An Online Interface to find Approximate Tandem Repeats in DNA Sequences

Nechama Gurwitz

August 30, 2006

1. Introduction – About Tandem Repeats

Tandem repeats are sequences of repeated nucleotides in DNA. Identifying the repeats in DNA sequences is very important, as molecular biologists use them for a variety of applications. Some diseases such as fragile-X mental retardation, Huntington's disease, myotonic dystrophy, spinal and bulbar muscular atrophy, and Friedreich's ataxia can be identified by substantial repeats of DNA patterns.[1] Repeated sequences are also used for human identification.[2]

Tandem repeats can be *perfect* repeats, consisting of a continuous repeated substring, e.g. *abcdabcdabcd.* However, most repeats are not perfect; they contain errors so that the repeated substrings are not identical. These *approximate* tandem repeats contain substitutions, e.g. *abcd abdd abcd,* as well as insertions and deletions, where there are extraneous characters or missing characters in the sequence, e.g. *abcd abc abcd abacd*.

2. Background

Dr. Sokol developed two programs to find approximate tandem repeats. One allows mismatches and the other allows mismatches as well as insertions and deletions.

2.1. Algorithm

The algorithm is recursive, splitting a string into increasingly smaller segments with $O(\log n)$ iterations, where n is the length of the string. For each substring, all repeats that cross the center of the string are located and reported.[3]

2.2. Prior Work

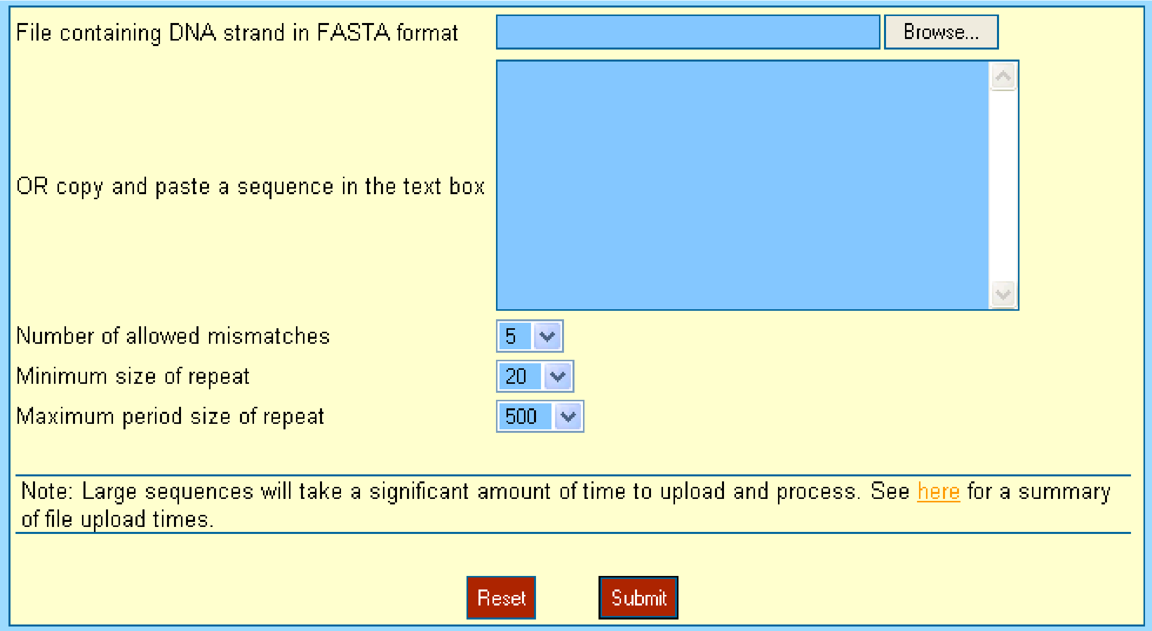
The first program that Dr. Sokol developed, in conjunction with Prof. Gad Landau, finds approximate tandem repeats with k mismatches. She also developed a post-processing filter to sift out insignificant repeats and prevent repeats from being reported twice in slightly different manners. The program was put online in a limited fashion and without the post-processing filter. The online program only processed sequences containing up to 1024 characters, which is a severe limitation, considering that typical DNA sequences contain millions of nucleotides. Additionally, the user was not able to upload the sequence directly from a file.

Recently, Dr. Sokol, working with a student, Justin Tojeira, produced a more comprehensive program to find tandem repeats over the edit distance.[4] This program allows insertions and deletions as well as mismatches.

3. My Work

3.1. Web Page

I created a web site for both of the tandem repeats programs, which includes an explanation of the algorithm, downloads of the original code, and references. There is a page for each program, containing a user-friendly form that allows the user to either upload a file or paste a DNA sequence in a text area. The form also prompts the user to select the parameters for the program from a range of possible choices (Figure 1). In addition, I added a counter to monitor how many visitors the site receives (Figure 2).



File containing DNA strand in FASTA format

OR copy and paste a sequence in the text box

Number of allowed mismatches: 5

Minimum size of repeat: 20

Maximum period size of repeat: 500

Note: Large sequences will take a significant amount of time to upload and process. See here for a summary of file upload times.

Figure 1

Visitors to this site: 00000536

Figure 2

3.2. CGI

In order to create a dynamic webpage that generates results based on user input, an interface is needed to allow a program to produce the webpage. The Common

Gateway Interface (CGI) allows a program to receive requests from a web browser and pass its results back to the browser where they are displayed.[5] As our programs were already written in C/C++, we used the same C++ programs in the CGI version. The only changes we made concerned the methods of input and output; however, the processing remained the same. Outputting a CGI program follows the same rules as standard C++, with the addition of an extra line in the beginning telling the browser to expect an HTML page, and HTML tags surrounding the information as necessary: e.g. *cout<<"<b>Hello</b>"<<endl;* produces **Hello.**

Input to the CGI program is more complicated. The program receives the values from a HTML form (which has an action of POST) in the standard input as a long string of names and values. The input must then be parsed to obtain the form values. The original online program had functions to parse the standard input, separating it into name-value pairs and then further separating the names from their values. However, this method would be too complicated to use with the addition of file upload. The browser sends information about the uploaded file, such as its content-type and content-length to the CGI program.[6] Parsing the information to get only the file contents is complex, so I searched for an existing class that would do the work for me. I used the RudeCGI™ class created by Matt Flood[7], which easily allowed me to obtain file contents, as well as the other form parameters. Using the class simplified the program and saved me a lot of work.

3.3 FASTA files

FASTA is a format used to store DNA sequences. It consists of a single description line, explaining what the sequence is, followed by all the sequence data. The description line is characterized by a single greater-than symbol (">") as the first character. The letter "N" symbolizes unknown nucleotides.[8] (Figure 3)

```
 >12 dna_rm:chromosome chromosome:RGSC3.4:12:1:46782294:1
TACAGATTTTAACTTCCTACAAAGGAATGGTTCAACAAGAGGAACCTTTACACAGAACAN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNCAAGTCTTGTTGTTAAATCA
AGCTACTACTTTGTTTAATTCCGTTTTCTGAAGGTGGAGTTGGTATGAAAATAAACTACT
GTATTTATTTGATAATACATCAATTCTGATGGGCTTGCTTTTAAACTATCCAATACATTA
CAGACTCTGCTTAATCTGAAAGGTTATGGTTTCAGGAAAATGTTCAGGCTGATGGCCTCT
```

Figure 3

Obviously, biologists would prefer to upload sequences in FASTA format, rather than be required to submit a plain text file. To allow the program to accept FASTA files, I created a function to first test that the file is a valid FASTA file. It ensures that the description line is present and that there are no extraneous characters. The function then removes the first line, as well as all whitespace, so that the program deals with a single, unbroken string of DNA nucleotides. If the file is not in FASTA format, the program will not process the file. It alerts the user that the file is not in FASTA format, and refers the use to a link explaining what the proper format should be.

3.4 Unknown Nucleotides

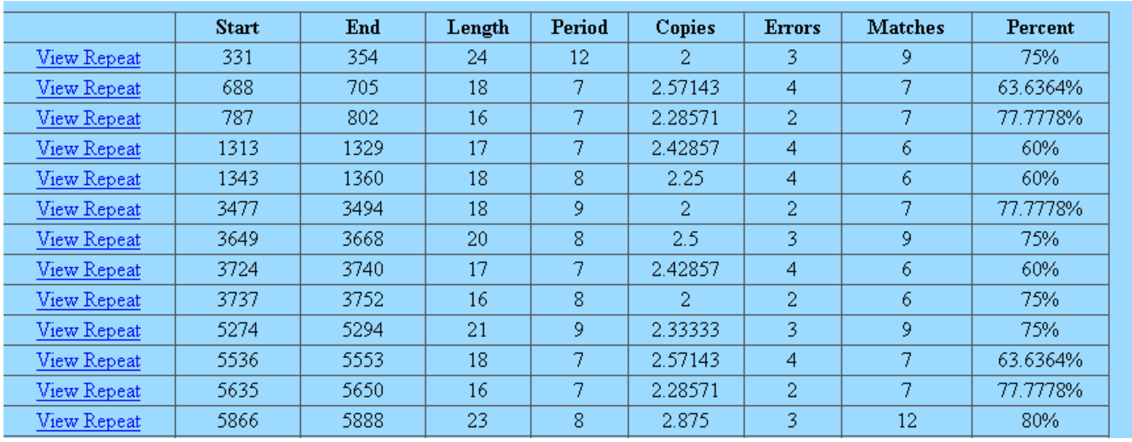
DNA sequences contain large amounts of unknown nucleotides; sometimes they comprise over fifty percent of the sequence. As mentioned previously, those nucleotides are symbolized by the letter "N". The prevalence of "N"'s skewed the results tremendously, as the program reported sequences consisting completely of unknown nucleotides as repeats. I created a string of over 8,000 characters with no approximate repeats, and we replace the "N"s in the sequence with the miscellaneous characters. I reuse the miscellaneous characters repeatedly; however, the user is restricted to choose a maximum period size of up to 4,000 characters (which corresponds to a total length of 8,000 characters). This way, there are no repeats of unknown characters in the results.

3.5. Post-Processing Filter

I modified the first program, combining the repeats finder with a post-processing filter. The filter refines the collection of repeats by suppressing insignificant results. The second program already incorporates filtering into the processing stage, so an additional filter is not necessary.

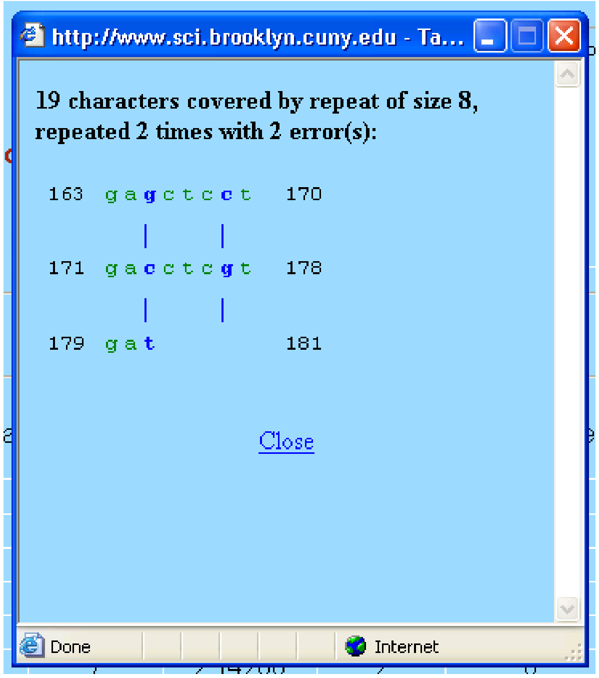
3.6. Output

Results are output in a succinct table, stating the statistics about the repeat, such as its start, end, length, and amount of errors (Figure 4). The "View Repeat" link allows the user to view the contents of the repeat. Mismatches are marked in a different color for clarity (Figure 5).



	Start	End	Length	Period	Copies	Errors	Matches	Percent
View Repeat	331	354	24	12	2	3	9	75%
View Repeat	688	705	18	7	2.57143	4	7	63.6364%
View Repeat	787	802	16	7	2.28571	2	7	77.7778%
View Repeat	1313	1329	17	7	2.42857	4	6	60%
View Repeat	1343	1360	18	8	2.25	4	6	60%
View Repeat	3477	3494	18	9	2	2	7	77.7778%
View Repeat	3649	3668	20	8	2.5	3	9	75%
View Repeat	3724	3740	17	7	2.42857	4	6	60%
View Repeat	3737	3752	16	8	2	2	6	75%
View Repeat	5274	5294	21	9	2.33333	3	9	75%
View Repeat	5536	5553	18	7	2.57143	4	7	63.6364%
View Repeat	5635	5650	16	7	2.28571	2	7	77.7778%
View Repeat	5866	5888	23	8	2.875	3	12	80%

Figure 4



19 characters covered by repeat of size 8, repeated 2 times with 2 error(s):

```
163  g a g c t c c t   170
         |       |
171  g a c c t c g t   178
         |       |
179  g a t             181
```

Figure 5

4. Conclusion and Future Plans

The online website provides a clear interface for biologists to find tandem repeats in sequences that they are studying. It enables them to quickly receive clear, organized results, which they will then study to find the implications.

In the future, we would like to run the program on thousands of sequences and create a tandem repeats database from the results. This would enable users to select a chromosome and immediately view its repeats, without first running the program again. Further improvements to the filtering process are in progress as well.

References

[1] C. T. Caskey et al. An unstable triplet repeat in a gene related to Myotonic Dystrophy. *Science*, 255:1256–1258, 1992.

[2] Agrawal S, Khan F, Talwar S, Nityanand S. Short tandem repeat technology has diverse applications: Individual identification, phylogenetic reconstruction and chimerism based post haematopoietic stem cell transplantation graft monitoring. *Indian Journal of Med Science* 2004;58:297-304.

[3] G. M. Landau, J. P. Schmidt and D. Sokol. An Algorithm for Approximate Tandem Repeats. *Journal of Computational Biology,* Volume 8, p. 1-18, 2001.

[4] D. Sokol, G. Benson, and J. Tojeira. Tandem Repeats over the Edit Distance. To be presented at the European Conference on Computational Biology (ECCB) 2006.
Also to appear in *Bioinformatics*.

[5] Common Gateway Interface: Overview. http://hoohoo.ncsa.uiuc.edu/cgi/intro.html

[6] Common Gateway Interface: CGI Script Input. http://hoohoo.ncsa.uiuc.edu/cgi/in.html

[7] RudeCGI C++ Library: http://rudeserver.com/cgiparser/index.html

[8] FASTA Format Description http://www.ncbi.nlm.nih.gov/blast/fasta.shtml