat with index i from the group, if p positions to left another repeat from the same group of repeats exists, that is, if $Prm(Pos(i) - p)$ is in the range of indexes of the current group of repeats. If so, the pair of repeats of length d is reported as a candidate for a *double k-mismatch tandem repeat* with period p and number of mismatches k.

The next step is validating the reported candidates. It is checked if a pair of repeats is indeed a part of one (or more) *k-mismatch double ATR with period p.* If the index of the left repeat is j, a *k-mismatch double ATR* of length $2p$ can begin anywhere in the input string between positions *Pos(j) - (p-d)* and *Pos(j).* The Hamming distance is calculated between consecutive strings of length p beginning at all possible positions. If for any of these positions it is equal exactly to k and the $totalScore$ of the repeat is acceptable, the *k-mismatch double ATR* is reported at this position. Otherwise, the candidate is rejected.

In the following stage, the *k-mismatch double ATR* found is extended to the left and to the right to obtain a *maximal K-mismatch ATR.* It is done one character at a time, by checking if a string of the length $2p$ that begins one character to the left (or right) from the current *double ATR* is a *K-mismatch double ATR.* For each found *K-mismatch maximal ATR with period* p, the $totalScore$ is calculated for all periods from 1 to p and the final period is changed to the one that corresponds to the maximum $totalScore$. This way every repeat is reported with the best fitting period. Also, all erroneous edges are cut off from the final ATR. Lastly, it is checked if the found repeat fulfills all the conditions determined by the input parameters. If yes, it is eventually accepted.

Algorithm described will find the same *K-mismatch maximal ATR* many times, extending different *k-mismatch double ATRs* found within that maximal repeat. Thus, all ATRs found, contained entirely in other repeats (or being the same repeat) are filtered out. To meaningfully decrease the amount of computations performed, the marking can be enabled with the *mark* parameter. An additional array of bits of length $|S|$ is used. Initially all bits are set to 0. After an ATR is found, all bits corresponding to its complete position are set to 1. This array of bits is a simple way to mark regions with found ATRs. If a candidate is entirely placed within a region already occupied by other, previously found, repeat, it is immediately rejected. When marking is on the results can be slightly different, because of change in order of searching for ATRs. Nevertheless, all regions with ATRs will always be reported, while the reduction in the amount of computation is significant.

2.2 Mreps

Mreps is a software for identifying tandem repeats in DNA sequences, presented in [17] and available on-line [27]. It exhaustivity looks for all tandem repeats with substitutions that satisfy some assumed criteria (these repeats meet the Definition 4). Then, the repeats go through a heuristic treatment in order to obtain more biologically relevant repetitions.

Parameters. Mreps takes as an input a sequence S and several parameters defining ATRs that are searched: minimum period and maximum period (*minPeriod* and *maxPeriod*), minimum exponent *minExp*, minimum total length *minSize*, maximum total length *maxSize*, resolution *res*, positions in the string S to start and end search for ATRs (*from* and *to*) and a flag enabling outputting small repeats *allowsmall*.

Algorithm. In the exhaustive search, the algorithm looks for the longest common extensions with *res* mismatches at each position of the input string S. These are used to find *res-mismatches double ATRs*. Then found double repeats are joined to obtain *res-mismatches maximal ATRs*. In the following heuristic approach, the period of each found repeat is changed according to the internal total score of that repeat and basing on a statistical analysis made by the authors on some artificial genomic sequences.

The main drawback of *mreps* is its inability to handle the N-regions of the input sequence. The program randomly changes every N to one of the characters it can process: A, C, T or G. Additionally, If there are too many N characters, the program does not operate at all, outputting an error. As a consequence, *mreps* can report an artificial repeats in regions where originally only N characters were present. Thus it is difficult to apply *mreps* to look for tandem repeats in e.g. human genome, as it contains large number of regions of N characters.

2.3 Tandem Repeat Finder

The Tandem Repeat Finder (TRF) it is a widely used statistically based method for searching for ATRs that can contain mismatches and indels, described in [23] and available on-line [28]. It collects exact matches as seeds and then processes and extends them, to find ATRs that satisfy the assumed statistical criteria.

Parameters. The TRF takes as an input a sequence *S* and parameters defining ATRs that are searched: alignment weights for match, mismatch and indels for Smith-Waterman style local alignment using wraparound dynamic programming (*match*, *mismatch* and *delta*), matching and indel probability (*PM* and *PI*), minimum alignment score to report the repeat *minScore* and maximum period size *maxPeriod.*

Algorithm. The algorithm has two main phases: detection and analysis. Alignment of two copies of a pattern of length n is modeled as n independent Bernoulli trials (coin-tosses), where head is interpreted as a match between aligned nucleotides and tail is a mismatch, insertion or deletion. During the detection component, for some small integer *k*, all possible k-length strings are listed and the input sequence is scanned to look for all positions of them. Next, the distance lists are created, collecting all pairs of matching strings placed at the same distance from each other. Then, the statistical criteria based on four distributions (depending on period, *PM*, *PI* and *k*) are applied, to find candidate tandem repeats. In the analysis component, found candidates are aligned with the surrounding sequence using wraparound dynamic programming. If at least two copies of a pattern are aligned and satisfy the *minScore* parameter, the ATR is reported.

The main disadvantage is that not all ATRs will be find. If consecutive copies of the repeat do not contain a series of *k* matching nucleotides at corresponding positions, such ATR will not be discovered. Also, the input parameters do not give much freedom in specifying kind of repeats searched.

3 Methodology

In this section the methodology used to compare results obtained with three evaluated tools is described. We present our program, the ATR-compare, which takes as an input two lists of ATRs and makes a summary and comparison between them. Next, the experiment performed is specified.

3.1 ATR-Compare

The ATR-compare is a program that we designed to make a comparison of the outputs of the evaluated tools for searching for ATRs. It takes as an input two sets of ATRs, each being a text file formatted in such a way, that every line represents one tandem repeat. Each repeat/line is composed of five features of the repeat (start position, end position, period, exponent, sequence), separated by a whitespace. The ATR-compare perform an extensive browsing through the two input lists of ATRs, to give a detailed comparison of them.

The output consists of number of text files. First is a quantitive comparison of the two input sets of ATRs, with division to different periods. Next, subsets of the input sets, satisfying a certain criteria, are reported in separate text files. Each such subset has a corresponding text file with statistics showing number of ATRs in the subset, classified according to period. The subsets are: set of ATRs identical in both input sets, set of ATRs almost identical in both input sets, differing only by period, set of ATRs from the first list contained entirely in one of ATR of the second list (but not equal in length) and vice versa, set of overlapping ATRs from both input lists, not belonging to any of the previous sets and lastly, two sets of ATRs unique to each of the input sets.

3.2 Experiment

The aim of the experiment was to compare ATRs that can be found with the evaluated tools. For that purpose all three programs were run for the same genomic sequence: Homo sapiens chromosome 22, GRCh37.p5 Primary Assembly (Accession.Version: NC_000022.10). In case of the mreps tool, because of its limitations, it was necessary to first find large N-regions of the input sequence and exclude them from the search. The *from* and *to* parameters were used to run the mreps for all other regions and then the obtained results were joined.

The input parameters of each tool were chosen to define searched ATRs in as similar way to other tools as it is possible. The parameters selected for BWatrs were: $minPeriod = 1$, $maxPeriod = 100$, $minExp = 3$, $minTotal = 4$, $K = 7$, $Kprc = 35$, $minScore = 75$, $mark = 1$. The parameters chosen for mreps were: $minPeriod = 1$, $maxPeriod = 100$, $minExp = 3$, $minSize = 4$, $res = 7$ and *allowsmall* enabled. For the TRF set of standard parameters was chosen: $match = 2$, $mismatch = 3$, $delta = 5$, $PM = 80$, $PI = 10$, $minscore = 14$, $maxperiod = 100$ and obtained results were filtered to contain only repeats with period larger or equal to 1, exponent larger or equal to 3 and total size larger or equal to 4. The output of each program was converted to satisfy the convention for input sets used by the ATR-compare. Then the ATR-compare was run for each pair of results sets and to set of ATRs found by one tool and the set of appropriate results of comparison of the two other tools.

4 Results

The experiment described in the previous section reveal that BWatrs found the highest total amount of ATRs: 1281009, while mreps and TRF found 1018136 and 158636 ATRs, respectively. In the Fig. 1 there is a quantitive comparison of the number of ATRs found by different tools stratified with respect to period. It can be observed that application of both tools, BWatrs and mreps has led to detection of much more ATRs with small periods than application of TRF and, conversely, to detection of less ATRs than TRF, for larger periods. The latter observation is caused by the fact that for TRs with larger period the restriction of allowable number of mismatches to 7, caused by assuming values of the parameters ($K = 7$ for BWatrs and $res = 7$ for mreps) can be violated when TRF is applied.

Further results of application of our program ATR-compare demonstrate that BWatrs has found 619250 ATRs identical to mreps and 38890 repeats identical

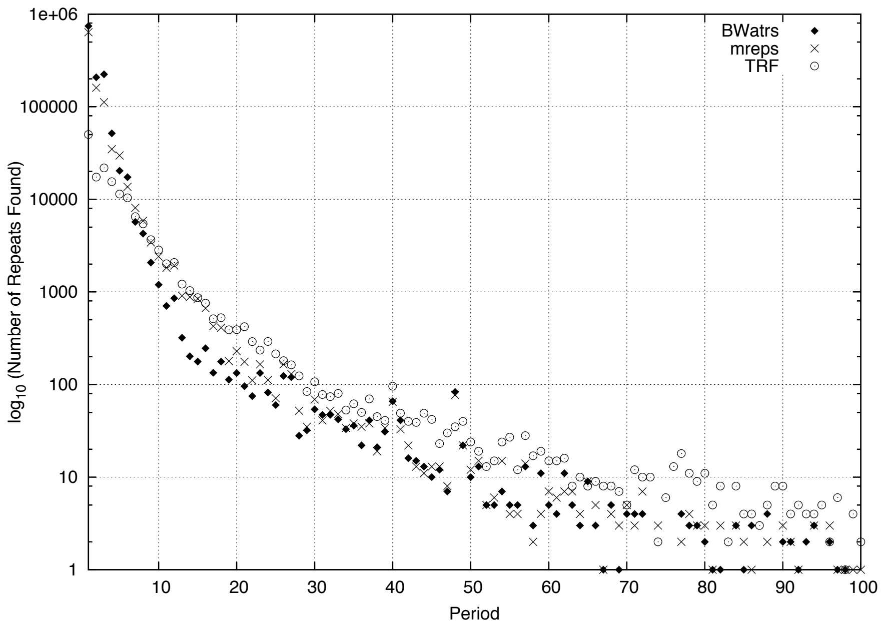

Fig. 1. Number of all ATRs (with distinction to different periods) found by each of the evaluated tools

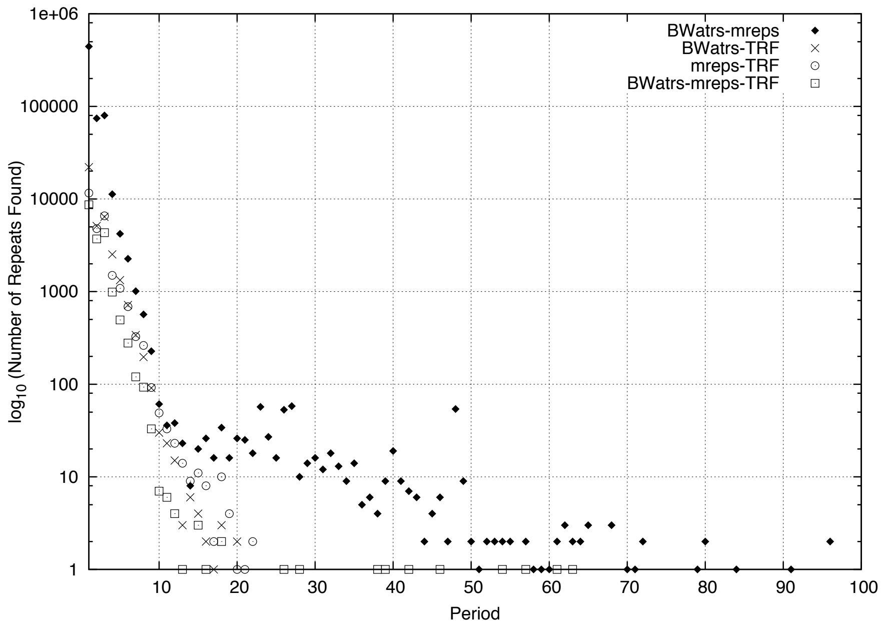

Fig. 2. Number of identical ATRs (with distinction to different periods) found by pairs of the evaluated tools and by all three tools together

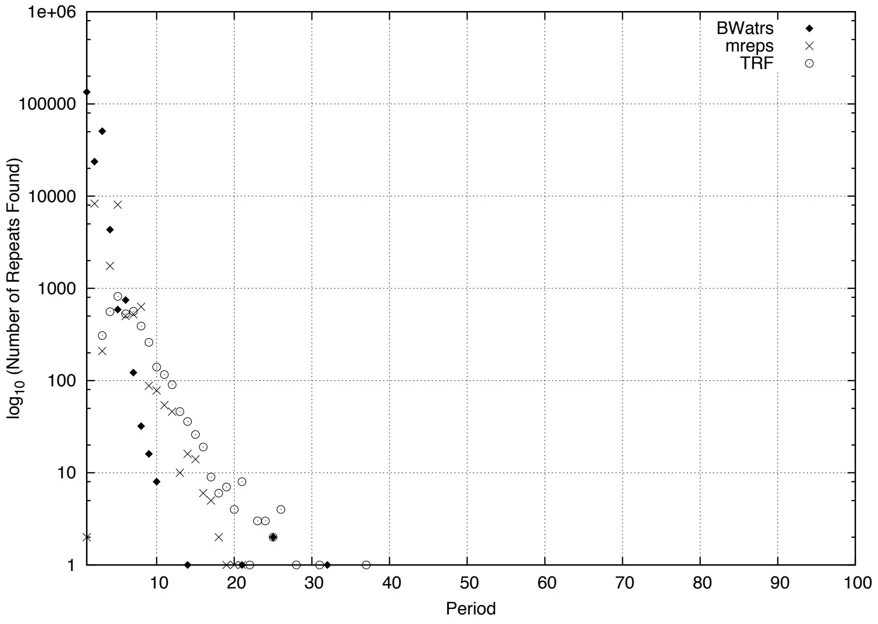

Fig. 3. Number of unique ATRs (with distinction to different periods) found by each of the evaluated tools

to TRF. Also, 27103 identical repeats were found between mreps and TRF. Lastly, 18759 exactly the same repeats were found by all evaluated tools. In the Fig. 2, the amount of found ATRs identical to each pair of tools and to all three tools is presented, with distinction with respect to different periods. Outcomes of BWatrs and mreps are quite similar one to another, but still there are many ATRs that differ.

In the Fig. 3, unique ATRs found by each tool are shown (also classified according to the period). BWatrs located the largest number of unique repeats: 214708, while mreps and TRF discovered 20311 and 3951 of such repeats, respectively. The subsets representing unique ATRs of BWatrs consist mainly of short repeats with small period, but there are also some, potentially interesting, longer repeats.

5 Conclusions and Future Work

It should be noted that both algorithms mreps and TRF are already frequently used in many genomic studies and are considered by their users as reliable tools for ATRs detection. Our study confirms efficiencies of existing tools for detection of ATRs. Nevertheless, application of two different tools mreps and TRF for exemplary chromosomal data and comparison to our new tool BWatrs show considerable variations between outcomes of different algorithms. Results of comparisons are presented in Figures 1, 2 and 3. Differences seen in figures 1, 2 and

3 stem from different constructions, different parameters, different definitions of repeatability and different search strategies between different algorithms. Some effects of using varying approaches for ATRs of different lengths were explained in the previous section. Our tool BWatrs is most sensitive for short ATRs. It also leads to detection of largest numbers of unique ATRs, as seen in figure 3. These properties can be advantageous in some applications, like inter species comparisons, homology detection between genomic sequences or more specialized, DNA assembly quality control.

As a conclusion, using effective text searching algorithms based on Burrows-Wheeler transform of entire chromosomal sequences allows us to obtain a new tool competitive to existing methodologies.

Our further research will involve further comparisons of results of ATR searchers focusing on biological significance of the ATRs found by different tools. One of the main features of the tandem repeat is its high probability to change in number of copies. It is caused by its structure that increases the likelihood of slippage mutation to occur [1], [4], [29]. Thus, as a meaningful ATR, we can understand a tandem repeat that differs in number of copies between two individuals. We plan to look for such repeats in the reference and the alternative human genomes, taking into account the subsets generated by the ATR-compare tool.

Acknowledgments. This work was supported by the European Union from the European Social Fund (grant agreement number: UDA-POKL.04.01.01-00-106/09) A.D., R.P. and by Polish National Science Center under project DEC-2011/01/B/ST6/06868 I.M, A.P.

References

1. Chakraborty, R., Kimmel, M., Stivers, D.N., Davison, L.J., Deka, R.: Relative mutation rates at di-, tri-, and tetranucleotide microsatellite loci. PNAS 94, 1041–1046 (1997)
2. Kruglyak, S., Durrett, R.T., Schug, M.D., Aquadro, C.F.: Equilibrium distributions of microsatellite repeat length resulting from a balance between slippage events and point mutations. PNAS 95, 10774–10778 (1998)
3. Pumpernik, D., Oblak, B., Borštnik, B.: Replication slippage versus point mutation rates in short tandem repeats of the human genome, Mol. Genet. Genomics 279(1), 53–61 (2008)
4. Leclercq, S., Rivals, E., Jarne, P.: DNA slippage occurs at microsatellite loci without minimal threshold length in humans: a comparative genomic approach. Genome Biol. Evol. 2, 325–335 (2010)
5. Vinces, M.D., Legendre, M., Caldara, M., Hagihara, M., Verstrepen, K.J.: Unstable Tandem Repeats in Promoters Confer Transcriptional Evolvability. Science 324, 1213 (2009)
6. McMurray, C.T.: Mechanisms of trinucleotide repeat instability during human development. Nat. Rev. Genet. 11(11), 786–799 (2010)
7. Jeffreys, A.J., Wilson, V., Thein, S.L.: Individual-specific 'fingerprints' of human DNA. Nature 316, 76–79 (1985)

8. Weber, J.L., Wong, C.: Mutation of human short tandem repeats. Hum. Mol. Genet. 2, 1123–1128 (1993)
9. Merkel, A., Gemmell, N.: Detecting short tandem repeats from genome data: opening the software black box. Brief. Bioinform. 9(5), 355–366 (2008)
10. Saha, S., Bridges, S., Magbanua, Z.V., Peterson, D.G.: Empirical comparison of ab initio repeat finding programs. Nucleic Acids Res. 36(7), 2284–2294 (2008)
11. Lerat, E.: Identifying repeats and transposable elements in sequenced genomes: how to find your way through the dense forest of programs. Heredity 104(6), 520–533 (2009)
12. Leclercq, S., Rivals, E., Jarne, P.: Detecting microsatellites within genomes: significant variation among algorithms. BMC Bioinformatics 8, 125 (2007)
13. Smit, A.F.A., Hubley, R., Green, P.: RepeatMasker, `http://repeatmasker.org`
14. Frith, M.C.: A new repeat-masking method enables specific detection of homologous sequences. Nucleic Acids Res. 39(4), e23 (2011)
15. Pokrzywa, R., Polanski, A.: BWtrs: A tool for searching for tandem repeats in DNA sequences based on the Burrows-Wheeler transform. Genomics 96, 316–321 (2010)
16. Pellegrini, M., Renda, M.E., Vecchio, A.: TRStalker: an efficient heuristic for finding fuzzy tandem repeats. Bioinformatics 26(12), 358–366 (2010)
17. Kolpakov, R., Bana, G., Kucherov, G.: mreps: efficient and flexible detection of tandem repeats in DNA. Nucleid Acids Research 31, 3672–3678 (2003)
18. Kurtz, S., Choudhuri, J.V., Ohlebusch, E., Schleiermacher, C., Stoye, J., Giegerich, R.: REPuter: The Manifold Applications of Repeat Analysis on a Genomic Scale. Nucleic Acids Res. 29(22), 4633–4642 (2001)
19. Ruitberg, C.M., Reeder, D.J., Butler, J.M.: STRBase: a short tandem repeat DNA database for the human identity testing community. Nucleic Acids Res. 29(1), 320–322 (2001)
20. Gelfand, Y., Rodriguez, A., Benson, G.: TRDB—The Tandem Repeats Database. Nucleic Acids Res. 35 (suppl. 1), D80–D87 (2007)
21. Sokol, D, Atagun, F.: TRedD—a database for tandem repeats over the edit distance. Database 2010, article ID baq003, 10.1093/database/baq003 (2010)
22. Danek, A., Pokrzywa, R.: Finding Approximate Tandem Repeats with the Burrows-Wheeler Transform. International Journal of Medical and Biological Sciences 6, 8–12 (2012)
23. Benson, G.: Tandem Repeats Finder: a program to analyze DNA sequences. Nucleic Acids Research 27, 573–580 (1999)
24. Pokrzywa, R.: Application of the Burrows-Wheeler Transform for searching for tandem repeats in DNA sequences. Int. J. Bioinf. Res. Appl. 5, 432–446 (2009)
25. Burrows, M., Wheeler, D.J.: A block-sorting lossless data compression algorithm, SRC Research Report 124, Digital Equipment Corporation, California (1994)
26. Ferragina, P., Manzini, G.: Opportunistic data structures with applications. In: Proceedings of the 41st Annual Symposium on Foundations of Computer Science, pp. 390–398. IEEE Computer Society, Washington, DC (2000)
27. mreps, `http://bioinfo.lifl.fr/mreps`
28. Tandem Repeat Finder, `http://tandem.bu.edu/trf/trf.html`
29. Bhargava, A., Fuentes, F.F.: Mutational Dynamics of Microsatellites. Molecular Biotechnology 44(3), 250–266 (2010)