RESEARCH

# Fast de novo transposable element annotation

Carly E Schaeffer[1†], Nathaniel D Figueroa[1†], Xiaolin Liu[1,2] and John E Karro[1,2,3,4*^]

[*]Correspondence: karroje@miamiOH.edu
[1]Department of Computer Science and Software Engineering,
Full list of author information is available at the end of the article
[†]Equal contributor ^Corresponding Author

**Abstract**

**Background:** The problem of *de novo* identification of transposable elements (the discovery and annotation of transposable elements without the use of a pre-complied profile) has been addressed by a number of tools – all of which are either based on computationally complex algorithms that cannot scale to whole-genome use, or whose sensitivity suffers significantly from the presence of sequence variation. Here we present phRAIDER (Pattern Hunter based Rapid Ab Initio Detection of Elementary Repeats), a tool that can quickly identify and mask tranposable elements in a a newly sequenced genome and is robust to the sequence variation present in real data.

**Results:** To be written later.

**Conclusions:** To be written later.

**Keywords:** transposable elements; elementary repeats; pattern hunter; genomic masking

## Background

Transposable elements, or TEs, are genomic sequences that have at some point had the capacity to insert copies of themselves into other genomic locations, resulting in homologous families of sequences spread across the genome. TEs are present in almost every higher order genome, covering as much as 45% of the human genome and 90% of the maize genome [1, 2]. TEs have proved an important source of data in a number of studies [3, 4, 5, 6], but given their prevalence, it is just important for those studying the genome to have TEs masked out – their bases replaced by N to allow for easy identification and filtering. Failure to filter can reek havoc with genomic analysis tools. For example. if not filtered, TEs can trigger huge numbers of false positives automated gene finding tool [7], as well as inflate tool runtime.

The most well known tools for repeat identification is RepeatMasker, with the new nHMMER showing promise as well) [8, 9]. Both work on the principle of a library-based search. That is, to identify transposable elements to a known family, the tools require some description of sequences in the family (e.g. a ancestral sequence for BLASTing, or a profile HMM description) that is used as the basis for identification. But, much like we ask how the snow plow driver gets to work, or how the author of a dictionary authors look up the spelling of a word [10], we must ask how the libraries on which the tools are based are initially compiled. Library-based tools are useless for the discovery of new families, and cannot be applied to a newly sequenced genome for which such a library has not been compiled.

Within mammalian species we can largely rely on homology relationships to port libraries across species. A library description of a human ALU sequence can likely

be used for the initial annotation of a newly sequenced primate genome, and many human LINEs will extend to rodent, or even more distant, mammalian species. This does not hold so well in plants: in many cases TE composition of a given plant organism is species-specific. For example, a rice-based TE library will only identify 25% of the TEs in the maize genome [7].

To solve this problem we turn to *de novo* TE identification tools: tools identifying TEs though a genome using only the genome sequence information. A number of such tools are discussed in the literature, operating on a number of different principles. Tools such as RECON and PILER are based on self-alignment, using WU-BLAST and LASTZ for the alignment [11, 12, 13, 14]. RECON show good sensitivity but is computationally intensive and infeasible for use on whole genomes (requiring 60 hours for 18Mb rice genome int 2013), while PILER achieves a good runtime with very low sensitivity [7]. ReAS [15] and RepeatScout [16] are based on $k$-mer searches, with the earlier showing less sensitivity than RECON [7]. RepeatGluer [17, 18] is based on a variation of DeBrujin graphs, which allows for a decomposition of TE families into domains, but is very, very computationally expensive. In Saha *et al.* the authors perform an extensive comparison of the tools, and conclude that RepeatScout is the best tool overall for assembled genomes, while ReAS the best when dealing with unassembled sequence fragments [19].

### Elementary Repeats

Another line of development, first proposed by Zheng and Lonardi [20], involves the use of *elementary repeats*. Similar to the RepeatGluer domains, elementary repeats are decompositions of TEs into basic building blocks. Identification of these building blocks are sufficient for the purpose of masking, and can be assembled into Transposable Elements for those interested in TEs themselves.

While it is notoriously difficult to mathematically model transposable elements [11], elementary repeats are more conducive to a formal description. For a given genome, a nucleotide sequence $r$ is an elementary repeat if:

1. It is of at least length $l$. (The length requirement.)
2. At least $f$ exact copies of $r$ appear in the genome. (The frequency requirement.)
3. There is no proper substring of $r$ that is of at least length $l$ and appears in the genome independently of $r$. (The minimality requirement.)
4. There is no string that contains $r$, appears in the genome the same number of times as $r$, and satisfies conditions 1-3. (The maximality requirement.)

The first two criteria are straight-forward, allowing us to designate length and frequency thresholds for our model. The minimality requirement ensures that an elementary repeat does not contain anything that could be viewed as an independent elementary repeat – that it is the "smallest building block" that we can find. While the maximality requirement ensures that it is not tied to some other, larger, elementary repeat.

Having proposed this definition, Zheng and Lonardi developed an identification algorithm that had a runtime quadratic in the genome size – so not of practical use on whole-genome analysis. This was refined to linear time by He and also by Huo *et al.* [21, 22] based on variations of suffix tree approaches, but are limited in their

ability to handle sequence variation. As transposable elements naturally suffer from copy mistakes and accumulate instance-specific base substitutions over time, this is a significant limitation.

### RAIDER

It was with the objective of creating a linear time identification algorithm that could handle variation through use of PatternHunter-like spaced seeds that we developed the prototype RIADER [23]. A rough implementation was first presented in Figueroa *et al.*, with more details in the Figueroa masters thesis [24, 25]. Not itself fully able to handle the spaced seed component, RAIDER was built along an alternate, but equivalent definition of elementary repeats based on $l$-mers (sequences of length $l$). Specifically, it was observed that the minimality condition could be rewritten as: There is no $l$-mer contained with $r$ that appears in the genome more times than $r$. From there we make our four core observations that form the basis for the RAIDER algorithm:

1. An $l$-mer cannot belong to two different elementary repeats.
2. Any $l$-mer in the genome that occurs $f$ or more times is either an elementary repeat or belongs to one.
3. Any two $l$-mers belonging to an elementary repeat must appear the same number of times in the genome.
4. If two sequences in the genome that are *maximally identical* (that is, cannot be extended in either direction and still be identical), these sequences cannot belong to a larger elementary repeat.

By "belong" we mean "is a substring of"; the reason for the terminology will be explained shortly. For discussion and proof, see the Figueroa Thesis [25].

Based on these observations, we discover we can find all elementary repeats in a single scan of the genome. Specifically, as we scan from left to right, we track $l$-mer occurrences and identify multiple copies of the same $l$-mer. When we find the same sequence of $l$-mers occurring multiple times in a row, we can mark it as a tentative family, then break it down later if we discover violations of the minimality condition. The algorithm is summarized in Figueroa *et al.*, and discussed in detail in the Figueroa Thesis [24, 25].

Results of the preliminary implementation were promising. Based on the Saha *et al.* analysis putting RepeatScout as the top *de novo* search tool, comparisons were made against that tool. RAIDER result quality was close, but slightly behind – with runtime order of magnitudes better. On human chromosome 22 we saw a 12× speedup from RepeatScout to RAIDER (2344 seconds to 192 seconds), while coverage of the RepBase [26] ancestral sequence actually improved (77% to 84%), while on mouse chromosome 19 we saw the same speedup with a significant drop in coverage (53% to 30%). On the full human genome RAIDER ran in 6.3 hours, while RepeatScout was unable to complete it run. For details, see Figueroa *et al.* [24].

### Spaced Seeds

PatternHunter, first proposed as a very successful augmentation to BLAST [23, 27], is based on the notion of *spaced seeds*. In the context of BLAST, instead of looking

for a contiguous substring match to trigger the alignment extension, PatternHunter uses a more refined match pattern. For example, the seed 1111110111111 would specify that we were looking for an exact match of two length-six substrings separated by a single base (which might or might not match). In the toy seed $s = 110101$, we would say the strings $AACACA$ and $AAGACA$ match because all miss-matched positions correspond to a 0 match. The string $AACACA$ does not math $TACACA$ (with respect to $s$), as the single miss-match does not correspond to the 1. Nor does $AACACA$ match any substring of $TTAACTCA$, though it does match a substring of $T\mathbf{AAGATA}C$.

The incorporation of spaced seeds into BLAST lead to significant improvements at virtually no cost in runtime [23]. In our context, we will use them to allow limited miss-matches between instances of "the same" elementary repeat. The preliminary version of RAIDER from Figueroa *et al.* made some attempt to incorporate this. It was successful enough to provide a proof-of-concept, but a significant amount of work was left to be done to allow for true space seed usage.

In this paper we present phRAIDER (PatternHunter-based RAIDER), a fast tool for the identification and making of transposible elements in both assembled and unassembled genomes. Code may be obtained [NEED WEB ADDRESS]. In our Methods section we present the new model that allows us to extend RAIDER to correctly use spaced seeds, and briefly outline the algorithm (with more details provided in the supplementary materials), with a analysis of phRAIDER performance in our Results.

## Methods

In order to allow the use of a spaced seed strategy, we will need to define a new model of elementary repeats that will accommodate it. In the following we will first outline our theoretical framework (described in detail in the Supplementary Materials), and then review the phRAIDER algorithm.

### Seeded Elementary Repeats

In order to present our new model of elementary repeats, we first need to define a few terms:

1. A spaced seed is a binary string that starts and ends with 1 symbols (as defined in Li *et al.* [23]).
2. A *sequence descriptor* is a DNA string that allows * symbols (indicating that a position where base content does not matter). A string matches a sequence descriptor of equal length if it has the same base in all non-* positions.
3. We say the frequency of $r$ in a genomic sequence $G$ (denoted $\nu_G(r)$) is the number of subsequences in $G$ that match $r$.
4. A spaced seed $s$ *hits* a substring of sequence descriptor $r$ if that substring is the same length of $s$ and, when aligned, every $*$ in the substring matches a 0 in the string. (Example: if $s = 11011$ and $r = AAA * TTTTT$, then $s$ hits both $AA * CC$ and $TTTTT$, but not $A * TTT$.)
5. The *decomposition* of $r$ (with respect to seed $s$) is the set of all substrings of $r$ that are hit by $s$. For the *generalized decomposition*, we take each member of the decomposition and replace by $*$ any base that aligned to a 0 in the

seed. (Example: if $s = 11011$ and $r = AAATTT$, then $AAATT$ is in the decomposition, so $AA * TT$ is in the generalized decomposition.)

6 We say a string $s$ *covers* a sequence descriptor $r$ if every base in $r$ is contained within at least one substring from the decomposition of $r$.

**Definition 1** *Given a fixed genomic sequence $G$, a integer $f$, and spaced seed $s$, a sequence descriptor $r$ is a repeat element descriptor if:*

1 *$s$ covers $r$. (Minimum length requirement.)*

2 *$\nu_G(r) \geq f$. (Frequency requirement.)*

3 *For every string $w$ in the generalized decomposition of $r$ (w.r.t. $s$), $\nu_G(w) = \nu_G(r)$. (Minimality requirement.)*

4 *There is no sequence descriptor $r'$ such that (1) $r$ is a substring of $r'$, (2) $\nu_G(r') = \nu_G(r)$, and (3) $r'$ satisfies conditions 1-3. (Maximality requirement.)*

We will refer to the set of sequences in the genome that match the repeat family descriptor to be the repeat element family of the descriptor.

**Theorem 1** *Definition 1 is equivalent to the Z&L definition of elementary repeats when $s$ contains no 0 symbols.*

In the Supplementary materials we prove our definition equivalent to the Figueroa definition (Definition 5 from the thesis) – which is proved to be equivalent to the Z&L definition in that work [24].

It is instructive to look at a specific example. Consider the toy seed 11011, let $f = 2$, and assume the genome contains a unique instance of $g_1 = AAAAA\mathbf{C}TTTTT$ and a unique instance of $g_2 = AAAAA\mathbf{G}TTTTT$. Consider the sequence descriptor $r_1 = AAAAA * TTTTT$. A simple inspection shows us that conditions (1)-(3) of Definition 1 hold, so if we assume the surrounding sequence is such that (4) holds as well, this is an elementary repeat descriptor for this family of sequences. (Note $AA*AA*TT*TT$ also qualifies – the elementary repeat descriptor is not necessarily unique.) But if we add a second instance of $g_3 = AAAAAGTTTTT$, something happens that was not possible in the unseeded version. While $r_1$ continues to meet the definition, the $g_2/g_3$ subsequence $AAAAGTTTT$ also induces an elementary repeat family $r_2 = AAAA*TTTT$. (Note that we cannot extend this by, for example, one base to the left, as the $AAAAA$ substring would cause a violation of condition (3)). So $r_2$ matches strings in $G$ that are substrings of those that $r_1$ matches – a condition not possible with the seedless elementary repeats. This also then violates one of our core observation on which RAIDER is broken: we now have an $l$-mer ($AAAA*$) that belongs to multiple descriptors.

A second complicating factor is that our repeat element descriptor is not a sequence of consecutive $l$-mers. In the seedless version, an elementary repeat of length $n$ can be viewed as a sequence of $n-l+1$ $l$-mers, each overlapping the last by exactly $l-1$ bases. In this definition we use members of the generalize decomposition, and find they are not necessarily offset by just one base: given the $g_1$ and $g_2$ example above, we see that the $r_1 = AAAAA * TTTTT$ descriptor is composed of $AA * AA$, $AA * TT$, and $TT * TT$ – each offset from the last by 3. As RAIDER assumed that it need a look-back of only one, this too breaks the algorithm.

### phRAIDER

In order to adapt the RAIDER algorithm to the new definition, we require observations parallel to those on which RAIDER was built. As it turns out, its easy to

adopt those. Consider, for example, the first observation: "An $l$-mer cannot belong to two different elementary repeats." We have already seen a violation, but can eliminate this by changing the definition of *belongs*:

**Definition 2** *Given a elementary repeat descriptor $r$ for seed $s$ with length $l$, we say a length $w$ substring belongs to $r$ if it matches a member of the generalized decomposition of $r$ (w.r.t. to $s$).*

We notice in the above example, the string $AAAAA$ belongs to the generalized decomposition, hence could not appear in both elementary repeats. But the string $AAAAT$ does not match any string in the generalized decomposition of $r$, and hence was free to belong to the descriptor of a "distinct" family. See the Supplementary Materials for a proof of all the core observations.

### phRAIDER Algorithm

TO DO: Give a pseudocode description of the algoirthm and some minimal discussion.

### Analysis

TO DO: Describe pra analysis and quality metrics.

## Results

Points to be made:

1. Show: RAIDER2 has better quality masking results then RS.
2. Show: RAIDER2 is better than RAIDER.
3. Show: Seeds matter.
4. Show: Runtime / memory usage

Possible tables / plots:

- Table showing results on different chromosomes (pra, runtime, memory).
- Plot of runtime against genome size (simulated data to control size)?
- Plot of memory usage (simlated data to control size)?
- We might want to compare the ability to detect transposable elements againsts other types of repeats. (Should be easy to do using simulator.)

## Conclusions

Hopefully will beconcluding that RAIDER is a better tool than RepeatScout. Posible discussion of tool features.

**Competing interests**
The authors declare that they have no competing interests.

**Author's contributions**
Text for this section ...

**Acknowledgements**
Text for this section ...

**Author details**
[1]Department of Computer Science and Software Engineering,. [2]Center for Molecular and Structural Biology,. [3]Department of Statistics,. [4]Department of Microbiology, Miami University, Oxford, USA.

**References**


1. Lander, E.S., Linton, L.M., Birren, B., Nusbaum, C., Zody, M.C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W., Funke, R., Gage, D., Harris, K., Heaford, A., Howland, J., Kann, L., Lehoczky, J., LeVine, R., McEwan, P., McKernan, K., Meldrim, J., Mesirov, J.P., Miranda, C., Morris, W., Naylor, J., Raymond, C., Rosetti, M., Santos, R., Sheridan, A., Sougnez, C., Stange-Thomann, N., Stojanovic, N., Subramanian, A., Wyman, D., Rogers, J., Sulston, J., Ainscough, R., Beck, S., Bentley, D., Burton, J., Clee, C., Carter, N., Coulson, A., Deadman, R., Deloukas, P., Dunham, A., Dunham, I., Durbin, R., French, L., Grafham, D., Gregory, S., Hubbard, T., Humphray, S., Hunt, A., Jones, M., Lloyd, C., McMurray, A., Matthews, L., Mercer, S., Milne, S., Mullikin, J.C., Mungall, A., Plumb, R., Ross, M., Shownkeen, R., Sims, S., Waterston, R.H., Wilson, R.K., Hillier, L.W., McPherson, J.D., Marra, M.A., Mardis, E.R., Fulton, L.A., Chinwalla, A.T., Pepin, K.H., Gish, W.R., Chissoe, S.L., Wendl, M.C., Delehaunty, K.D., Miner, T.L., Delehaunty, A., Kramer, J.B., Cook, L.L., Fulton, R.S., Johnson, D.L., Minx, P.J., Clifton, S.W., Hawkins, T., Branscomb, E., Predki, P., Richardson, P., Wenning, S., Slezak, T., Doggett, N., Cheng, J.F., Olsen, A., Lucas, S., Elkin, C., Uberbacher, E., Frazier, M., Gibbs, R.A., Muzny, D.M., Scherer, S.E., Bouck, J.B., Sodergren, E.J., Worley, K.C., Rives, C.M., Gorrell, J.H., Metzker, M.L., Naylor, S.L., Kucherlapati, R.S., Nelson, D.L., Weinstock, G.M., Sakaki, Y., Fujiyama, A., Hattori, M., Yada, T., Toyoda, A., Itoh, T., Kawagoe, C., Watanabe, H., Totoki, Y., Taylor, T., Weissenbach, J., Heilig, R., Saurin, W., Artiguenave, F., Brottier, P., Bruls, T., Pelletier, E., Robert, C., Wincker, P., Smith, D.R., Doucette-Stamm, L., Rubenfield, M., Weinstock, K., Lee, H.M., Dubois, J., Rosenthal, A., Platzer, M., Nyakatura, G., Taudien, S., Rump, A., Yang, H., Yu, J., Wang, J., Huang, G., Gu, J., Hood, L., Rowen, L., Madan, A., Qin, S., Davis, R.W., Federspiel, N.A., Abola, A.P., Proctor, M.J., Myers, R.M., Schmutz, J., Dickson, M., Grimwood, J., Cox, D.R., Olson, M.V., Kaul, R., Shimizu, N., Kawasaki, K., Minoshima, S., Evans, G.A., Athanasiou, M., Schultz, R., Roe, B.A., Chen, F., Pan, H., Ramser, J., Lehrach, H., Reinhardt, R., McCombie, W.R., de La Bastide, M., Dedhia, N., Blöcker, H., Hornischer, K., Nordsiek, G., Agarwala, R., Aravind, L., Bailey, J.A., Bateman, A., Batzoglou, S., Birney, E., Bork, P., Brown, D.G., Burge, C.B., Cerutti, L., Chen, H.C., Church, D., Clamp, M., Copley, R.R., Doerks, T., Eddy, S.R., Eichler, E.E., Furey, T.S., Galagan, J., Gilbert, J.G., Harmon, C., Hayashizaki, Y., Haussler, D., Hermjakob, H., Hokamp, K., Jang, W., Johnson, L.S., Jones, T.A., Kasif, S., Kaspryzk, A., Kennedy, S., Kent, W.J., Kitts, P., Koonin, E.V., Korf, I., Kulp, D., Lancet, D., Lowe, T.M., McLysaght, A., Mikkelsen, T., Moran, J.V., Mulder, N., Pollara, V.J., Ponting, C.P., Schuler, G., Schultz, J., Slater, G., Smit, A.F., Stupka, E., Szustakowski, J., Thierry-Mieg, D., Thierry-Mieg, J., Wagner, L., Wallis, J., Wheeler, R., Williams, A., Wolf, Y.I., Wolfe, K.H., Yang, S.P., Yeh, R.F., Collins, F., Guyer, M.S., Peterson, J., Felsenfeld, A., Wetterstrand, K.A., Patrinos, A., Morgan, M.J., de Jong, P., Catanese, J.J., Osoegawa, K., Shizuya, H., Choi, S., Chen, Y.J., Szustakowki, J., Consortium, I.H.G.S.: Initial sequencing and analysis of the human genome. Nature **409**(6822), 860–921 (2001)
2. SanMiguel, P., Tikhonov, A., Jin, Y.K., Motchoulskaia, N., Zakharov, D., Melake-Berhan, A., Springer, P.S., Edwards, K.J., Lee, M., Avramova, Z., Bennetzen, J.L.: Nested retrotransposons in the intergenic regions of the maize genome. Science (New York, NY) **274**(5288), 765–768 (1996)
3. Arndt, P.F., Hwa, T.: Identification and measurement of neighbor-dependent nucleotide substitution processes. Bioinformatics (Oxford, England) **21**(10), 2322–2328 (2005)
4. Karro, J.E., Peifer, M., Hardison, R.C., Kollmann, M., von Grünberg, H.H.: Exponential decay of GC content detected by strand-symmetric substitution rates influences the evolution of isochore structure. Molecular Biology and Evolution **25**(2), 362–374 (2008)
5. Mugal, C.F., von Grünberg, H.H., Peifer, M.: Transcription-Induced Mutational Strand Bias and Its Effect on Substitution Rates in Human Genes. Molecular Biology and Evolution **26**(1), 131–142 (2008)
6. Hardison, R.C.: Covariation in Frequencies of Substitution, Deletion, Transposition, and Recombination During Eutherian Evolution. Genome research **13**(1), 13–26 (2003)
7. Jiang, N.: Overview of repeat annotation and de novo repeat identification. Methods in molecular biology (Clifton, N.J.) **1057**(Chapter 20), 275–287 (2013)
8. Smit, A.F., Hubley, R., Green, P.: RepeatMasker Open 4.0
9. Wheeler, T.J., Eddy, S.R.: nhmmer: DNA homology search with profile HMMs. Bioinformatics (Oxford, England) **29**(19), 2487–2489 (2013)
10. Pratchett, T.: Hogfather: A Novel of Discworld. HarperPrism, New York
11. Bao, Z., Eddy, S.R.: Automated de novo identification of repeat sequence families in sequenced genomes. Genome research **12**(8), 1269–1276 (2002)
12. Edgar, R.C., Myers, E.W.: PILER: identification and classification of genomic repeats. Bioinformatics (Oxford, England) **21 Suppl 1**(Suppl 1), 152–8 (2005)
13. Lopez, R., Silventoinen, V., Robinson, S., Kibria, A., Gish, W.: WU-Blast2 server at the European Bioinformatics Institute. Nucleic acids research **31**(13), 3795–3798 (2003)
14. Harris, R.S.: Improved pairwise alignment of genomic DNA. PhD thesis, The Pennsylvania State University (2007)
15. Li, R., Ye, J., Li, S., Wang, J., Han, Y., Ye, C., Wang, J., Yang, H., Yu, J., Wong, G.K.-S., Wang, J.: ReAS: Recovery of ancestral sequences for transposable elements from the unassembled reads of a whole genome shotgun. PLoS computational biology **1**(4), 43 (2005)
16. Price, A.L., Jones, N.C., Pevzner, P.A.: De novo identification of repeat families in large genomes. Bioinformatics (Oxford, England) **21 Suppl 1**(Suppl 1), 351–8 (2005)
17. Pevzner, P.A., Pevzner, P.A., Tang, H., Tesler, G.: De novo repeat classification and fragment assembly. Genome research **14**(9), 1786–1796 (2004)
18. Zhi, D., Raphael, B.J., Price, A.L., Tang, H., Pevzner, P.A.: Identifying repeat domains in large genomes. Genome biology **7**(1), 7 (2006)
19. Saha, S., Bridges, S., Magbanua, Z.V., Peterson, D.G.: Empirical comparison of ab initio repeat finding programs. Nucleic acids research **36**(7), 2284–2294 (2008)
20. Zheng, J., Lonardi, S.: Discovery of Repetitive Patterns in DNA with Accurate Boundaries. In: Fifth IEEE

Symposium on Bioinformatics and Bioengineering (BIBE'05), pp. 105–112. IEEE, ??? (2005)
21. He, D.: Using suffix tree to discover complex repetitive patterns in DNA sequences. Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference **1**, 3474–3477 (2006)
22. Huo, H., Wang, X., Stojkovic, V.: An Adaptive Suffix Tree Based Algorithm for Repeats Recognition in a DNA Sequence. In: 2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, pp. 181–184. IEEE, ??? (2009)
23. Li, M., Ma, B., Kisman, D., Tromp, J.: Patternhunter II: highly sensitive and fast homology search. Journal of bioinformatics and computational biology **2**(3), 417–439 (2004)
24. Figueroa, N.: RAIDER: Rapid Ab Initio Detection of Elementary Repeats. PhD thesis, Oxford (2013)
25. Figueroa, N., Liu, X., Wang, J., Karro, J.: RAIDER: Rapid Ab Initio Detection of Elementary Repeats. In: Advances in Bioinformatics and Computational Biology, pp. 170–180. Springer, Cham (2013)
26. Jurka, J., Kapitonov, V.V., Pavlicek, A., Klonowski, P., Kohany, O., Walichiewicz, J.: Repbase Update, a database of eukaryotic repetitive elements. Cytogenetic and genome research **110**(1-4), 462–467 (2005)
27. Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic acids research **25**(17), 3389 (1997)