Advanced Studies in Biology, Vol. 9, 2017, no. 4, 217 - 224
HIKARI Ltd, www.m-hikari.com
https://doi.org/10.12988/asb.2017.7926

C# Desktop Application with Connection to the European Nucleotide Archive for Fast Identification of Tandem Repeats

Done Stojanov

Faculty of Computer Science
University „Goce Delcev" – Stip
Krste Misirkov 10-A, Stip 2000, Republic of Macedonia

Abstract

In order to facilitate the process of computational detection and analysis of tandem repeats in large DNA samples, a desktop application in C# was developed. This application works both with randomly provided content and real samples retrieved from the *European Nucleotide Archive*. In spite of tandem's structure, details regarding the total number of tracked tandems, their range within the sample, the structure of the pattern sequence and the running time of the application are also printed, what underlines its comprehensiveness.

Keywords: Software Tool, Tandem, Repeats

Introduction

Repeated DNA motifs are commonly referred as tandem repeats. These repeated motifs can be found in coding and non-coding DNA. Depending of the length of the repeating unit, there are three major categories of tandem repeats: micro-satellites whose length is usually less than 6 base pairs, mini-satellites which are tandem repeats longer than micro-satellites but shorter than 100 base pairs and satellites which are categorized as tandems of maximum size (usually longer than 100 base pairs). Due to its highly expressed polymorphic nature, tandem repeats are the leading genetic markers nowadays.

When tandem repeats have to be computationally identified, we face up with two major problems. First of all the repeating unit is not known in advance and second each successive copy may differs its precursors in terms of base substitutions or insertions (deletions) of one or more elements.

Current methodologies that address this issue mutually differ in several aspects. There are methods such as [4] that search for only for exact repetitions, with modest practical implementation to real samples. More comprehensive approaches such as [3] also consider base substitutions within repetitions.

Some of the methods try to improve the computational performance, by limiting its application only to micro-satellites. A typical example for this is [5], while [6] can be applied to all types of repeats. As expected, the method of Rivals [5] has favorable computational performance compared to Sagot’s methodology [6], since tandem repeats of shorter repeating units are analyzed.

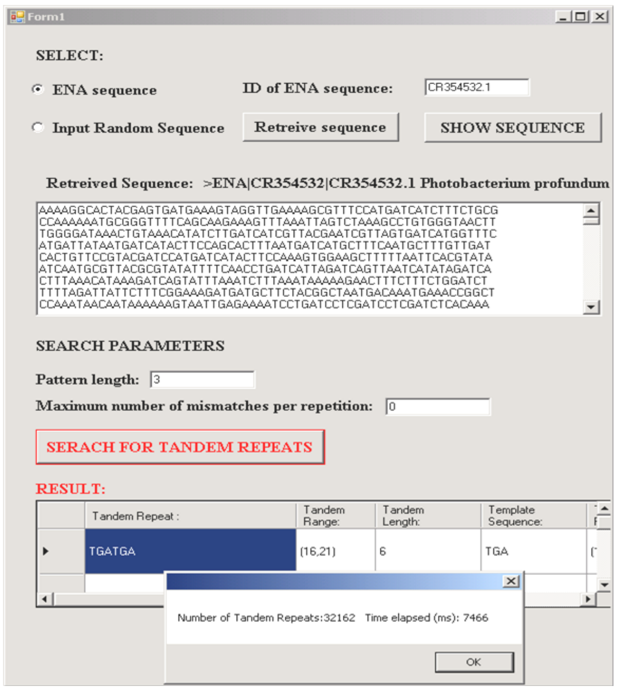
Benson’s program [1] called Tandem Repeats Finder has been the most exploited program for this purpose. This program allows identification of tandem repeats, but the output is filtered according to user’s preferences, what can be seen as a potential drawback. On the other hand, STAR [2] is able to detect tandems regardless any user-specified metrics.

Combinatorial approaches, such as [7] and methods that utilize the computational benefits of suffix tree [8] were also proposed. The methodology proposed by Sokol [7] does not impose limitations upon the types of tandems on which it can be applied, but this algorithm has unfavorable computational performances due to its computational nature. On the other hand [8] utilizes the benefits of the suffix trees in order to speed up and optimize the space requirements for this purpose.

For this purpose a desktop application in C# was developed. The software allows fast tracking of non-crossing tandem repeats. When tested on real samples, retrieved from the European Nucleotide Archive, this program tracked all repeats in few second on machine with moderate hardware performance. The structure of this software tool, the algorithm behind it and the analysis of the obtained results when applied to real samples are subjects of discussion in this paper.

Software Tool

The software tool (*Figure 1*) is able to detect exact or approximate non-crossing tandem repeats. In this version of the software indels (insertions/deletions) are not considered, but only limited number of substitutions per copy. If indels are considered, the computational demands would increase that may limit the application of the software on machines with moderate hardware performance.



SELECT:

(● ENA sequence) ○ Input Random Sequence

ID of ENA sequence: CR354532.1

Retreived Sequence: >ENA|CR354532|CR354532.1 Photobacterium profundum

```
AAAAGGCACTACGAGTGATGAAAGTAGGTTGAAAAGCGTTTCCATGATCATCTTTCTGCG
CCAAAAAATGCGGGTTTTCAGCAAGAAAGTTTAAATTAGTCTAAAGCCTGTGGGTAACTT
TGGGGATAAACTGTAAACATATCTTGATCATCGTTACGAATCGTTAGTGATCATGGTTTC
ATGATTATAATGATCATACTTCCAGCACTTTAATGATCATGCTTTCAATGCTTTGTTGAT
CACTGTTCCGTACGATCCATGATCATACTTCCAAAGTGGAAGCTTTTTAATTCACGTATA
ATCAATGCGTTACGCGTATATTTTCAACCTGATCATTAGATCAGTTAATCATATAGATCA
CTTTAAACATAAAGATCAGTATTTAAATCTTTAAATAAAAAGAACTTTCTTTCTGGATCT
TTTTAGATTATTCTTTCGGAAAGATGATGCTTCTACGGCTAATGACAAATGAAACCGGCT
CCAAATAACAATAAAAAAGTAATTGAGAAAATCCTGATCCTCGATCCTCGATCTCACAAA
```

SEARCH PARAMETERS

Pattern length: 3

Maximum number of mismatches per repetition: 0

RESULT:

Tandem Repeat :	Tandem Range:	Tandem Length:	Template Sequence:
TGATGA	(16,21)	6	TGA

Number of Tandem Repeats:32162 Time elapsed (ms): 7466

Figure 1. Interface of the software tool

There are three input parameters:

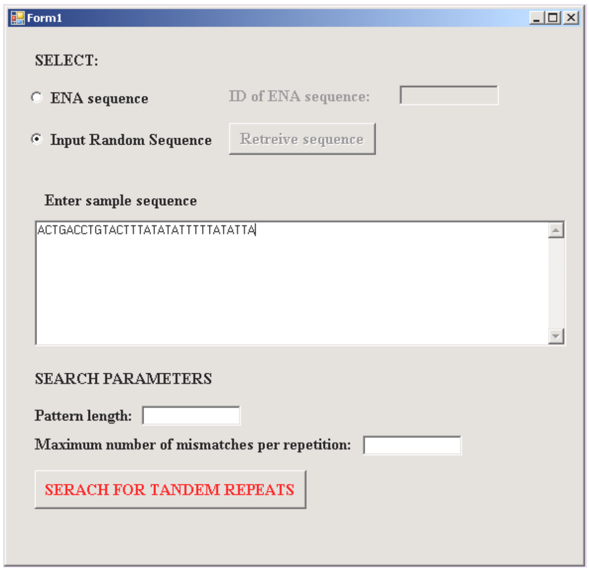
I. ***The sample being analyzed for repeats***

II. ***Template's length*** (k)

III. ***Maximum number of mismatches allowed per copy*** (m)

Regards the first input parameter (*the sample being analyzed for repeats)* this application offers two possibilities (*Figure 1, radio button controls*).

The first option is to work with real sample retrieved from the *European Nucleotide Archive (**ENA**)*. The ENA identifier of the sample has to be provided in the upper text control (*Figure 1*). The retrieval from the database is provided by a *Web Client ()* object. The retrieved content in *FASTA* data format is stored in local textual file. The name of the retrieved sample and its content can be also shown (*Figure 1*). Another option is to work with randomly provided content (*Figure 2*).


Form1

SELECT:

ENA sequence   ID of ENA sequence:

[✓ Input Random Sequence]   Retreive sequence

Enter sample sequence

ACTGACCTGTACTTTATATATTTTTATATTA

SEARCH PARAMETERS

Pattern length:

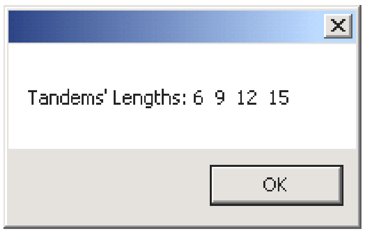
Maximum number of mismatches per repetition:

SERACH FOR TANDEM REPEATS

Figure 2. Working with random input data

The second input parameter (template's length (k)) defines repetition's length. For instance, if $k = 3$ one of the possible tandems is: *ACA ACA ACA*…, on the other hand if $k = 4$ *ACAC ACAC ACAC* … is an example for tandem, where the template *ACAC* of $k = 4$ nucleotides has two adjacent copies.
However the software is able to detect not only exact repeats, but also approximate tandem repeats, where the number of mismatches per copy does not exceed the third input parameter (m). For instance if $k = 4$ and $m = 2$ and *ACAC* is taken as a template, *ACAC AGAC AGAG* … will be also reported as a repeat, since the second copy *AGAC* differs the previous *ACAC* only one element, while the second copy *AGAG* differs the pattern *ACAC* two nucleotides, which is the maximum allowed number of substitutions allowed per repetition.
The output results are printed in *dataGridView* control. For each tracked tandem, in this control we can see: its structure, the offset in the sample, its length, the template and the offset of the template in the sample.

After search, a cumulative analysis is also provided regards: *the total number of tandem repeats*, *the running time of the application* (in milliseconds) (*Figure 1, bottom panel*) and *the variability in tandems' lengths* (*Figure 3*).


Tandems' Lengths: 6 9 12 15

OK

Figure 3.Variability in tandems' lengths

Algorithm

The algorithm that stands behind the software tool is given in addition. The analyzed sample, template's length and the maximum number of base substitutions allowed per copy are the input parameters, while the output is the list of tracked tandems. The way in which this program operates is that each $k - mer$ (word of k consequtive nucleotides) which can be extended at least k nucleotides to the left and to the right and which is not part of already tracked tandem is taken as a template. Each left and right positioned $k - mer$ that does not differ from the template more than m nucleotides is merged with it. This continues until the first left and the first right $k - mer$ that differ more than m nulcotides regards the template are identifed.

For instance, if the random sample *s=AAAAGTAGTTGTAAACTCCTC…* is searched for tandems for $k = 3$ and $m = 1$, *AGT* is taken at first for template (*AAAAGTAGTTGTAAACTCCTC…*). Left-positioned extension is not possible, since its precursor *AAA* differs from *AGT* more than $m = 1$. Two right- positioned extensions are possible, because the following $k - mer$s: *AGT* and *TGT* do not mismatch the template *AGT* more than $m = 1$ elements (*AAAAGTAGTTGTAAACTCCTC…*). Further right positioned extension is not possible (the next right-positioned $k - mer$ *AAA* differs from the template *AGT* more than $m = 1$ element). Now the same procedure is repeated in the unexplored space *AAACTCCTC…* (*AAAAGTAGTTGTAAACTCCTC…*), taking *CTC* as a next template.

INPUT: sample **s,** template's length **k**, maximum number of mismatches $\boldsymbol{m}$
OUTPUT: list of tandems **L**
NOTATIONS:|s| length of the sample

```
i=1;
L=[]; //the list of tandem is empty at the beginning//
START: template=$s_{k+i}s_{k+i+1} \dots s_{2k+i-1}$; //candidate for template//
            if (at least k nucleotides precede and succeed template in s){
            rightExtension=false;
            leftExtension=false;
//try right extension//
templateEnd= $2k + i - 1$;
rightOffset= $2k + i - 1$;
while(template and $s_{rightOffset+1}s_{rightOffset+2} \cdots s_{rightOffset+k}$ differ at most $m$ elements and
rightOffset$\leq$ |s|)
rightOffset+=k;
if (rightOffset!=templateEnd)
//if there is at least one right adjacent copy with at most $m$ mismatces//
{
rightExtension=true;
TandemEnd=rightOffset;
}
else
{
```

```
i++;
goto START
}
//end of right extension//

//try left extension//
templateBeginning=k+i;
leftOffset=k+i;
while(template and $s_{\mathrm{leftOffset-k}} s_{\mathrm{leftOffset-(k-1)}} \cdots s_{\mathrm{leftOffset-2}} s_{\mathrm{leftOffset-1}}$ differ at most $m$ elements
and leftOffset≥ 1)
leftOffset-=k;
if(leftOffset!=templateBeginning)
//if there is at least one left adjacent copy with at most $m$ mismatches//
{
leftExtension=true;
TandemBeginning=leftOffset;
}
else
{
i++;
goto START
}
//end of left extension//

if (leftExtension=true or rightExtension=true)
Add $s_{\mathrm{TandemBeginning}} \cdots s_{\mathrm{TandemEnd}}$ in the list of tandems L}
```

Results and discussion

Photobacterium profundum SS9 chromosome 2 (ENA ID: CR354532.1) containing approximately 2,2 millions nucleotides was searched for tandem repeats applying the software tool. Tests for different combinations of the input parameters $(k, m), k = 3,4$ and $m = 0,1,2$ were performed on Acer Aspire 5570Z computer with Genuine Intel CPU at 1,73 GHz and 2 GB RAM and the number of tracked tandems, tandems' lengths and the running time of the application were recorder, Table 1.

More approximate $(m \neq 0)$ than exact tandem repeats $(m = 0)$ for each template's length $(k = 3$ and $k = 4)$ were found, Table 1, Figure 4. In terms of tandems' length variability, longer tandems were tracked if more mismatches were allowed per copy, Table 1. The longest tracked tandems (39-bases and 56-bases) are found for $k = 3, m = 2$ and $k = 4, m = 2$. In most of the cases results were obtained in less than a half minute. As expected, the more mismatches were allowed per copy, more time was required to output the results.

Table 1. Results of applying the software tool on *Photobacterium profundum SS9 chromosome 2*

Sequence	***k***	***m***	**Number of Tandem Repeats**	**Tandems' Lengths**	**Time (s)**
Photobacterium profundum SS9 chromosome 2 (2 237 943 bp) ID: CR354532.1	3	0	32162	6,9,12,15	7,466
		1	141125	6,9,12,15,18,21,24,27,30,33,39	26,310
		2	179779	6-105	31,111
	4	0	6636	8,12,20	3,577
		1	53589	8,12,16,20,24	10,522
		2	136600	8,12,16,20,24,28,32,26,40,44,48,52,56	23,479

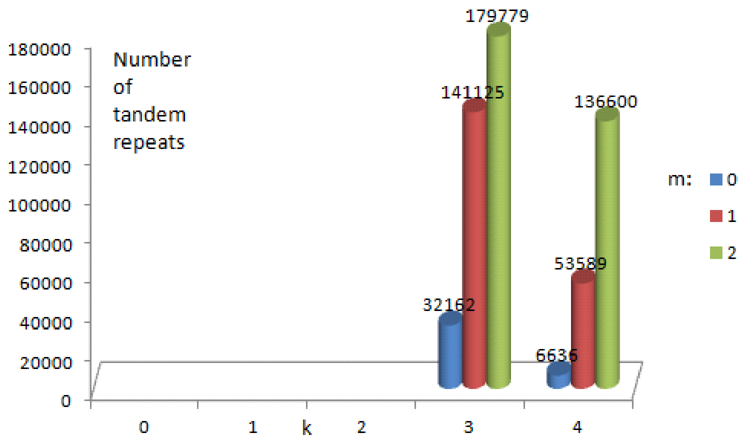


Figure 4. Number of tandem repeats per (k, m) combination

Conclusion

A desktop application in C# was developed for identification of tandem repeats in large DNA samples, such as chromosomes or even entire genomes. Due to application's flexibility, the user is able to analyze randomly provided content or real samples retrieved from the *European Nucleotide Archive* based on sample's ENA ID. Once the content has been downloaded, it can be analyzed for different (k, m) input parameters. Obtained results showed that approximate and exact tandem repeats can be identified on computer with moderate hardware performance in less than a half a minute.

Acknowledgements. This study was supported by the research project *Development of Secure and Reliable Techniques for Data Communication*, funded by „Goce Delcev” University in Stip.

References

[1] G. Benson, Tandem repeats finder: a program to analyze DNA sequences, *Nucleic Acids Research*, **27** (1999), 573-580. https://doi.org/10.1093/nar/27.2.573

[2] O. Delgrange and E. Rivals, STAR: an algorithm to search for tandem approximate repeats, *Bioinformatics*, **20** (2004), 2812-2820. https://doi.org/10.1093/bioinformatics/bth335

[3] R. Kolpakov and G. Kucherov, Finding maximal repetitions in a word in linear time, *40th Annual Symposium on Foundations of Computer Science*, IEEE, 1999, 596-604. https://doi.org/10.1109/sffcs.1999.814634

[4] M.G. Main and R.J. Lorentz, An O(n log n) algorithm for finding all repetitions in a string, *Journal of Algorithms*, **5** (1984), 422-432. https://doi.org/10.1016/0196-6774(84)90021-x

[5] E. Rivals, O. Delgrange, J.P. Delahaye, M. Dauchet, M.O. Delorme, A. Hénaut and E. Ollivier, Detection of significant patterns by compression algorithms: the case of approximate tandem repeats in DNA sequences, *Bioinformatics*, **13** (1997), 131-136. https://doi.org/10.1093/bioinformatics/13.2.131

[6] M.F. Sagot and E.W. Myers, Identifying satellites and periodic repetitions in biological sequences, *Journal of Computational Biology*, **5** (1998), 539-553. https://doi.org/10.1089/cmb.1998.5.539

[7] D. Sokol, G. Benson and J. Tojeira, Tandem repeats over the edit distance, *Bioinformatics*, **23** (2007), e30-e35. https://doi.org/10.1093/bioinformatics/btl309

[8] J. Stoye and D. Gusfield, Simple and flexible detection of contiguous repeats using a suffix tree, *Theoretical Computer Science*, **270** (2002), 843-856. https://doi.org/10.1016/s0304-3975(01)00121-9

Received: October 16, 2017; Published: November 17, 2017