BIOINFORMATICS ORIGINAL PAPER
Vol. 27 no. 11 2011, pages 1466–1472
doi:10.1093/bioinformatics/btr176

Genome analysis

Advance Access publication April 6, 2011

Alignment-free detection of local similarity among viral and bacterial genomes

Mirjana Domazet-Lošo[1,2] and Bernhard Haubold[1,*]

[1]Department of Evolutionary Genetics, Max-Planck-Institute for Evolutionary Biology, 24306 Plön, Germany and [2]Department of Applied Computing, Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia

Associate Editor: Alex Bateman

ABSTRACT

Motivation: Bacterial and viral genomes are often affected by horizontal gene transfer observable as abrupt switching in local homology. In addition to the resulting mosaic genome structure, they frequently contain regions not found in close relatives, which may play a role in virulence mechanisms. Due to this connection to medical microbiology, there are numerous methods available to detect horizontal gene transfer. However, these are usually aimed at individual genes and viral genomes rather than the much larger bacterial genomes. Here, we propose an efficient alignment-free approach to describe the mosaic structure of viral and bacterial genomes, including their unique regions.

Results: Our method is based on the lengths of exact matches between pairs of sequences. Long matches indicate close homology, short matches more distant homology or none at all. These exact match lengths can be looked up efficiently using an enhanced suffix array. Our program implementing this approach, alfy (ALignment-Free local homologY), efficiently and accurately detects the recombination break points in simulated DNA sequences and among recombinant HIV-1 strains. We also apply alfy to *Escherichia coli* genomes where we detect new evidence for the hypothesis that strains pathogenic in poultry can infect humans.

Availability: alfy is written in standard C and its source code is available under the GNU General Public License from http://guanine.evolbio.mpg.de/alfy/. The software package also includes documentation and example data.

Contact: haubold@evolbio.mpg.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on February 16, 2011; revised on March 11, 2011; accepted on March 30, 2011

1 INTRODUCTION

In contrast to most eukaryotes, viruses and bacteria reproduce asexually. Nevertheless, their genomes are littered with evidence of occasional and localized recombination, usually referred to as ‘horizontal gene transfer’. Horizontal gene transfer is observable as an abrupt change in homology along an alignment of two or more sequences. The importance of horizontal gene transfer for the evolution of immune evasion and antibiotic resistance of pathogenic microbes has been known for decades (Maynard Smith *et al.*, 1991). Correspondingly, there has been and still is a lot of interest in devising computer programs to automate the detection of mosaic gene or genome structure (Langille *et al.*, 2010).

Most of the current methods for annotating mosaic genome structure are based on alignments and target viral genomes or individual bacterial genes (Kosakovsky Pond *et al.*, 2009; Westesson and Holmes, 2009). These methods are highly accurate but tend to be too slow for the analysis of whole bacterial genomes. Recently, Didelot *et al.* (2010) expanded their original Monte Carlo Markov Chain method for inferring recombination rates (Didelot and Falush, 2007) by also localizing the recombination events in alignments of whole bacterial genomes. However, even given such an alignment, the computational burden of their, again, highly accurate coalescent-based method is considerable.

If speed is of concern, dramatic improvements are possible by simplifying or eliminating altogether any explicit model under which inference is made, and in addition by not basing the analysis on a full multiple sequence alignment. Rozanov *et al.* (2004) provide a web-based genotyping tool for viral genomes hosted by the NCBI. Their algorithm essentially consists of sliding a window across the query sequence and blasting the windows against the set of subject sequences to determine the closest homologs across the query.

An even more radical departure from model- and alignment-based treatment of mosaic genome structure is the alignment-free method for HIV typing proposed by Wu *et al.* (2007). Alignment-free methods of sequence comparison tend to be very fast, albeit less accurate than alignment-based sequence comparison (Reinert *et al.*, 2009; Vinga and Almeida, 2003). Best known among the alignment-free approaches are those based on k-word composition, which is also what Wu *et al.* (2007) have used. Here all substrings of length k are counted in the two sequences compared and a distance between the vectors of word counts is defined. Unfortunately, it is difficult to convert distances between word count vectors to true evolutionary distances.

We have recently developed two methods that start from the lengths of exact matches between pairs of sequences and convert these to substitutions per site (Haubold *et al.*, 2011, 2009). The advantage of this approach is that exact match lengths can be looked up efficiently using suffix trees implemented as their more recent abstractions known as enhanced suffix arrays (Abouelhoda *et al.*, 2002) and compressed suffix arrays (Ferragina *et al.*, 2008).

In this article, we use exact match lengths between a query and a potentially large set of subject sequences to solve a specialized,

[*]To whom correspondence should be addressed.

but frequently occurring problem in sequence analysis: at each position in the query sequence, we wish to determine which subject sequence—if any—is most closely related to the query. This closest neighbors problem is common in microbiology and correspondingly we apply our method to the genomes of HIV and *Escherichia coli*.

The most prevalent form of HIV is the M group of HIV-1, which is divided into eight subtypes (A, B, C, D, F, G, H, J) based on the sequence of their 9 and 10 kb genomes. However, recombination between these subtypes in multiply infected individuals leads to the emergence of new forms of the virus. Accurate detection of the origin of individual HIV-1 genome segments has clinical implications and the development of software to automate this procedure has received corresponding attention (Kosakovsky Pond *et al.*, 2009).

Closely related strains of bacteria form 'pan-genomes' consisting of a common core and a variable part that is horizontally mobile (Tettelin *et al.*, 2005). One example for such a group is *E.coli*, which has a genome size ranging from 4.6 to 5.7 MB. This taxon comprises commensal strains as well as strains that are pathogenic in humans and homeothermic animals. An important question concerning *E.coli* is to what extend strains that are pathogens of live stock might cause infections in humans (Johnson *et al.*, 2007).

In the following sections, we first describe our alignment-free method for determining local homology. We test this method through simulations and then use it to classify HIV-1 recombinant strains and to compare *E.coli* genomes. When comparing the *E.coli* genomes, we not only look for switches in local homology indicative of horizontal gene transfer, but also for regions with no close homolog among the subject sequences. Investigation of these 'private' genomic regions leads us to the detection of new evidence for the long-held suspicion that animal *E.coli* pathogens can also infect humans (Johnson *et al.*, 2007).

2 METHODS

Given a query DNA sequence, Q, and a set of subject DNA sequences, $\mathbf{S}=\{S_1,...,S_n\}$, we wish to determine the subject sequences that Q is most closely related to at every position p in Q. For instance, in Figure 1A $Q[1,2]=\mathtt{TA}$ matches $S_3[1,2]$ and $Q[3,4]$ matches $S_2[1,2]$. We therefore say that $Q[1,2]$ is most closely related to S_3 and $Q[3,4]$ is most closely related to S_2. In this section, we describe a simplified version of our algorithm to locally annotate Q. The algorithm we actually implemented is detailed in Algorithm S1 of the Supplementary Material.

When comparing Q to a specific subject, S_i, we denote by $h_{i,p}$ the length of the shortest prefix of the query suffix $Q[p,|Q|]$ that starts at position p and is absent from S_i. We follow our previous convention of calling these shortest absent prefixes *shustrings* for SHortest Unique subSTRINGS (Haubold and Wiehe, 2006; Haubold *et al.*, 2005). The length of the longest shustring starting at $Q[p]$ when compared to all subject sequences is $H_p=\max_{1\leq i\leq n}h_{i,p}$. Notice that H_p is bounded by the query length: $H_p\leq|Q|-p+1$. For our example sequences in Figure 1A $h_{1,1}=|\mathtt{T}|=1$; $h_{2,1}=|\mathtt{T}|=1$; $h_{3,1}=|\mathtt{TAG}|=3$; and $H_1=\max\{1,1,3\}=3$.

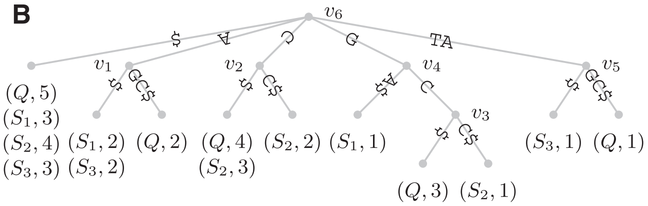
The set of subject sequences that induce the longest shustring at $Q[p]$ is $\mathbf{S}_p=\{S_i\in\mathbf{S}|h_{i,p}=H_p\}$. For our example, sequences in Figure 1A $\mathbf{S}_1=\{S_3\}$.

To determine which segment of Q is most closely related to which subject sequence, we compute all values of H_p and $\mathbf{S}_p$. This is done by constructing a generalized suffix tree consisting of the query and all subject sequences (Fig.1B). We assume that at every branch node v_i of this tree we can look up the length of the concatenated labels along the path from the root to v_i, that is, the *string depth* of v_i. For example, the string depth of v_3 in Figure 1B is 2.

This generalized suffix tree is traversed once, and whenever a leaf node $w=(Q,p)$ is encountered, H_p is computed as one plus the string depth of the lowest common ancestor node of w and any leaf node referring to a subject sequence (Fig. 1C). Moreover, $\mathbf{S}_p$ is the set of subject sequences in the subtree rooted on that lowest common ancestor.

A

	1	2	3	4	5
Q	T	A	G	C	$
S_1	G	A	$		
S_2	G	C	C	$	
S_3	T	A	$		



C

Query position, p	1	2	3	4	5
H_p	3	2	3	2	1
$\mathbf{S}_p$	$\{S_3\}$	$\{S_1,S_3\}$	$\{S_2\}$	$\{S_2\}$	$\{S_1,S_2,S_3\}$

D

$$([1,2],\{S_3\},3),([3,5],\{S_2\},3)$$

Fig. 1. Finding closest neighbors intervals. (**A**) Unaligned input sequences. (**B**) Generalized suffix tree of input sequences. (**C**) Table of longest shustring lengths, H_p, and the sets of subject sequences that induce the longest shustring length, $\mathbf{S}_p$. (**D**) The two closest neighbors intervals along Q. See text for further details.

Given the table of H_p and $\mathbf{S}_p$ values in Figure 1C, we next cover Q with contiguous intervals of closest neighbors. Such closest neighbors intervals have the form $([lb,rb],\mathbf{S}',h)$, where lb and rb are the left and right borders, $\mathbf{S}'=\mathbf{S}_{lb}$, and $h=H_{lb}$. The intervals are constructed by traversing the array $H_1,H_2,...,H_{|Q|}$ and whenever $H_p>H_{p-1}$, we do two things:

(1) close the current closest neighbors interval by setting $rb\leftarrow p-1$ and $\mathbf{S}'\leftarrow\mathbf{S}_{lb}$;

(2) open a new closest neighbors interval with $lb\leftarrow p$ and $h\leftarrow H_p$.

For our example in Figure 1C, this procedure results in the detection of the two closest neighbors intervals $([1,2],\{S_3\},3)$ and $([3,5],\{S_2\},3)$.

Finally, the closest neighbors intervals are summarized into larger regions of similarity. This is done using a sliding window approach: for each subject sequence referred to by the closest neighbors intervals within a window, the shustring lengths are summed and the one or more subject sequences with the greatest sum are designated the closest neighbors of Q at the given position.

Haubold *et al.* (2009) derived the null distribution of shustring lengths as a function of the number of mismatches between pairs of sequences. In the limit of 0.75 mismatches per site this corresponds to the distribution of shustring lengths expected by chance alone. We can therefore also identify query regions without close homologs among the subject set. Such regions are characterized as having an average shustring length that is less than the maximal shustring length expected by chance alone.

2.1 Implementation

The procedure for finding closest neighbors intervals can easily be generalized to more than one query sequence and this is the version that we have implemented. However, when dealing with large datasets the auxiliary table of H_p and $\mathbf{S}_p$ values would impose a substantial memory overhead. So instead of explicitly constructing this table, we directly build the set of closest neighbors intervals during suffix tree traversal. This yields a much more memory efficient implementation, and by keeping the closest neighbors as an ordered binary tree the concomitant cost in runtime is minimal (Algorithm S1).

The final program is called alfy for ALignment-Free local homologY and is written in standard C. The suffix tree was implemented as an enhanced suffix array (Abouelhoda *et al.*, 2002) using the suffix array library distributed by Manzini and Ferragina (2002).

2.2 Runtime analysis

alfy determines closest homologs in three phases: it begins by constructing an enhanced suffix array of all m query and n subject sequences. This array is then traversed to compute the list of closest neighbors intervals. Finally, the list of closest neighbors intervals is subjected to sliding window analysis to produce the desired annotation of the query. Suffix array construction for m query and n subject sequences of length L ideally takes time $O((m+n)L)$. Insertion of a single interval node in an interval tree takes $O(\log L)$; the construction of one tree per query sequence thus takes a total of $O(mL\log L)$ time. Traversal of m interval trees takes time $O(mL)$. Therefore, the whole procedure takes time $O((m+n)L+mL\log L+mL)$. Notice that if a single query is compared to a large number of subjects, that is, $m=1$ and $n \gg \log L$, the run time approaches $O(nL)$.

2.3 Phylogeny reconstruction

Pairwise substitution rates were estimated from whole *E.coli* genomes using the program kr (Domazet-Lošo and Haubold, 2009). The substitution rates were clustered using the neighbor joining algorithm implemented in the program neighbor, which is part of the software package PHYLIP (Felsenstein, 1989). Trees were midpoint-rooted using retree and drawn using drawgram, which are also part of PHYLIP.

3 RESULTS

In the following sections, we first apply alfy to simulated and then to empirical data to explore the efficiency and accuracy of the program.

3.1 Runtime and memory consumption

We simulated sequence samples consisting of one query and 10 or 100 subject sequences with 0.13 or 0.006 segregating sites/nucleotide, respectively, between the query and its closest neighbor using the program Dawg (Cartwright, 2005). Figure 2A shows the runtime on a single AMD Opteron 2.3 GHz processor as a function of sequence length. The runtime grows approximately linearly in the sample size and the sequence length, as expected from the analysis in Section 2.2. Similarly, the memory requirement is linear in the amount of sequence data, as expected for a suffix array-based procedure (Fig. 2B).

3.2 Accuracy of homology detection

To test the accuracy of local homology detection by alfy, we simulated samples of three pure subject sequences and one recombinant query again using Dawg (Cartwright, 2005). As shown in Figure 3, this was done by generating sequences with distance $2s$ between Q and S_1 or between Q and S_2. Five such segments were generated, each segment evolving either along genealogy G_1 or along genealogy G_2.

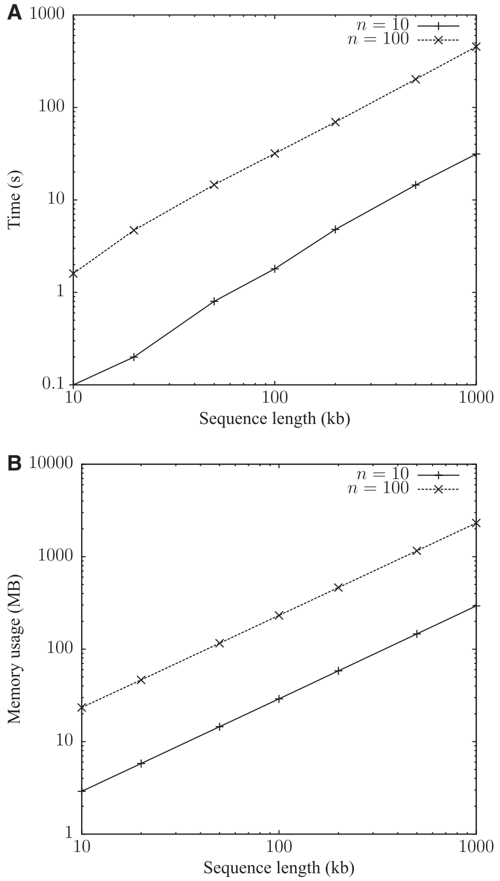


Fig. 2. Runtime (**A**) and memory consumption (**B**) of alfy as a function of sequence length and sample size, *n*.

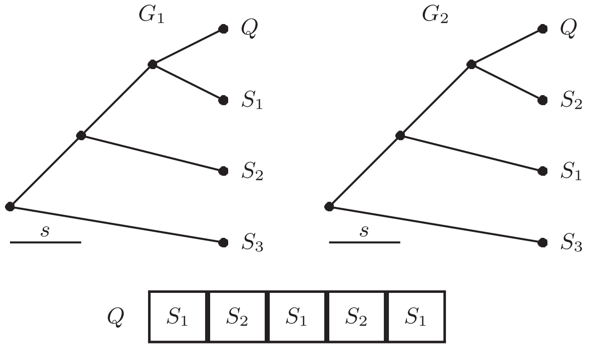


Fig. 3. Samples of four sequences were simulated in five segments alternately along genealogies G_1 and G_2. The closest homolog of query Q therefore switched between S_1 and S_2 along its length as illustrated at the *bottom*.

Detection accuracy was scored as the fraction of nucleotides correctly assigned to S_1 or S_2. For segments of length 2 kb, the detection accuracy increased with distance up to $s \approx 0.05$, while it decreased significantly for distances greater than $s \approx 0.1$ (Fig. 4A). Short segments were generally more difficult to locate correctly than long segments and as a result Figure 4A with fragment length 2 kb displays a lower overall accuracy than Figure 4B with fragment length 20 kb. Moreover, in Figure 4B longer windows (2 and 20 kb) give more accurate results for small and large ($s > 0.1$) distances,

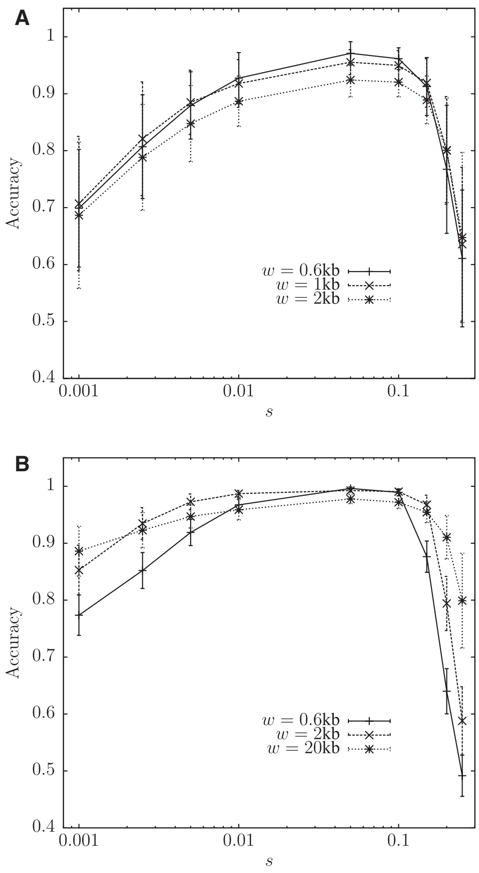


Fig. 4. Accuracy of local homology detection as a function of evolutionary distance, *s*, using sliding windows of length *w*. (**A**) Query consists of 2 kb fragments; (**B**) query consists of 20 kb fragments. The query structure and the meaning of *s* are illustrated in Figure 3.

while for intermediate *s* the detection accuracy is less sensitive to window length.

3.3 Application to genome data

We applied our homology detection method to viral and bacterial genomes in order to solve two common tasks: genotyping of circulating recombinant forms of HIV-1, and detection of horizontal gene transfer and unique segments among *E.coli* genomes.

3.3.1 Detection of circulating recombinant forms of HIV-1 We started the analysis by comparing alfy to two fast HIV-1-subtyping tools: the NCBI genotyping tool (Rozanov *et al.*, 2004), and the *k*-word-based clustering method by Wu *et al.* (2007). These programs and alfy were applied to two published HIV-1 datasets using the approach outlined by Wu *et al.* (2007), who considered the top two closest sequences as valid annotations. The first dataset consisted of 91 circulating recombinant forms (CRFs, query) compared to 42 pure subtype strains (subject) (Leitner *et al.*, 2005; Wu *et al.*, 2007). The accuracy of alfy was 93.4%, while Wu *et al.* (2007) reported the accuracy of the NCBI tool as 73.4%, and that of their own tool as 87.3%. The window size for alfy was 300, the same as that used in the NCBI tool.

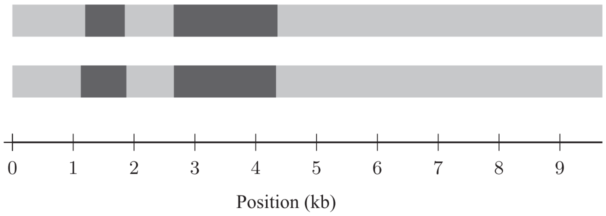


Fig. 5. HIV-1 strain A_DQ083238 is an A/C recombinant. The position of the material transferred from a C strain is shown in dark gray as diagnosed by SCUEAL (*top*) and alfy (*bottom*). alfy was run with window size 300 bp.

The second dataset consisted of a query set of 266 HIV-1 CRFs compared to a subject set consisting of 42 pure subtype strains and 65 recombinant strains of known subtype (Leitner *et al.*, 2005; Wu *et al.*, 2007). Here, alfy subtyped 263 strains correctly, the NCBI tool 264 and Wu *et al.* (2007) annotated 242 strains correctly.

In addition to these two fast typing methods, we compared alfy to SCUEAL. This is a phylogeny-based HIV-1 subtyping software that relies on a genetic algorithm to pick the most likely recombinant pattern given a query sequence and an alignment of subject sequences (Kosakovsky Pond *et al.*, 2009). Its authors report that it detects < 1% false positives on simulated data and we used it to type a single HIV-1 strain (A_DQ083238) against our standard 42 pure strains. This query was originally annotated as a pure A strain, but we have previously noted that it is an A/C recombinant (Domazet-Lošo and Haubold, 2009). Correspondingly, SCUEAL and alfy report the transfer of two large C fragments amounting to 24.3% (SCUEAL) and 25.1% (alfy) of the genome (Fig. 5). The location of the two C-segments determined by SCUEAL (Fig. 5, *top*) is similar to the annotation by alfy (Fig. 5, *bottom*).

We also used the comparison between strain A_QD083238 and the 42 reference strains to benchmark the resource requirements of alfy, the NCBI tool, the Wu *et al.* (2007) tool and SCUEAL. Table 1 shows that alfy took 0.4 s for this analysis. Unfortunately, the timing of the NCBI tool is not straight forward since this is only available as a web service. To get a rough estimate, we repeatedly ran BLAST with a 300 bp query taken from the query strain against the 42 reference strains. 97 such calls would cover the query in 100 bp steps and this took 1.4 s, which is presumably an upper bound on the true runtime. The Java program by Wu *et al.* (2007) took 3.5 min, while SCUEAL took 6.9 h.

As to memory requirements, BLAST used much less memory than alfy (2.2 KB versus 9 MB; Table 1). This is expected for a method that indexes the short query (BLAST) as opposed to indexing the long subject (alfy). Wu *et al.* (2007) required the most memory (2.2 GB), while SCUEAL still used over 17 times more memory than alfy (160 MB).

3.3.2 Investigating genomes of pathogenic E.coli To test the hypothesis that *E.coli* strains pathogenic in birds might also infect humans, Johnson *et al.* (2007) sequenced a strain of avian pathogenic *E.coli* (APEC_O1) and compared it to the genomes of four fully sequenced *E.coli* strains, three of which cause urinary tract infection in humans (Fig. 6). Consistent with their hypothesis, they found that APEC_O1 is very similar to the human uropathogenic strain UTI89.

Figure 7 shows the comparison between APEC_O1 as query and the remaining four *E.coli* strains as subject using alfy with a window

Table 1. Time and memory requirements of four HIV-1 typing tools when applied to one query and 42 subject genomes

Tool	Time	Memory
alfy	0.4 s	9 MB
BLAST	1.4 s	2.3 KB
Wu *et al.* (2007)	3.5 m	2.2 GB
SCUEAL	6.9 h	160 MB

size of 1 kb. This analysis of the 25.0 MB of sequence data required 85 s runtime and 961 MB of memory on our reference machine. As expected, a high proportion (90.5%) of the APEC_01 genome is most similar to its closest relative, UTI89. However, like Johnson *et al.* (2007) we found that between positions 4.71 and 4.76 MB, there is a 48 kb region that has been horizontally transferred from a strain like CFT073 (Fig. 7, *arrow*). Johnson *et al.* (2007) also observed that 4.5% of APEC_O1 open reading frames (ORFs) had no homolog among the other *E.coli* strains investigated, that is, these ORFs were private to APEC_01. This is similar to our estimate of 3.0% private nucleotides in that strain.

The number of fully sequenced *E.coli* strains contained in RefSeq has grown to 29 since the study by Johnson *et al.* (2007). Figure 8 shows the phylogeny of these strains with the five taxa investigated by Johnson *et al.* (2007) (Fig. 6) marked in bold. The analysis of the 151.8 MB that make up these genomes plus APEC_01 took alfy 744 s and occupied 5.8 GB of memory.

The private fraction of the genome shrank from 3.0% to 0.7% and Table 2 shows the 10 unique regions of length 1000 bp or greater that we identified. The longest private region was located between 2 190 501 and 2 196 650. We used megablast to compare these 6150 bp against the full nucleotide database (nr) and found a single hit that spanned the entire query. This hit comprised part of the gene cluster that specifies the O1-antigen of *E.coli* serogroup O1. It is filed under Accession GU299791 and was recently sequenced from *E.coli* strain G1632 that was isolated from a patient with urinary tract infection (Li *et al.*, 2010). An optimal local alignment between all 10 301 bp of the O1-antigen gene cluster and the genome of APEC_O1 resulted in an alignment comprising 10 300 bp with 11 mismatches and 3 single nucleotide indels. This high similarity between the O1-antigen region of a clinical isolate and an avian pathogenic *E.coli* is strong additional evidence that avian pathogenic *E.coli* can cause urinary tract infections in humans.

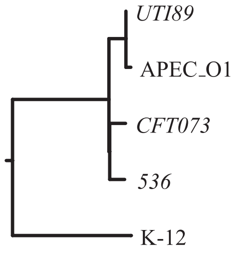


Fig. 6. Whole-genome phylogeny of the five *E.coli* strains investigated by Johnson *et al.* (2007) based on distances calculated using kr (Domazet-Lošo and Haubold, 2009). The three uropathogenic strains are *italicized*.

4 DISCUSSION

The 'conversion of data to knowledge' is perhaps the greatest challenge in the field of genomics (Brenner, 2010). This conversion depends on the availability of computational tools that match the

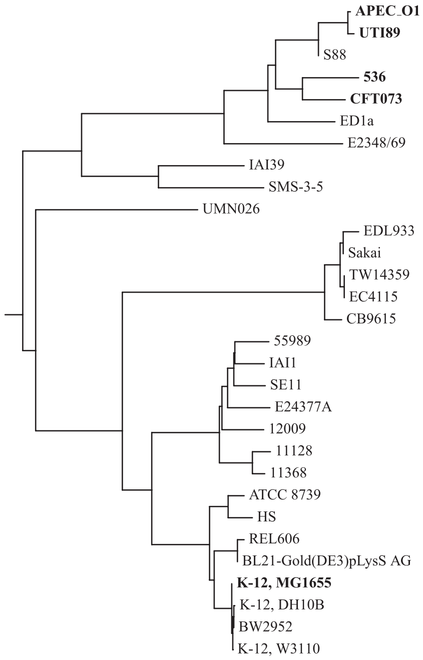


Fig. 8. Whole-genome phylogeny of 30 strains of *Escherichia coli* based on distances calculated using kr (Domazet-Lošo and Haubold, 2009). The five strains shown in Figure 6 are typeset in *bold*.

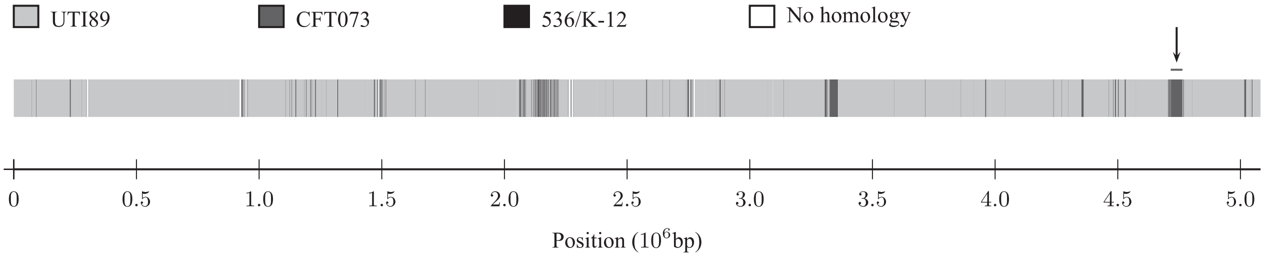


Fig. 7. The genome of *E.coli* strain APEC_O1 compared to the four other *E.coli* strains shown in Figure 6. The horizontal *bar* pointed to by the *arrow* indicates the 48 kb region transferred from CFT073; length of sliding window: 1 kb.

Table 2. Private regions of APEC_O1 when compared to the 29 strains of *E.coli* clustered in Figure 8

Start	End	Length
2 190 501	2 196 650	6150
2 066 251	2 069 600	3350
2 273 751	2 276 500	2750
1 513 501	1 515 850	2350
925 801	928 000	2200
943 201	945 150	1950
2 768 051	2 769 150	1100
1 474 551	1 475 650	1100
2 774 001	2 775 050	1050
2 745 551	2 746 550	1000

capabilities of the instruments used to gather the data in the first place. In the realm of sequence analysis, the first big advance came through optimal inexact matching methods (Needleman and Wunsch, 1970; Smith and Waterman, 1981). By combining fast exact matching with the inherently slower inexact matching techniques, fast programs like BLAST and its descendants were created (Altschul and Lipman, 1990). In this article, we take this expansion of the role of exact matching in sequence comparison to its logical conclusion by eliminating the inexact matching phase altogether when looking for close local homologs of a query sequence among a set of subject sequences. The result is not a substitute for residue-by-residue alignment but a guide to where this level of detail might be applied most profitably.

We based our program on the enhanced suffix array library by Manzini and Ferragina (2002) because it is known to be one of the most efficient implementations available (Puglisi *et al.*, 2007). As a result, enhanced suffix array construction in alfy remains fast even when analyzing sets of closely related bacterial genomes (Fig. 8). Such run time behavior is not guaranteed, as in the limit of sorting very long identical suffixes the library regresses to its worst case run time, which is quadratic in the number of characters. However, in our experience this has not been a problem in any of our applications ranging from HIV-1 genomes and near-identical *E.coli* genomes in this article to the complete genomes of 12 *Drosophila* species (Domazet-Lošo and Haubold, 2009).

In software based on suffix arrays, the time-consuming construction of the enhanced suffix array is often kept separate from its quick traversal. This makes it possible to apply the traversal to a pre-computed index. However, in alfy we regard each run as a one-off analysis during which the enhanced suffix array is computed on the fly. An implementation based on pre-computed suffix arrays would, of course, be much faster. For example, 80% of the time taken to compare APEC_O1 to 29 *E.coli* genomes is used for the construction of the enhanced suffix array. Given a pre-computed enhanced suffix array, the runtime of alfy could therefore be reduced up to 5-fold. More important than the exact runtime details of our particular implementation are therefore the program's general properties: for a single query, both its run time and its memory requirement scale linearly with the size of simulated subject sets (Fig. 2).

Our simulations in Figure 4 show that such algorithmically minimal utilization of time and memory is sufficient to obtain reasonably accurate detection of closest neighbors as long as the number of mutations in the sample is not too small or greater than ~0.2 substitutions per site between the query and its closest neighbor. Another requirement for accuracy is that the window size does not exceed the size of the shortest recombinant segment one wishes to detect. Otherwise, the annotation process is hampered by averaging shustring lengths over two or more regions of distinct ancestry. This consideration determined our choice of window size. In the case of HIV, 300 bp windows are also used by the NCBI genotyping tool. In the case of *E.coli*, we were interested in transfer events spanning at least 1 kb, hence the window size of 1 kb for that analysis.

Given these restrictions, alfy could classify recombinant group M HIV-1 strains as reliably as the NCBI tool (Rozanov *et al.*, 2004) and the program by Wu *et al.* (2007). It is certainly possible to achieve greater accuracy as shown by the phylogeny-based tool SCUEAL (Kosakovsky Pond *et al.*, 2009), albeit at the cost of escalating the runtime from 0.4 s to 6.9 h for the analysis of a single HIV-1 genome. While such extensive runtime requirements may be acceptable in patient care, it makes SCUEAL difficult to apply to the over 500 times larger genomes of bacteria like *E.coli*.

In our comparison between the avian pathogen APEC_O1 and other *E.coli* strains we first replicated the analysis by Johnson *et al.* (2007). This showed that APEC_O1 is most closely related to strain UTI86, which causes urinary tract infections in humans (Figs 6 and 7). Spliced into this 'clonal frame' (Milkman and McKane Bridges, 1990) are regions horizontally transferred from other *E.coli* strains, most notably the 48 kb or more obtained from a strain similar to CFT073, another human pathogen (Fig. 7).

Johnson *et al.* (2007) speculated that the APEC_O1-specific regions they found might be particularly helpful when searching for a link to human disease. We therefore compared APEC_O1 to the 29 *E.coli* genomes currently available (Fig. 8) and found that the longest private region is part of the gene cluster specifying the O1 serotype of APEC_O1. Fortuitously, the entire 10 kb covering the O1 region had just been sequenced in a clinical isolate (Li *et al.*, 2010) resulting in a match to the corresponding region in APEC_O1 with only one mismatch per kilobase. Since APEC_O1 infects chickens (Johnson *et al.*, 2007), this is further evidence that *E.coli* pathogens of poultry can infect humans.

The near identity of the O-antigen cluster in APEC_O1 and a clinical isolate could have been discovered by simply blasting the newly sequenced O1 antigen cluster against GenBank. However, our approach to study the regions private to APEC_O1 was the one suggested without the benefit of hindsight (Johnson *et al.*, 2007). Moreover, alfy directly identifies regions without strong homology, which makes it a convenient tool for this kind of analysis.

ACKNOWLEDGEMENTS

We thank Peter Pfaffelhuber and Angelika Börsch-Haubold for helpful comments on the manuscript.

Funding: M.D.-L. was supported by a doctoral fellowship of the Max-Planck Society and the Croatian Ministry of Science, Education and Sports (grant 036-0361983-2012 to Mirta Baranović).

Conflict of Interest: none declared.

REFERENCES

Abouelhoda,M. (2002) The enhanced suffix array and its applications to genome analysis. In *Proceedings of the Second Workshop on Algorithms in Bioinformatics*, *Vol. 2452 of Lecture Notes in Computer Science*, Springer, New York, pp. 449–463.

Altschul,S.F. and Lipman,D.J. (1990) Protein database searches for multiple alignments. *Proc. Natl Acad. Sci. USA* **87**, 5509–5513.

Brenner,S. (2010) Sequences and consequences. *Phil. Trans. R. Soc. B*, **365**, 207–212.

Cartwright,R. (2005) DNA assembly with gaps (Dawg): simulating sequence evolution. *Bioinformatics*, **21** (Suppl. 3), iii31–iii38.

Didelot,X. and Falush,D. (2007) Inference of bacterial microevolution using multilocus sequence data. *Genetics*, **175**, 1251–1266.

Didelot,X. *et al.* (2010) Inference of homologous recombination in bacteria using whole genome sequences. *Genetics*, **186**, 1435–1449.

Domazet-Lošo,M. and Haubold,B. (2009) Efficient estimation of pairwise distances between genomes. *Bioinformatics*, **25**, 3221–3227.

Felsenstein,J. (1989) PHYLIP - phylogeny inference package (Version 3.2). *Cladistics*, **5**, 164–166.

Ferragina,P. *et al.* (2008) Compressed text indexes: from theory to practice. *ACM J. Exp. Algorithmics*, **13**, 1.12:1–1.12:31.

Haubold,B. (2011) Alignment-free estimation of nucleotide diversity. *Bioinformatics*, **17**, 449–455.

Haubold,B. and Wiehe,T. (2006) How repetitive are genomes? *BMC Bioinformatics*, **7**, 541.

Haubold,B. *et al.* (2005) Genome comparison without alignment using shortest unique substrings. *BMC Bioinformatics*, **6**, 123.

Haubold,B. *et al.* (2009) Estimating mutation distances from unaligned genomes. *J. Comput. Biol.*, **16**, 1487–1500.

Johnson,T. *et al.* (2007) The genome sequence of avian pathogenic Escherichia coli strain O1:K1:H7 shares strong similarities with human extraintestinal pathogenic E.coli genomes. *J. Bacteriol.*, **189**, 3228–3236.

Kosakovsky Pond,S.L. *et al.* (2009) An evolutionary model-based algorithm for accurate phylogenetic breakpoint mapping and subtype prediction in HIV-1. *PLoS Comput. Biol.*, **5**, e1000581.

Langille,M.G.I. (2010) Detecting genomic islands using bioinformatics approaches. *Nat. Rev. Microbiol.*, **8**, 373–382.

Leitner,T. *et al.* (2005) HIV sequence compendium. *Technical Report LA-UR 06-0680*, Theoretical Biology and Biophysics Group, Los Alamos National Laboratory, NM.

Li,D. *et al.* (2010) A multiplex PCR method to detect 14 *Escherichia coli* serogroups associated with urinary tract infectcions. *J. Microbiol. Methods*, **82**, 71–77.

Manzini,G. and Ferragina,P. (2002) Engineering a lightweight suffix array construction algorithm. In *ESA '02: Proceedings of the 10th Annual European Symposium on Algorithms*. Springer, London, UK, pp. 698–710.

Maynard Smith,J. *et al.* (1991) Localized sex in bacteria. *Nature*, **349**, 29–31.

Milkman,R. and McKane Bridges,M. (1990) Molecular evolution of the *Escherichia coli* chromosome. III. Clonal frames. *Genetics*, **126**, 505–517.

Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.

Puglisi,S.J. *et al.* (2007) A taxonomy of suffix array construction algorithms. *ACM Comput. Surv.*, **39**, 4.

Reinert,G. *et al.* (2009) Alignment-free sequence comparison (i): statistics and power. *J. Comput. Biol.*, **16**, 1615–1634.

Rozanov,M. *et al.* (2004) A web-based genotyping resource for viral sequences. *Nucleic Acids Res.*, **32**, W654–W659.

Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.

Tettelin,H. *et al.* (2005) Genome analysis of multiple pathogenic isolates of *Streptococcus agalactiae*: implications for the microbial 'pan-genome'. *Proc. Natl Acad. Sci. USA*, **102**, 13950–13955.

Vinga,S. and Almeida,J. (2003) Alignment-free sequence comparison—a review. *Bioinformatics*, **19**, 513–523.

Westesson,O. and Holmes,I. (2009) Accurate detection of recombinant breakpoints in whole-genome alignments. *PLoS Comput. Biol.*, **5**, e1000318–e1000318.

Wu,X. *et al.* (2007) Nucleotide composition string selection in HIV-1 subtyping using whole genomes. *Bioinformatics*, **23**, 1744–1752.