Journal of Bioinformatics and Computational Biology
Vol. 12, No. 1 (2014) 1350018 (24 pages)

DOI: 10.1142/S0219720013500182

Imperial College Press
www.icpress.co.uk

EXHAUSTIVE COMPUTATION OF EXACT DUPLICATIONS VIA *SUPER* AND *NON-NESTED LOCAL* MAXIMAL REPEATS

EDDY TAILLEFER* and JONATHAN MILLER†

Physics and Biology Unit, Okinawa Institute of Science and Technology
1919-1 Tancha, Onna-son, Kunigami-gun 904-0412, Japan
**etaillefer@oist.jp*
†*jm@oist.jp*

Received 15 May 2013
Revised 9 October 2013
Accepted 15 October 2013
Published 18 December 2013

We propose and implement a method to obtain all duplicated sequences (repeats) from a chromosome or whole genome. Unlike existing approaches our method makes it possible to simultaneously identify and classify repeats into super, local, and non-nested local maximal repeats. Computation verification demonstrates that maximal repeats for a genome of several gigabases can be identified in a reasonable time, enabling us to identified these maximal repeats for any sequenced genome. The algorithm used for the identification relies on enhanced suffix array data structure to achieve practical space and time efficiency, to identify and classify the maximal repeats, and to perform further post-processing on the identified duplicated sequences. The simplicity and effectiveness of the implementation makes the method readily extendible to more sophisticated computations. Maxmers can be exhaustively accounted for in few minutes for genome sequences of dozen megabases in length and in less than a day or two for genome sequences of few gigabases in length. One application of duplicated sequence identification is to the study of duplicated sequence length distributions, which our found to exhibit for large lengths a persistent power-law behavior. Variation of estimated exponents of this power law are studied among different species and successive assembly release versions of the same species. This makes the characterization of the power-law regime of sequenced genomes *via* maximal repeats identification and classification, an important task for the derivation of models that would help us to elucidate sequence duplication and genome evolution.

Keywords: Genome sequence; maximal repeat; overlapping; suffix array; length distribution; power-law.

1. Introduction

The repetitive structure of genomic DNA plays a key role in genome biology: Sequence duplication is a primary contributor to genome evolution and a potential source of new genes[1–3]; duplication is often accompanied by genomic rearrangement[4];

human disease can be associated with duplications.[5] For these reasons intensive experimental and computational effort has been invested over the years in identifying and characterizing genomic repeats. Classifications of repetitive sequence structure include tandem repeat, simple or complex repeat, interspersed repeat, transposable element, and segmental duplication. Segmental duplication has been defined in a number of ways; for example, two or more segments of DNA[6] longer than 1,000 base-pairs with at least 90% identity.[7] Duplicate frequency distributions within DNA sequences have been applied to elucidate specific repetitive features of the genome.[8–11] The latter studies focused on the frequency distribution of k-mers (exact duplicates of length k) and revealed features of the Zipf plots[12] shared by genomic non-coding regions and natural language texts[8]; demonstrated that frequency distributions of genomic k-mer sequences exhibit a Pareto-lognormal-like distribution[9]; examined species-dependent differences in modalities of k-mer frequency distributions[10]; and classified 40-mers within and across chromosomes based on the character of their frequency distributions.[11] Here we adopt a model-independent approach to identifying repeats: We study exact duplicates of all lengths, irrespective of any classification; our characterization can be productively applied also to natural language texts or computer code.

Computational analysis of sequence duplication is further motivated by recent observation of a power-law regime in the length distribution of sequences conserved *among* multiple divergent genomes[13,14] — sequences of which at least one copy can be found in each of two or more genomes. Those *inter*-genomic computations were performed by hash-table based search[15] and whole-genome alignment[13,15] using the BLAST[16,17] alignment tool BLASTZ[18] (supplanted by LASTZ[19]). Whole-genome alignment has recently demonstrated an *intra*-genome power-law regime in the length distribution of exactly duplicated sequence *within* a single genome[20]; however, practical whole-genome alignment methods are heuristic and time-consuming; hash-table based intersection unwieldy and can make it difficult to recover sequence features such as local maximality (see below) and copy number.

Although a number of different methods have been deployed for *de novo* identification of repeats in genome sequence,[21–24] they are subject to limitations on the types of duplication they can identify, input sequence length, information output, speed, and efficiency of memory usage.[25,26] Software such as MUMmer,[27] REPuter,[28] and Vmatch[29] has been developed to efficiently identify repeats in large genome sequences. MUMmer relies on suffix *tree*[30] data structure, whereas REPuter and Vmatch rely on suffix *array*[31,32] data structure. REPuter identifies all exact duplicates, but its limits on input sequence length constrain its application; furthermore, its output format places heavy demands on memory — in order of n^2 for a n-size input sequence — and requires post-processing of the output. REPuter has been superseded by Vmatch, which can identify both exact duplicates and near-exact duplicates, but relies on disk storage rather than memory to minimize intermediate computations, yielding prohibitive access times. MUMmer's suffix trees make relatively inefficient use of memory compared to suffix arrays[33] that employ a contiguous

table to achieve compact and efficient memory usage, allowing parallelization or vectorization of the computation. Suffix trees employ pointer-based data structure that creates sparsity in memory and requires more memory to store the same amount of information. A suffix tree also requires more memory than a suffix array merely to store its minimal structure.[33]

We propose and implement here a method to obtain all duplicated sequences in a chromosome or whole-genome. All exactly duplicated sequences are identified and classified. Duplicated sequences are naturally divided into two disjoint classes, *super* maximal repeats and *local* maximal repeats, where a maximal repeat (or *maxmer*) is defined as a *set* of identical contiguous chromosomal sub-sequences that is not a subset of any other maxmer. We also introduce a sub-class of local maxmer called *non-nested* local maxmer that allows us to represent the repetitive sub-sequences of a given sequence with a minimum number of maxmers. Enhanced suffix arrays (ESA) are employed to efficiently compute maximal repeats. Unlike MUMmer or Vmatch our method both classifies maximal repeats and computes local maximal repeats. Computation verification demonstrates that maximal repeat identification and classification for a genome of several gigabases can be accomplished in a reasonable time for any sequenced genome.

The paper is organized as follows. Section 2 introduces *super*, *local*, and *non-nested local*, and presents an algorithm for their identification and classification. Section 3 demonstrates our algorithms by computing maxmer length distributions of chromosome and whole-genome sequences (WGS) from several species.

2. Identification of Super, Local, and Non-nested Local Maxmers

This section describes maximal repeats (maxmers) and their classification into super, local, and non-nested local maxmers. A sequence S_α is a length-n string that takes its values over an alphabet Σ (ex, $\Sigma = \{\mathtt{A}, \mathtt{T}, \mathtt{G}, \mathtt{C}\}$). See detailed definition in Appendix A.

2.1. *Super, local, and non-nested local maxmers*

The distinction among maxmers is motivated by the need to account in a model-independent way for copies of subsequences (see Appendix A for definitions and properties of maxmers). Super, local, and non-nested local maxmers enable decomposition of the set of maxmers into subsets. We define maxmer subsets as follows:

- An occurrence is a length-ℓ sub-sequence in S_α.
- A repeat is a set of identical occurrences, each with a different location in S_α.
- A maximal repeat (**maxmer**) is a repeat that includes a pair of occurrences, no extensions of which themselves yield a repeat.
- A super maximal repeat (**super maxmer**) is a maxmer that never occurs as a subsequence of another maxmer.
- A local maximal repeat (**local maxmer**) is a maxmer that is not a super maxmer.

A **nested occurrence** is an occurrence of a local maxmer that is contained in a longer maxmer. We may then define the following subset of maxmers:

- A **non-nested local maxmer** is a local maxmer with at least one occurrence that is not nested.

For example, in the sequence

NLAREPLNOREPTFCGIREPTLSIG

the sub-sequence REPT is a super maxmer, whereas the sub-sequence REP is a local maxmer. In addition, because the first occurrence of the sub-sequence REP is not nested into any longer maxmer, REP is also a non-nested local maxmer.

Note that local and super split the set of maxmers in S_α into two sets, whereas the non-nested local maxmer set is a subset of local maxmer. As shown in Appendix A.4 there exist subsets of any local maxmer each of whose elements can be extended to elements of a longer super maxmer. Therefore, either a local maxmer is non-nested or all its occurrences are nested into longer super maxmers. It follows that the set of super maxmers together with the set of non-nested local maxmers (over all possible maxmer lengths) constitute a compact representation of the maxmers.

What we call here a "non-nested local maxmer" has previously been described in the literature as a "near-supermaximal repeats" in, Ref. 34 and "largest maximal repeats" in Ref. 35 with different but equivalent definitions; however, Ref. 34 does not provide an explicit algorithm to compute non-nested local maxmers and the method presented in Ref. 35 does not address the relationship of non-nested local maxmers to local and super maxmers and their classification. The algorithm proposed in Ref. 35 relies on a suffix tree data structure instead of a suffix array. Here, maxmers are identified and classified using an *enhanced suffix array* data structure (see Appendix A.5 for a more formal description). We propose an algorithm (see Fig. A.2 of Appendix A.6) that for any given sequence S_α allows us to identify all maxmers; obtain for each maxmer its length and number of occurrences; and classify each maxmer as local, non-nested local, or super.

3. Application to DNA Sequence Analysis

Deoxyribonucleic acid (DNA) is a directed unbranched polymer composed of chemical building blocks, the four nucleotides (or bases) A, G, C, and T, covalently linked into a string. Two oppositely-directed polymers pair "complementary" bases A with T and G with C to yield a double-stranded DNA molecule in which each strand is the reverse-complement of the other. We define a chromosome as a length-n sequence S_α over the set of symbols $\Sigma = \{\mathtt{A}, \mathtt{T}, \mathtt{G}, \mathtt{C}, \mathtt{N}\}$, where N represents a location that has not yet been assigned a base. Most chromosomes occur naturally in this symmetric double-stranded form. To account for this symmetry, we append to the single-stranded sequence its reverse complement, with a separator symbol inserted between

the two. A *whole-genome* is composed of a set of chromosomes and is represented by the concatenation of chromosomes with separator symbols inserted between them.

The primary goal of our algorithm is the identification and classification of maxmers. Although in this paper, we study the length distribution of maxmers as a direct application of the algorithm, its range of application is considerably broader. For example, the content of each maxmer can also be used in applications such as annotation. In the following, we focus on computation of the length distribution of maxmers as a quantitative tool for characterizing repetitive genomic sub-sequences.[8,36,37]

3.1. *Computation of maxmer length distributions for genome sequences*

Obtaining the length distribution requires computing the number of maxmers while accounting for maxmer subset type.

The length distribution computation takes as input the double-strand genome sequence. A genome sequence is usually available as a formatted text file that contains only one strand of n bases, so that the length of the input string for our computation becomes $N = |\mathrm{S}_\alpha| = 2n + 1$. Because of the symmetry of the double-stranded DNA molecule, we expect that

- if a repeat R is identified within the forward strand, then a corresponding repeat $\underline{R}$ will be identified within the reverse complemented strand, where the sequence of $\underline{R}$ is the reverse complement of R.
- inverted maxmers will be identified as maxmers containing at least one occurrence on each strand.

The length distribution refers to the number $F(\ell)$ of maxmers (super and/or non-nested local) of each length ℓ, with $\ell \geq \ell_{\min}$, where $\ell_{\min}$ is the minimum sequence length of the maxmers that would be identified. The total number of maxmers M_{eff} is then given by

$$M_{\text{eff}} = \sum_{\ell \geq \ell_{\min}} F(\ell).$$

The number of maxmers discarded because all their occurrences are nested into longer maxmers, M_{nest} is also counted for comparison with M_{eff}.

3.2. *Length distributions of natural genome sequences*

To demonstrate our algorithm, we computed the length distributions of 682 genome sequences (chromosome and whole-genome) from 45 eukaryotic species.

Representative length distributions chosen from among 14 species are plotted in Figs. 1–5 (plots of length distributions of all analyzed genome sequences can be found in the supporting information). Rice chromosome sequences were obtained from the National Center for Biotechnology Information (NCBI) public databases; all the other sequences were obtained from the University of California, Santa Cruz (UCSC)

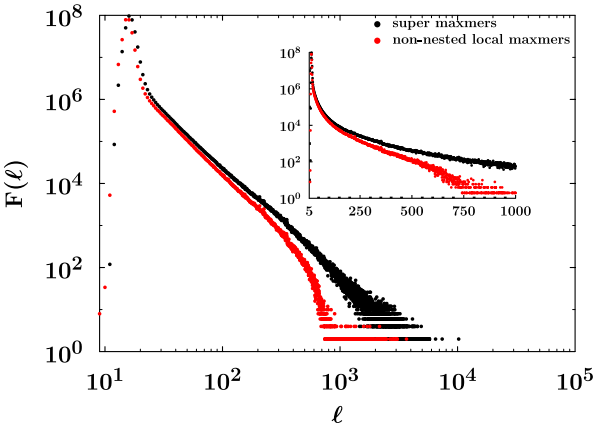


Fig. 1. Plots of super and non-nested local maxmer length distributions for frog whole-genome.

genome browser. Each figure exhibits the length distribution on log–log axes in the main frame, and semilog axes in the inset.

For frog whole genome, Fig. 1 shows the length distributions of super maxmers and non-nested local maxmers.

The power-law like behavior exhibited by the plots in Fig. 1 was first obtained for a small number of chromosomes by self-alignment.[20] Sequence alignment is a

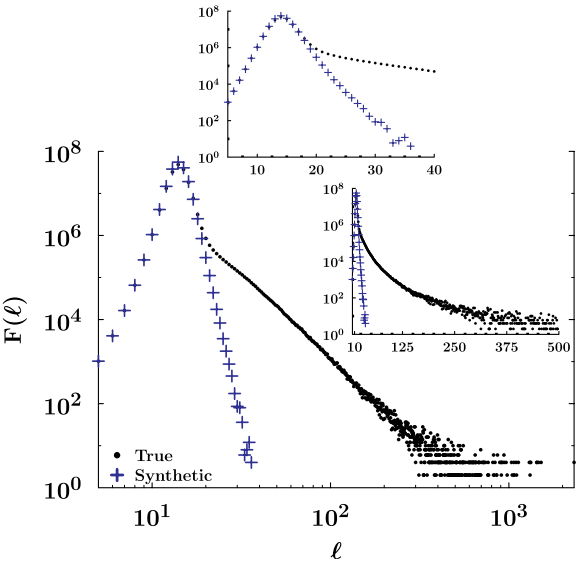


Fig. 2. Length distributions (super + non-nested local maxmers) for the horse chromosome 3 and a synthetic sequence of the same length generated by a first-order Markov model with transition probabilities and base composition obtained from horse chromosome 3 ($\ell_{\min} = 5$).

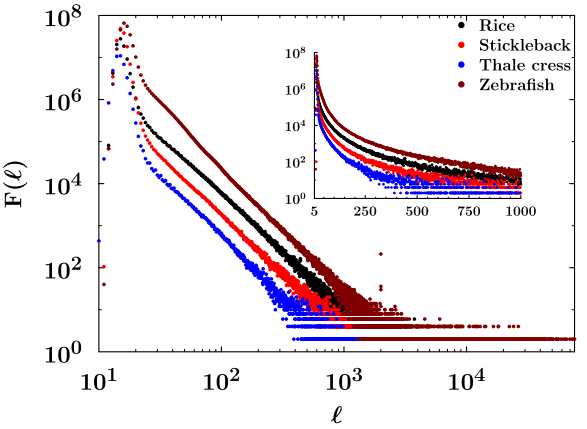


Fig. 3. Length distribution of super maxmers, for rice, stickleback, thale cress, and zebrafish whole-genome sequences $\ell_{\min} = 5$.

heuristic process whose outcome depends on the method applied and parameters chosen. Only once the power-law behavior was reproduced as described here, wherein the distribution computed is defined independently of any algorithm, could it be confidently asserted that the power-law behavior was not an artifact of the alignment method. Power-law behavior is often clearly elucidated by the maxmer length distribution; for example, as shown in Fig. 2, length distributions of horse chromosome 3 and synthetic sequence exhibit vastly different behavior for large ℓ, indicating that the probability of duplicating long maxmers by chance within an independent site model (our null model) is remote. Deviation from a randomly-generated synthetic sequences observable for nearly all chromosomes we studied (see the supporting

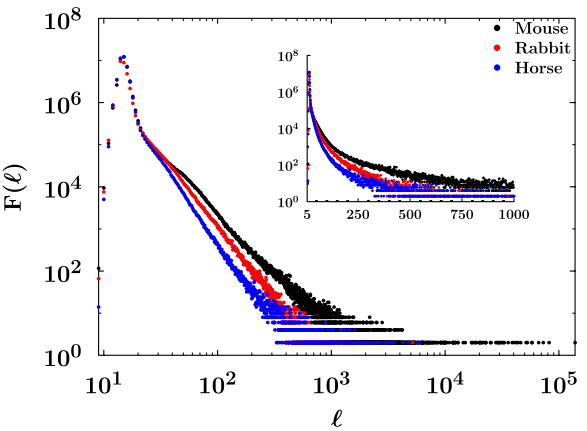


Fig. 4. Length distribution of super maxmers for mouse, rabbit and horse chromosome X sequences $\ell_{\min} = 5$.

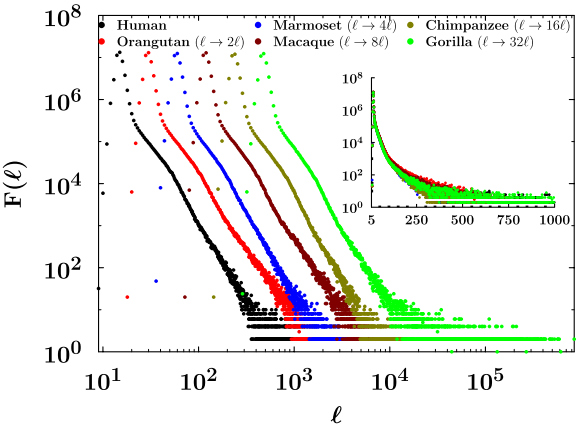


Fig. 5. Length distribution of super maxmers for human, orangutan, marmoset, macaque, chimpanzee, and gorilla chromosome X sequences. Each distribution curve is plotted with a translation along the x-axis, indicated by $\ell \to k\ell$, excepted for the human curve which is taken as the translation reference ($k = 1$) $\ell_{\min} = 5$.

information for a comprehensive set of plots). Our maxmer identification scheme enables a systematic, quantitative measurement of this persistent feature. For example, although the amplitudes of the curves often undergo multifold enhancement when bases are counted modulo the most frequent base substitution event,[38] C↔T/G↔A, it turns out that the power-law behavior remains unchanged; this observation yields an important constraint on evolutionary dynamics.[20,39] Additionally, as much as half of each natural genome occurs as so-called "repetitive sequence[a];" length distributions of chromosomes from which these repetitive sequences have been excised (RM for repeat-masking) nevertheless exhibit a power-law. (see the plots for RM, 2b, and RM-2b in the supporting information.)

Table 1 summarizes maxmer counting statistics for the plotted length distribution. For each genome sequence only bases unambiguously assigned a nucleotide $\Sigma = \{\mathtt{A}, \mathtt{G}, \mathtt{C}, \mathtt{T}\}$ were tabulated. Maxmers containing a sequence with a symbol not included in Σ (for example N) were discarded. Table 1 shows that about half of the maxmers (35% to 51%, last column) are not counted in M_{eff} above.

Length distributions from distinct species and even chromosomes within a single species can exhibit a variety of different shapes (Figs. 3–5). To better appreciate the variation of the slope, maximum-likelihood estimation α_{MLE} of the exponent parameter α of a power-law model $P_{\text{PL}}(\ell) = (\alpha - 1)(\ell/\ell_{\min})^{-\alpha}/\ell_{\min}$ is applied to the power-law regime (large length-ℓ regime, see Fig. 2) of the maxmer distribution, for several publicly available genome sequences.[40] Figure 6 shows that estimated exponents differ among species, with $-\alpha_{\text{MLE}}$ varying between -4.4 and -1.9. For

[a]sequences that include Alu, SINE, LINE, transposon, tandem repeat, and so on.

Table 1. Statistics summary of plotted length distribution; `chr.`: chromosome `WG`: whole-genome.

Species name [chr. number]	Version	S_{cov} [a]	Sequence size[b] (bp.)	ℓ_{max} [c]	$\%_{mcov}$ [d]	$\%_{ncov}$ [e]	M_{eff}	M_{nest}
X. tropicalis [WG]	xenTro2	7.65	1,359,400,017	10,223	30.1	89.79	603,940,204	558,042,142
E. caballus [3]	equCab2	6.8	118,104,910	1,992	6.7	98.85	71,742,115	52,897,342
O. sativa [WG]	IRGSP3.0	6	370,733,456	56,270	32.7	99.98	166,376,374	146,660,553
G. aculeatus [WG]	gasAcu1	6	391,398,261	6,154	9.1	97.65	238,003,144	164,476,014
A. thaliana [WG]	ath1	7–10	118,997,677	44,000	12.5	99.84	68,461,660	50,318,269
D. rerio [WG]	danRer5	5.7	1,273,638,207	76,102	37.5	99.73	522,926,640	541,811,249
H. sapiens [X]	hg19	8–10	151,100,560	119,819	20.5	97.31	77,254,422	68,556,270
P. abelii [X]	ponAbe2	6	148,146,383	1,599	20.3	94.85	76,171,360	67,441,885
C. jacchus [X]	calJac3	6	131,921,481	1,286	16.5	92.87	70,982,672	60,454,717
M. mulatta [X]	rheMac2	5.1	141,500,845	20,561	17.4	91.92	74,172,899	64,629,691
P. troglodytes [X]	panTro2	6	130,929,871	1,047	15.6	84.27	69,607,360	59,777,854
G. gorilla [X]	gorGor3	2.1	143,041,859	26,508	18.4	92.60	73,480,081	64,318,565
M. musculus [X]	mm9	7	162,080,892	138,651	32.1	97.26	70,876,081	65,575,376
O. cuniculus [X]	oryCun2	7.48	106,137,172	5,196	20.1	95.02	55,006,117	48,310,304
E. caballus [X]	equCab2	6.8	121,614,486	5,952	12.5	97.99	68,724,998	55,007,061

[a]Reported assembly sequence coverage in number of fold (×).
[b]Only for bases in $\Sigma = \{\mathtt{A}, \mathtt{G}, \mathtt{C}, \mathtt{T}\}$.
[c]Length of the largest maxmer.
[d]Maxmer coverage: percentage of the genome sequence covered by maxmers of length $\ell \geq 30$.
[e]Nucleotide coverage: percentage of bases that are not equal to `N` for a sequence composed of symbols in $\Sigma = \{\mathtt{A}, \mathtt{G}, \mathtt{C}, \mathtt{T}, \mathtt{N}\}$. Where, `N` represents loci for which a site has not yet been assigned a base. Nucleotide coverage is an indicator of the level of completion of the sequence assembly, and therefore an indicator of assembly quality.

different versions of the genome assembly of a given species, estimated exponents show similar or converging values (see zebrafish and mouse in Fig. 6).

The genome sequence of a given species is often available in successive releases of the assembly. Genome assembly is the stage of a sequencing project where DNA fragments of overlapping sequence are patched together into a reconstruction of the presumed natural chromosomal sequence. Assemblies may be revised subsequent to error correction, when new sequence reads become available, or when new assembly methods are developed.[41–43] The genome sequence quality and accuracy is generally expected to improve with every new assembly release, although there have been exceptions. We have found that successive assembly versions involving minor (say, few bases or chunks of sequences representing less than 1% of the genome size) changes in the genome sequence do not exhibit noticeable differences in the length distributions; however, as shown in Fig. 8 for the horse chromosome X and in Fig. 9 for the purple sea urchin whole genome, important changes in the genome assembly (e.g. addition or deletion of a large overall fraction of bases) can have significant impact on the distribution. Table 2 gives the genome assembly information for those two species.

The length distributions of the *Equus caballus* chromosome X in Fig. 8 show a clear variation of length distribution with the genome assembly. While length distributions recapitulate similar slopes from one assembly version to another, the

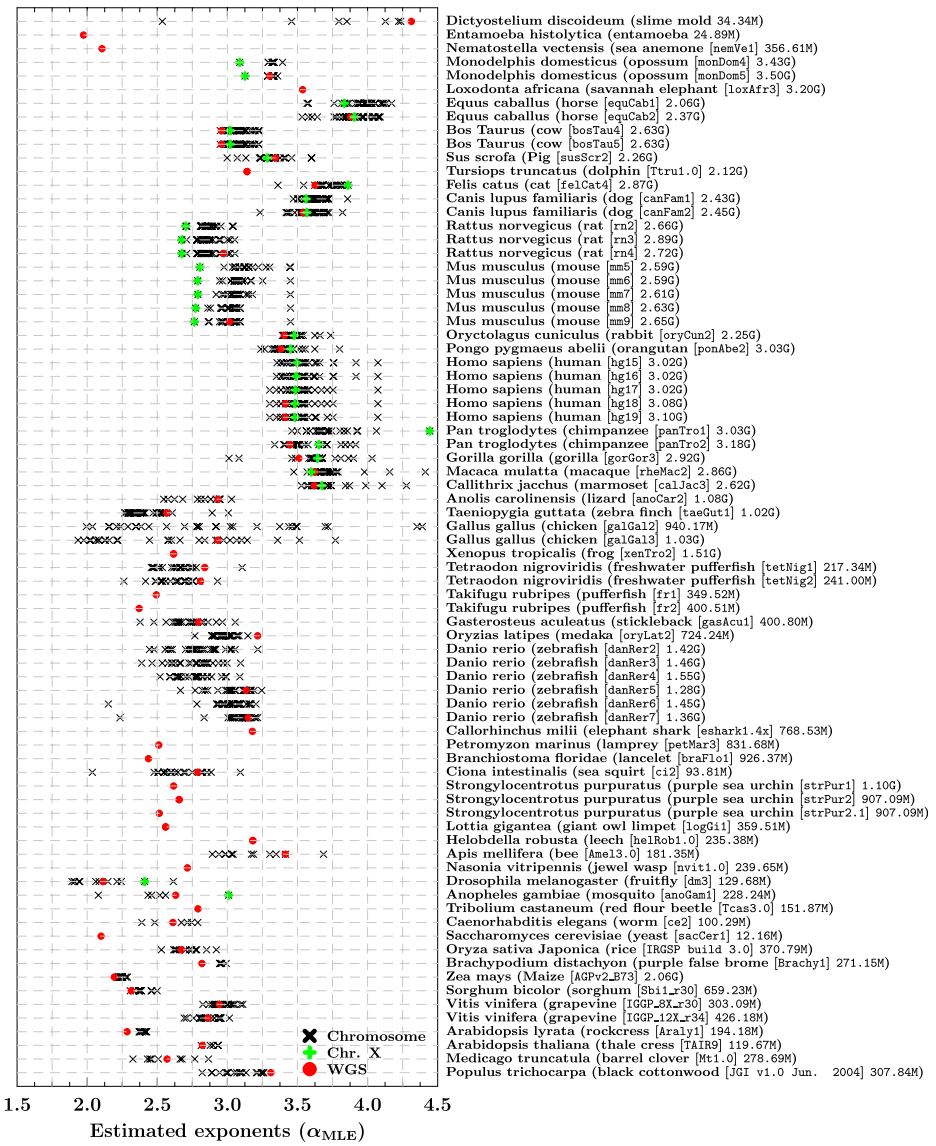


Fig. 6. Maximum-likelihood estimation results of exponent parameter α of a power-law model, obtained from the length distribution of super maxmers for several chromosome and WGS ($\ell_{\min} = 5$). The sex chromosome (Chr. X) of some the species are highlighted with green plus marks. The species names in the right vertical coordinate read: binomial name (common name [`assembly version`] `genome size in base-pair`). The specie names appear in the same order that they appear in the phylogenetic cladogram (Fig. 7).

power-law regime exhibits a significant increase in the number of maxmers, where the distribution of `equCab2` appears to be a translation of `equCab1` on the $F(\ell)$ axis. The increase in number of maxmers is fully accounted by for the increase in genome size from version `equCab1` to `equCab2`, as shown in Table 2. The version `equCab1` was

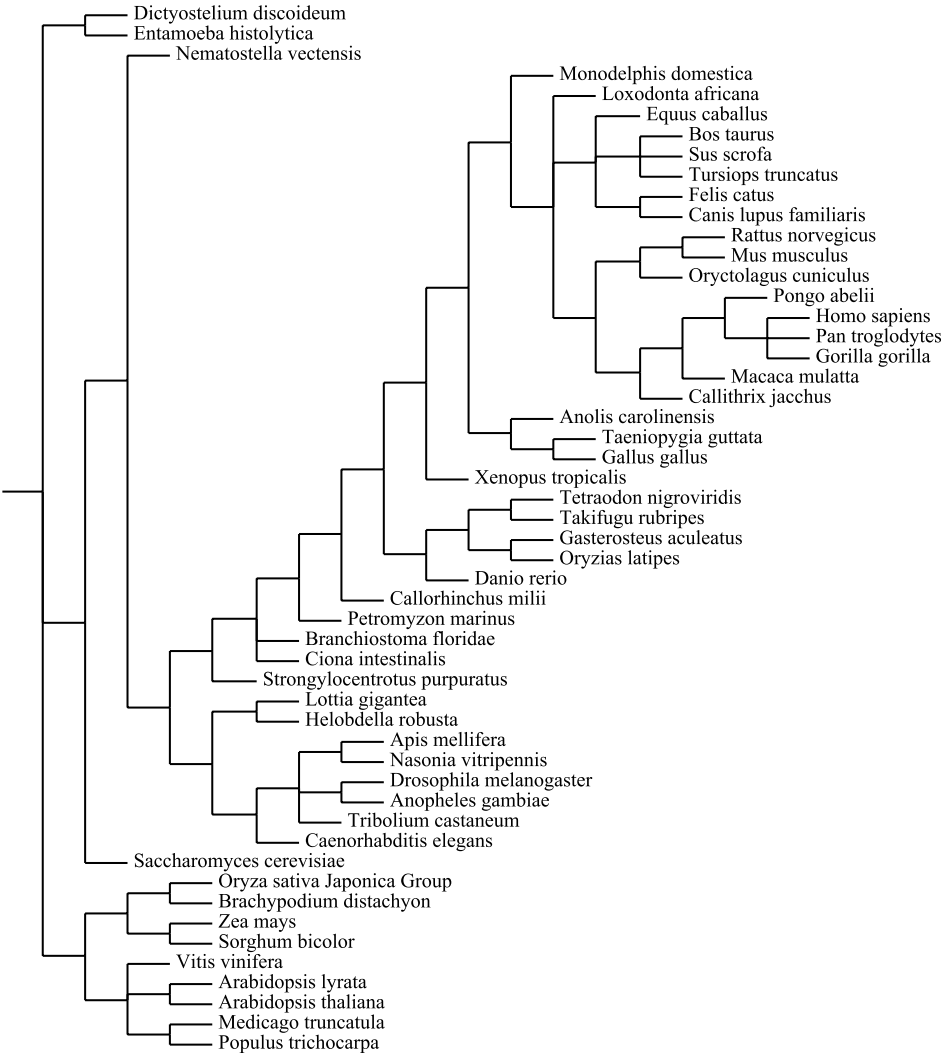


Fig. 7. Cladogram of the investigated species. The phylogenetic relation between species are based on the classification in the NCBI taxonomy database (http://www.ncbi.nlm.nih.gov/Taxonomy/CommonTree/wwwcmt.cgi).

the first public release of the horse genome by Broad Institute; it was closely followed by the assembly version `equCab2` (http://www.avma.org/onlnews/javma/apr07/070401n.asp).

Figure 9 illustrates the length distributions of *Strongylocentrotus purpuratus* whole-genome sequence. The length distributions superpose for maxmer length up to around $\ell = 400$ bases, but deviate in the tail. The sharp dropoff in the tail of the length distribution of `strPur1` starting in the neighborhood of $\ell = 1,000$ may originate in the large number of undetermined bases (symbol `N`; see Table 2) that cut

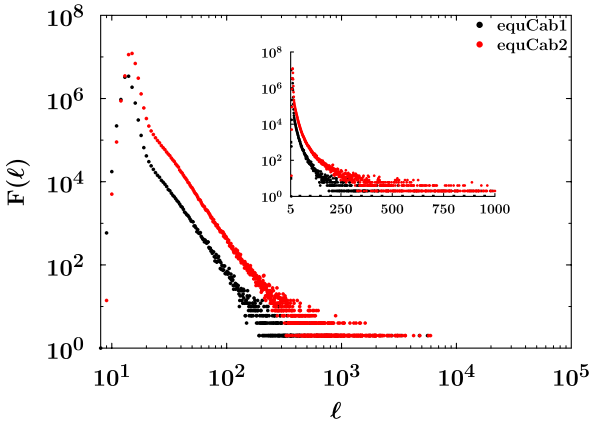
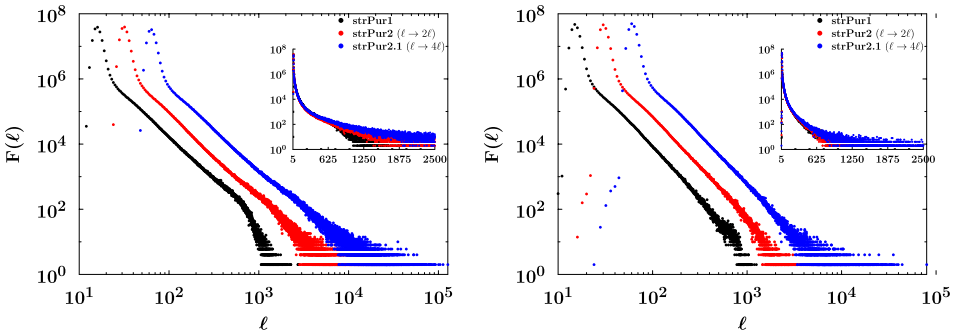


Fig. 8. Variation of length distribution function of the super maxmers to the assembly version for the horse chromosome X $\ell_{\min} = 5$.



(a) Urchin (super maxmers)

(b) Urchin (non-nested local maxmer)

Fig. 9. Variation of length distribution with assembly version for the purple sea urchin whole-genome sequence $\ell_{\min} = 5$. Curves for `strPur2` and `strPur2.1` are translated to the right.

off contiguous runs of identical bases at around $\ell = 1,000$.[40] This dropoff disappears in subsequent assemblies with improved sequence coverage, and the tail of the length distribution becomes larger and heavier. According to (https://www.hgsc.bcm.edu/content/sea-urchin-genome-project), version `strPur1` was heavily contaminated.

Table 2. Genome assembly information.

Name	Release date	$\%_{\text{ncov}}$	Size[f] [bp.]	Source
`equCab1`	Jan. 2007	98.57	31, 349, 299	Broad
`equCab2`	Sep. 2007	97.99	124, 114, 077	Institute
`strPur1` (Spur_0.5)	Apr. 2005	76.21	1, 096, 072, 416	BCM
`strPur2` (Spur_2.0)	Jun. 2006	89.29	907, 085, 737	HGSC
`strPur2` (Spur_2.1)	Sep. 2006	99.99	907, 094, 382	

[f] The size includes all bases in $\Sigma = \{\mathtt{A}, \mathtt{G}, \mathtt{C}, \mathtt{T}, \mathtt{N}\}$.

In version `strPur2`, several "contigs of contaminating (non-*S. purpuratus*) sequence and overlapping (second haplotype contigs)" were removed. In version `strPur2.1`, although most of the contamination is believed to have been purged, errors such as "misassemblies of repeat sequences," "collapses of repeat regions," and "artificial duplications in polymorphic regions," may remain. So, contamination may explain the cut-off in the tail of the length distribution, the large genome size, and the low nucleotide coverage ($\%_{\text{ncov}}$) in `strPur1`.

Figures 6, 8, and 9 suggest that maxmer length distributions reflect assembly quality. Furthermore, investigation of correlation between assembly quality (nucleotide coverage $\%_{\text{ncov}}$) and maxmer length distribution parameters for $\ell > 30$, in Fig. 10, shows that large maxmer length (ℓ_{max}), high maxmer coverage ($\%_{\text{mcov}}$),

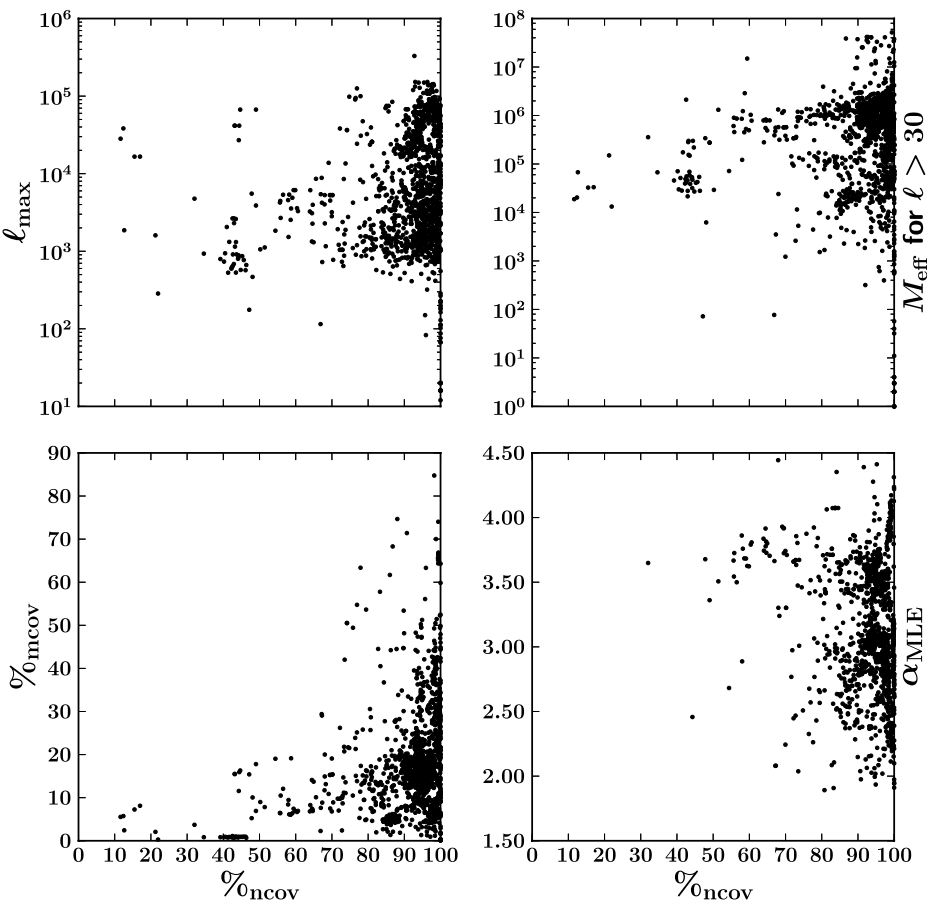


Fig. 10. Correlation between nucleotide coverage ($\%_{\text{ncov}}$) and parameters of maxmer length distribution ($F(\ell)$) in the power-law regime ($\ell \geq 30.$), where $F(\ell)$ parameters are ℓ_{max}, maxmer coverage ($\%_{\text{mcov}}$), number of maxmers (M_{eff}) and α_{MLE}.

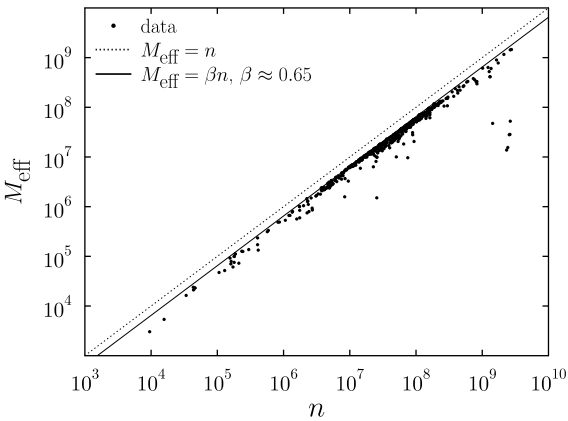


Fig. 11. Total number of maxmers M_{eff}, versus length of genome sequence n, for 682 different genome sequences over 45 species.

large number of maxmers in the power-law regime (M_{eff}), and clusterization of estimated α_{MLE}, agree with a high assembly quality.

3.3. *Computational complexity*

The computational cost of our proposed algorithm was investigated for practical use, within a large range of available assembled genome species. The total number of maxmers M_{eff} is plotted against the length of the genome sequence n in Fig. 11. This figure allows us to assess the memory consumption of our algorithm. If w_{s} is the size of an integer in bytes, and n the length of the genome sequence in bases, the enhanced suffix array data structure is of size $2w_{\text{s}}Qn + 1$ bytes, where Q is a constant integer representing the number of tables needed for storing the structure in memory. Each identified maxmer features can be successively stored in table or discarded once counted; however, in the former case the size of this table is practically bounded by $w_{\text{s}}M_{\text{eff}} \approx w_{\text{s}}\beta n$, where β is a fixed scalar value. Finally, we estimate the size of the structure required for retrieval of local minimum ℓ-intervals; this size is bounded by $3w_{\text{s}}(2n + 1)$ (see Ref. 44). The size of each table is proportional to n so that the overall space consumption of the proposed algorithm is linear (i.e. $O(n)$).

The computation times for construction of the suffix array is of order $O(n \log(n))$ and the computations of the longest common prefix and the retrieval of all ℓ-intervals are each of order $O(n)$.[33] In our proposed algorithm (Fig. A.2 of Appendix A.6) additional computation time is required for the identification and classification of each maxmer and is proportional to the number of occurrences of this maxmer. In the worst case the number of occurrences of a given maxmer may be approximated by the integer value k at the maximum of a Poisson distribution, $p_\lambda(k) = \lambda^k e^{-\lambda}/(k!)$, where the interval λ is taken to be the sequence size n[45,46]; however, since this quantity can only be determined empirically, the overall computation time order is

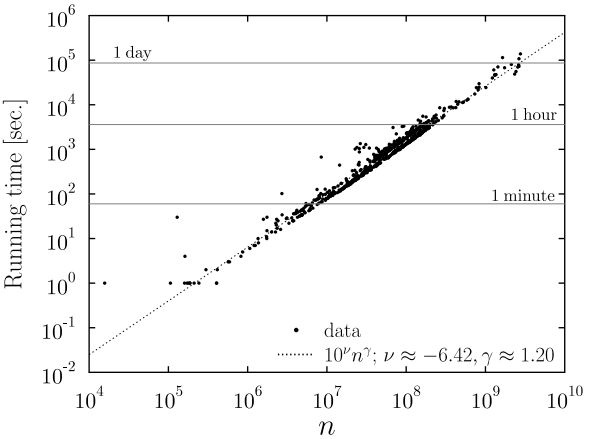


Fig. 12. Run time of length distribution computation versus genome sequence length n for 682 different sequences from 45 species. Dots indicated run time of each computation. The dotted line is a linear regression performed in log–log cordinates.

estimated based on real computations. The full run time of the length distribution computation is plotted against the genome sequence size in Fig. 12. The computations were performed using a SGI® Altix® 4,700 server with 1.66 GHz Itanium2 CPU[b] and one terabyte of memory. According to Fig. 12, the effective overall computation time is of order $O(n^{1.2})$, which is not linear time, but much faster than quadratic time. Length distributions are computed in few minutes for genome sequences of lengths around $n = 24$ megabases (using around 1.5 gigabytes of memory) and in less than a day or two for the largest genome sequences, around $n = 2.8$ gigabases (using around 250 gigabytes of memory).

4. Conclusion

We proposed and implemented the identification of super, local, and non-nested local maxmers within a genome sequence, and compute the duplicated sequence length distribution of those maxmers. The algorithm proposed for the identification and classification of the maxmers achieves practical space and time efficiency by using enhanced suffix data structure. The simplicity of the implementation makes the method extendible to more sophisticated computations. Computation verification shows that the length distribution for a genome of several gigabase pairs can be computed in a reasonable time, enabling us to compute these distributions for all sequenced genomes. Empirical observation of computed distribution of duplicated sequence lengths in natural genomes, shows that the distribution is scale-free and follows a consistent power-law behavior for large range of lengths, and across a

[b]This architecture has known memory access speed limitation. We could reproduce some of these computations on a one terabyte Dell® PowerEdge R910 with 1.87 GHz Xeon CPU, and observed a twofold increase in speed.

variety of species. Comparison of maxmer length distributions among assembly release versions suggests a potential application in the objective measure of genome assembly quality and progression. It is hoped that the general notion of maxmer proposed here can contribute quantitatively to our understanding of sequence duplication and genome evolution in general using sequence data alone, and without pre-imposed conceptions about what is or is not biologically relevant.

Appendix A

We define local, super and non-nested local maxmers in a more formal way. Then, we describe an algorithm for maxmer identification and classification that can directly be derived from those definitions. The content of this appendix was partly presented in Ref. 47.

A.1. *Basic definition*

Let the *sequence* S_α be a string of symbols, and let $n = |\mathrm{S}_\alpha|$ be the length of the string. $\mathrm{S}_\alpha[k]$, $\forall\, k = 1, 2, \ldots, n$, denotes the symbol at position k in the string, with $\mathrm{S}_\alpha[k] \in \Sigma$, where Σ is the alphabet (e.g. $\Sigma = \{\mathtt{A}, \mathtt{T}, \mathtt{G}, \mathtt{C}, \mathtt{N}\}$).

Let $@ \notin \Sigma$ ($\$ \notin \Sigma$) be a separator (terminator) lexicographically greater than any symbol in Σ. A string $\mathrm{S}_\alpha\$$ will be referred to as a *terminated* string.

A *chromosome* sequence is represented by the string $\mathrm{S}_\beta = \mathrm{S}_\alpha @ \underline{\mathrm{S}_\alpha} \$$ on the symbol set $\Sigma = \{\mathtt{A}, \mathtt{T}, \mathtt{G}, \mathtt{C}, \mathtt{N}\}$, constructed from the initial sequence S_α, where $\underline{\mathrm{S}_\alpha}$ is the reverse complement of S_α.

A *sub-sequence* of S_α of length ℓ is defined as a substring $\mathrm{S}_\alpha[i]\mathrm{S}_\alpha[i+1]\cdots \mathrm{S}_\alpha[i+\ell-1]$, where $\ell \geq 1$, and $1 \leq i < i + \ell - 1 \leq n$.

A.2. *Occurrence and repeat*

An **occurrence**, $C_k = \mathrm{S}_\alpha[i_k]\mathrm{S}_\alpha[i_k+1]\cdots\mathrm{S}_\alpha[i_k+\ell-1]$, is defined as a length-ℓ subsequence of S_α where $\ell \leq |\mathrm{S}_\alpha|$; We call i_k an **occurrence index.** Given S_α, a **repeat** is defined as set of identical symbol sub-sequences $R = \{C_1, C_2, \ldots, C_p\}$ containing $p \geq 2$ occurrences of length ℓ. A repeat is uniquely specified by length ℓ, number of occurrences p, and a list of occurrence indices, $i_1, i_2, \ldots, i_p$.

A.3. *Context and extendibility*

The **left-context** (**right-context**) of an occurrence of length ℓ indexed by i_k is defined as $\mathrm{S}_\alpha[i_k-1](\mathrm{S}_\alpha[i_k+\ell])$. An occurrence of a given repeat R is said to be **left-extendible** (**right-extendible**) if there exists in R another element having identical left-context (right-context). An occurrence is said to be **extendible** if it is either left-extendible or right-extendible; and to be **inextendible** if neither left-extendible nor right-extendible.

A.4. *Maximal repeats* (*maxmers*)

We define super and local, non-nested local, and maximal repeats, and show that the set of non-nested local maximal repeats added to the set of super maximal repeats constitutes a compact set of the maxmers in S_α.

A repeat R is **maximal** if there exists among the p occurrences of the repeat at least two occurrences whose left- and right-contexts both differ from each other. That is, R is maximal if $\exists\, k, j \in 1, 2, \ldots, p$, with $k \neq j$ such that $S_\alpha[i_k - 1] \neq S_\alpha[i_j - 1]$ and $S_\alpha[i_k + \ell] \neq S_\alpha[i_j + \ell]$. A maximal repeat is referred to as a **maxmer**.

A repeat R is **super maximal** (a **super maxmer**) if the repeat contains no extendible occurrence. That is, R is maximal if $\forall\, k \neq j \in 1, 2, \ldots, p, S_\alpha[i_k - 1] \neq S_\alpha[i_j - 1]$ and $S_\alpha[i_k + \ell] \neq S_\alpha[i_j + \ell]$. A maximal repeat that is not super maximal is said to be **local maximal** (a **local maxmer**).

The definitions of super and local maxmers split the set of maxmers into two disjoint sets. Moreover, a repeat R has the following properties

- *R is super maximal* $\Rightarrow p \leq |\Sigma|$.

Proof. Because there are only $|\Sigma|$ symbols that can be assigned to either the left- or right-context of any C_i of the maxmer, it follows that if there are more than $|\Sigma|$ occurrences, at least two of them are extendible, proving the property. □

- $p = 2 \Rightarrow$ *R is super maximal.*

Proof. A set containing exactly 2 occurrences, both of which are extendible, is not a maxmer. □

- *R is local maximal* $\Rightarrow p \geq 3$.

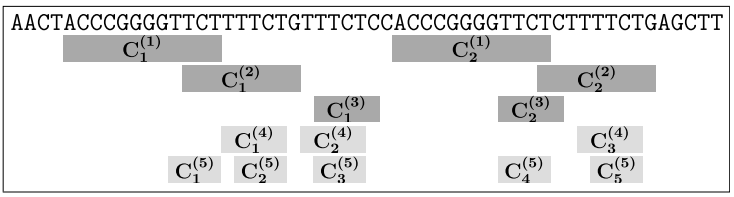
Proof. Corollary of previous property. □

- *R is local maximal* of length $\ell \Rightarrow \exists\, m$ proper subsets $R_i \subsetneq R$, containing q_i occurrences each of which is extendible respectively to elements of m super maxmers R'_i of length $\ell'_i > \ell, i = 1, 2, \ldots, m$, where $q = (\sum_{i=1}^{m} q_i) \leq p$ and $\forall\, i, q \geq q_i \geq 2$.

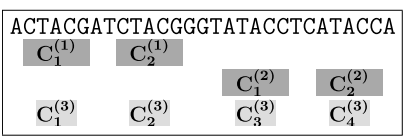
Proof. Let C_i be an occurrence of a given maxmer R of length ℓ, where $C_i \in \{C_1, C_2, \ldots, C_p\}$. The property is proved recursively. According to the definition of a local maxmer, q occurrences of the p occurrences of a local maxmer ($q \leq p$) can be split into m disjoint subsets sharing identical left- or right-context. Iterating on $k, k \geq 1, i = 1, 2, \ldots, m^{(k)}$ with $q_1^{(1)} = q$, we construct $q_i^{(k)}$ satisfying $\forall\, i, k, q_i^{(k)} \geq 2$ and

$$q_1^{(k)} = \sum_{i=1}^{m^{(k)}} q_i^{(k+1)}, \tag{A.1}$$

where within each subset i the $q_i^{(k+1)}$ occurrences are extended to occurrences of a longer maxmer. The index k indicates that the occurrences of an extendible maxmer



(a) Local and super maxmer; nested occurrences



(b) Local maxmers with all occurrences nested

Fig. A.1. Example of maxmers and nested occurrences within DNA nucleotide sequences; $\Sigma = \{\mathtt{A}, \mathtt{G}, \mathtt{C}, \mathtt{T}\}$. The occurrences of the local versus super maxmer are distinguished by light and dark gray backgrounds, respectively.

can be recursively divided into $m^{(k)}$ subsets of occurrences. Applying (A.1) successively to a given local maxmer R, the subsets of occurrences formed by successive division ultimately extend to super maxmers, according to the previous properties. □

From the last property, it follows that any local maxmer has a subsets of its occurrences that can be extended to element of a (longer) super maxmer. Therefore, if we defined a **nested** occurrence as an occurrence of a local maxmer that is extendible, we can define a **non-nested local maxmer** as a local maxmer having at least one occurrence that is not nested.

Noticing that because some local maxmers may have all of their occurrences nested, the set of maxmers of a given chromosome sequence can be compactly represented by the union of the set of non-nested local maxmers with the set of super maxmers.

Figure A.1 illustrates local and super maximal structure within a given DNA sequence. In Fig. A.1(a), the maxmer structure for lengths $\ell \geq 4$ consists of three super maxmers $\{C_i^{(1)}\}_{i=1,2}$, $\{C_i^{(2)}\}_{i=1,2}$, and $\{C_i^{(3)}\}_{i=1,2}$, and, two local maxmers $\{C_i^{(4)}\}_{i=1,2,3}$, and $\{C_i^{(5)}\}_{i=1,\ldots,5}$. The three pairs of occurrences $(C_1^{(2)}, C_2^{(2)})$, $(C_1^{(4)}, C_3^{(4)})$, and $(C_2^{(5)}, C_5^{(5)})$, exhibit a three-level structure of nesting. In a similar way, $C_3^{(5)}$ is nested into $C_1^{(3)}$ and $C_4^{(5)}$ is nested into $C_2^{(3)}$. Moreover, the maxmer $\{C_i^{(4)}\}_{i=1,2,3}$ is a non-nested local maxmer because its occurrence $C_2^{(4)}$ is not nested in any other maxmer occurence. Finally, Fig. A.1(b) gives an example of a local maxmer, where all the occurrences $\{C_i^{(3)}\}_{i=1,\ldots,4}$ are nested into the occurrences of two super maxmers $\{C_i^{(1)}\}_{i=1,2}$ and $\{C_i^{(2)}\}_{i=1,2}$.

Table A.1. Enhanced suffix array of ACTCTCTGCTCT.

i	SA[i]	LCP[i]	suffix
1	1	0	ACTCTCTGCTCT
2	2	0	CTCTCTGCTCT
3	4	4	CTCTGCTCT
4	9	4	CTCT
5	6	2	CTGCTCT
6	11	2	CT
7	8	0	GCTCT
8	3	0	TCTCTGCTCT
9	5	3	TCTGCTCT
10	10	3	TCT
11	7	1	TGCTCT
12	12	1	T

A.5. *Enhanced suffix array*

An *enhanced suffix array* data structure consists of array tables amongst which the suffix array table SA is computed first.

The suffix array table of an n-length string S_α is an array of integers ranging from 1 to n that specify the lexicographic order of the n suffixes of the string. This suffix array provides a lexicographical index of all possible sub-strings within S_α as a table of size identical to string length. Fast practical suffix index sort methods can perform the index sort in linear time and linear space.[48–52]

The second table LCP contains the longest common prefixes shared by the (SA[$i-1$])th and (SA[i])th suffixes of the string, where $i = 2, 3, \ldots, n$ and LCP[1] = 0. Given the suffix array SA, the LCP table can be efficiently computed in linear time and space,[44] or simultaneously with the computation of the suffix array from the terminated string.[32] Table A.1 shows an example of the enhanced suffix array of the DNA sub-sequence ACTCTCTGCTCT.

The ESA allows the retrievial and classification of all maxmers, by using the **ℓ-intervals** embedded into ESA structure. An interval of indices $[i, i+1, \ldots, j], i < j \leq n$ is an **ℓ-interval[i, j]** if

(1) LCP[i] $< \ell$,
(2) LCP[k] $\geq \ell$ for $i+1 \leq k \leq j$,
(3) LCP[k] $= \ell$ for at least one k with $i+1 \leq k \leq j$,
(4) LCP[$j+1$] $< \ell$.

Thus a ℓ-interval $[i, j]$ corresponds to a set of suffixes containing $j - i + 1$ instances of a given sub-sequence with length ℓ. Alternatively, a ℓ-interval can be thought of as a repeat set with occurrences of length ℓ. Given a suffix array SA and the LCP table, all ℓ-intervals can be obtained in linear time.[44] Maxmers can be identified and classified using the ℓ-interval structure and table.[33] Table A.2 gives an example of the ℓ-intervals, $\ell \geq 2$ for the enhanced suffix array in Table A.1.

Table A.2. ℓ-intervals ($\ell \geq 2$) for `ACTCTCTGCTCT`.

ℓ	i, j	$\mathrm{SA}[i], \mathrm{SA}[i+1], \ldots, \mathrm{SA}[j]$
2	$2, 6$	$2, 4, 9, 6, 11$
3	$8, 10$	$3, 5, 10$
4	$2, 4$	$2, 4, 9$

Require: SA, LCP
  procedure PROCESS_INTERVAL(i, j)
    $\ell \leftarrow \mathrm{LCP}[j]$
    if $\ell \geq \ell_{\min}$ **then**
      initialize tables LCTT and RCTT with zeros
      $t_{\mathrm{mm}} \leftarrow$ **false**
      $\sigma \leftarrow \mathrm{S}_\alpha\big[\mathrm{SA}[i] - 1\big]$
      for $k \leftarrow i, j$ **do**
        $\sigma_l \leftarrow \mathrm{S}_\alpha\big[\mathrm{SA}[k] - 1\big]$
        $\sigma_r \leftarrow \mathrm{S}_\alpha\big[\mathrm{SA}[k] + \ell\big]$
        $\mathrm{LCTT}[\sigma_l] \leftarrow \mathrm{LCTT}[\sigma_l] + 1$
        $\mathrm{RCTT}[\sigma_r] \leftarrow \mathrm{RCTT}[\sigma_r] + 1$
        if $\sigma \neq \sigma_l$ **then**
          $t_{\mathrm{mm}} \leftarrow$ **true**
        end if
      end for
      if $\begin{cases} j - i + 1 \leq |\Sigma| \\ \text{and} \\ \forall \sigma \in \Sigma \begin{cases} \mathrm{LCTT}[\sigma] \leq 1 \\ \mathrm{RCTT}[\sigma] \leq 1 \end{cases} \end{cases}$ **then**
        PROCESS_SUPER(ℓ, i, j)
      else if $t_{\mathrm{mm}} =$ **true then**
        PROCESS_LOCAL_START(ℓ, i, j)
        for $k \leftarrow i, j$ **do**
          $\sigma_l \leftarrow \mathrm{S}_\alpha\big[\mathrm{SA}[k] - 1\big]$
          $\sigma_r \leftarrow \mathrm{S}_\alpha\big[\mathrm{SA}[k] + \ell\big]$
          if $\mathrm{LCTT}[\sigma_l] > 1$ or $\mathrm{RCTT}[\sigma_r] > 1$ **then**
            PROCESS_NON_NESTED_OCC(k)
          else
            PROCESS_NESTED_OCC(k)
          end if
        end for
        PROCESS_LOCAL_STOP
      end if
    end if
  end procedure

Fig. A.2. Algorithm for maxmer identification and classification.

A.6. *Maxmer identification and classification algorithm*

Given the enhanced suffix array tables SA, LCP, and given the ℓ-interval $[i, j]$, the left and right context of the length-ℓ repeat occurrence $C_k, k = 1, 2, \ldots, j - i + 1$, are obtained by $S_\alpha[SA[i] - 1]$ (or 1 if $SA[i] = 1$) and $S_\alpha[SA[i] + \ell]$ respectively, where, $\ell = LCP[j]$.

In Fig. A.2, we give an algorithm, PROCESS_INTERVAL, for assessing whether a given ℓ-interval is a local maxmer, a super maxmer, or not a maxmer. Then, in the case of a local maxmer, each occurrence is processed to determine if it is nested or not. The b-size ($b = |\Sigma|$) tables LCTT and RCTT, take their indices σ from the alphabet $\Sigma(\sigma \in \Sigma)$, and count the number of each symbol of the alphabet appearing in the left and right contexts, respectively, of the maxmer occurrences. In this algorithm, the variable t_{mm} is used to assess whether a ℓ-interval is a maxmer or not. It takes the value **true** if there exist at least two occurrences having different left context. The maxmer condition on the right context is always verified because the ℓ-interval contains at least one elements that is not right extensible (see definition of ℓ-interval in A.5).

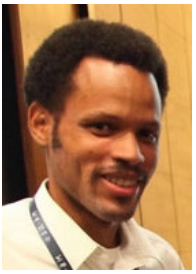
The functions (which has to be defined according to the application) PROCESS_SUPER, PROCESS_LOCAL_START, PROCESS_NON_NESTED_OCC, PROCESS_NESTED_OCC, PROCESS_LOCAL_STOP, allow to post-process the identified and labeled maxmer.

References

1. Ohno S, *Evolution by Gene Duplication*, Springer-Verlag, Berlin, 1970, ISBN 0-04-575015-7.
2. Ohta T, Further simulation studies on evolution by gene duplication, *Evolution* **42**(2): 375–386, 1988.
3. Taylor JS, Raes J, Duplication and divergence: The evolution of new genes and old ideas, *Ann Rev Genet* **38**:615–643, 2004.
4. Kazazian HH Jr, Mobile elements: Drivers of genome evolution, *Science* **303**(5664): 1626–1632, 2004.
5. Bailey JA, Eichler EE, Primate segmental duplications: Crucibles of evolution, diversity and disease, *Nat Rev Genet* **7**:552–564, 2006.
6. Cheung J, Estivill X, Khaja R, MacDonald JR, Lau K, Tsui LC, Scherer SW, Genome-wide detection of segmental duplications and potential assembly errors in the human genome sequence, *Genome Biol* **4**(4):R25, 2003.
7. Zhang L, Lu HHS, Chung WY, Yang J, Li WH, Patterns of segmental duplication in the human genome, *Mole Biol Evol* **22**(1):135–141, 2004.
8. Mantegna RN, Buldyrev SV, Goldberger AL, Havlin S, Peng CK, Simons M, Stanley HE, Linguistic features of noncoding DNA sequences, *The Amer Phys Soc* **73**(23):3194–3171, 1994.
9. Csűrös M, Noé L, Kucherov G, Reconsidering the significance of genomic word frequencies, *TRENDS Genet* **23**(11):543–546, 2007.
10. Chor B, Horn D, Goldman N, Levy Y, Massingham T, Genomic DNA k-mer spectra: Models and modalities, *Genome Biol* **10**(10):R108.1–10, 2009.

11. Sindi SS, Hunt BR, Yorke JA, Duplication count distributions in DNA sequences, *Phys Rev E* **78**(6):061912(1–11), 2008.
12. Zipf GK, *Human Behaviour and the Principle of Least Effort*, Addison-Wesley, Cambridge, MA, 1949.
13. Salerno W, Havlak P, Miller J, Scale-invariant structure of strongly conserved sequence in genomic intersections and alignments, *Proc Nat Acad Sci* **103**(35):13121–13125, 2006.
14. Miller J, Colossal ultraconservation and super-colossal ultraconservation, IPSJ SIG Technical Report **2009-BIO-17**(7):1–8, 2009.
15. Tran T, Havlak P, Miller J, MicroRNA enrichment among short 'ultraconserved' sequences in insects, *Nucleic Acids Res* **34**(9):e65, 2006.
16. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ, Basic local alignment search tool, *J Molec Biol* **215**(3):403–410, 1990.
17. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ, Gapped BLAST and PSI-BLAST: A new generation of protein database search programs, *Nucleic Acids Res* **25**(17):3389–3402, 1997.
18. Schwartz S, Kent WJ, Smit A, Zhang Z, Baertsch R, Hardison RC, Miller W, Human-mouse alignments with BLASTZ, *Genome Res Meth* **13**:103–107, 2003.
19. Harris RS, Improved pairwise alignment of genomic DNA. Ph.D. Thesis, The Pennsylvania State University. 2007. Available at http://www.bx.psu.edu/∼rsharris/rsharris_phd_thesis_2007.pdf. Software available at http://www.bx.psu.edu/∼rsharris/lastz/. Accessed October 2013.
20. Gao K, Miller J, Algebraic distribution of segmental duplication lengths in whole-genome sequence self-alignments, *PLoS One* **7**(6):e18464, 2011.
21. Castelo AT, Martins W, Gao GR, TROLL — Tandem repeat occurrence locator, *Bioinformatics* **18**(4):634–636, 1999.
22. Benson G, Tandem repeats finder: A program to analyze DNA sequences, *Nucleic Acids Res* **27**(2):573–580, 1999.
23. Lefebvre A, Lecroq T, Dauchel H, Alexandre J, FORRepeats: Detects repeats on entire chromosomes and between genomes, *Bioinformatics* **19**(3):319–326, 2003.
24. Becher V, Deymonnaz A, Heiber P, Efficient computation of all perfect repeats in genomic sequences of up to half a gigabyte, with a case study on the human genome, *Bioinformitics* **25**(14):1746–1753, 2009.
25. Saha S, Bridges S, Magbanua ZV, Peterson DG, Computational approaches and tools used in identification of dispersed repetitive DNA sequences, *Tropical Plant Biol* **1**(1):85–96, 2008.
26. Saha S, Bridges S, Magbanua ZV, Peterson DG, Empirical comparison of *ab initio* repeat finding programs, *Nucleic Acids Res* **36**(7):2284–2294, 2008.
27. Delcher AL, Phillippy A, Carlton J, Salzberg SL, Fast algorithms for large-scale genome alignment and comparision, *Nucleic Acids Res* **30**(11):2478–2483, 2002.
28. Kurtz S, Choudhuri JV, 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.
29. Kurtz S, The Vmatch large scale sequence analysis software. Available at http://www.vmatch.de/. Accessed October 2013.
30. Ukkonen E, On-line construction of suffix trees, *Algorithmica* **14**(3):249–260, 1995.
31. Gonnet GH, Baeza-Yates RA, New indices for text: PAT trees and PAT arrays, in Frakes WB, Baeza-Yates RA (eds.), *Information Retrieval: Data Structures and Algorithms*, Prentice-Hall, Upper Saddle River, NJ, USA, pp. 66–82, 1992.
32. Manber U, Myers G, Suffix arrays: A new method for on-line string searches, *SIAM J Comput* **22**(5):935–948, 1993.

33. Abouelhoda MI, Kurtz S, Ohlebusch E, Replacing suffix trees with enhanced suffix arrays, *J Discrete Algorithms* **2**(1):53–84, 2004.
34. Gusfield D, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*, Cambridge University Press, US, 1997, ISBN 0-521-58519-8.
35. Nicolas J, Rousseau C, Siegel A, Peterlongo P, Coste c Fran Durand P, Tempel S, Valin AS, Mahé F, Modeling local repeats on genomic sequences, Technical Report RR-6802, INRIA, 2008.
36. Luscombe NM, Qian J, Zhang Z, Johnson T, Gerstein M, The dominance of the population by a selected few: Power-law behaviour applies to a wide variety of genomic properties, *Genome Biol* **3**(8):0040.1–0040.7, 2002.
37. Martindale C, Konopka AK, Oligonucleotide frequencies in DNA follow a Yule distribution, *Comput Chem* **20**(1):35–38, 1996.
38. Gojobori T, Li WH, Graur D, Patterns of nucleotide substitution in pseudogenes and functional genes, *J Mole Evol* **18**(5):360–369, 1982.
39. Koroteev MV, Miller J, Scale-free duplication dynamics: A model for ultraduplication, *Phys Rev E* **84**(6):061919(1–10), 2011.
40. Taillefer E, Miller J, Algebraic length distribution of sequence duplications in whole genomes, *Int Conf Natural Computation*, Shanghai, China, pp. 1454–1460, 2011.
41. Pop M, Salzberg SL, Shumway M, Genome sequence assembly: Algorithms and issues, *Computer* **35**(7):47–54, 2002.
42. Pop M, Genome assembly reborn: Recent computational challenges, *Computer* **10** (4):354–366, 2009.
43. Narzisi G, Mishra B, Comparing de novo genome assembly: The long and short of it, *PloS One* **6**(4):e19175, 2011.
44. Kasai T, Lee G, Arimura H, Arikawa S, Park K, Linear-time longest-common-prefix computation in suffix arrays and its applications, in Amir A, Landau G (eds.), *Combinat Patt Match*, Springer-Verlag, Berlin, Heidelberg, pp. 181–192, 2001.
45. Barbour AD, Chryssaphinou O, Compound poisson approximation: A userś guide, *Ann Appl Prob* **11**(3):964–1002, 2001.
46. Lippert RA, Huang H, Waterman MS, Distributional regimes for the number of k-word matches between two random sequences, *Proc Nat Acad Sci USA* **99**(22):13980–13989, 2002.
47. Taillefer E, Miller J, Exhaustive computation of exact sequence duplications in whole genomes *via* super and local maximal repeats, *Int Conf Computer Engineering and Bioinformatics*, Cairo, Egypt, pp. 22–29, 2011.
48. Kärkkäinen J, Sanders P, Burkhardt S, Simple linear work suffix array construction, *J ACM* **53**(6):918–936, 2006.
49. Nong G, Zhang S, Chan WH, Two efficient algorithms for linear time suffix array construction, *IEEE Trans Comput*, IEEE Computer Society Digital Library. IEEE Computer Society, 2010. Available at http://doi.ieeecomputersociety.org/10.1109/TC.2010.188.
50. Mori Y, SAIS: An implementation of the induced sorting algorithm. Available at http://yuta.256.googlepages.com/sais.
51. Mori Y, Libdivsufsort: A lightweight suffix-sorting library. Available at http://code.google.com/p/libdivsufsort/.
52. Puglisi SJ, Smyth WF, Turpin AH, A taxonomy of suffix array construction algorithms, *ACM Comput Surveys* **39**(2), 2007, article 4.



Eddy Taillefer joined the Okinawa Institute of Science and Technology (OIST) Graduate University in 2009 as part of the Physics and Biology unit to work toward computational analysis of genome sequence data and to develop new bioinformatics algorithms. He obtained his Ph.D. in Computer Science from the Department of Knowledge Engineering and Computer Science, Doshisha University in 2008. His research interests include computational science, applied computer science, and bioinformatics.

Jonathan Miller discovered the first zinc finger in his Biology Ph.D. thesis, as an experimentalist in Aaron Klugs lab at MRC LMB Cambridge. Subsequently, he obtained Ph.D. in Physics at Caltech, resolving a longstanding oversight in Onsager's theory of statistical mechanics of the two-dimensional ideal fluid. After postdoctoral work in condensed matter theory at AT&T Bell Labs, the University of Chicago, NEC Research Institute, and Princeton University, he joined the faculty of Baylor College of Medicine, where he began the first investigation of conserved sequence length distributions in natural genomes. Those eventually led OIST to the duplication length studies, one aspect of which is discussed here.