Published online 6 June 2011

Nucleic Acids Research, 2011, Vol. 39, No. 15 ***e101***
doi:10.1093/nar/gkr177

An alternative approach to multiple genome comparison

Alban Mancheron, Raluca Uricaru and Eric Rivals*

LIRMM - CNRS, Université Montpellier 2 - CC 477, 161, rue Ada, 34095 Montpellier Cedex 5, France

Received May 11, 2010; Revised and Accepted March 14, 2011

ABSTRACT

Genome comparison is now a crucial step for genome annotation and identification of regulatory motifs. Genome comparison aims for instance at finding genomic regions either specific to or in one-to-one correspondance between individuals/strains/species. It serves e.g. to pre-annotate a new genome by automatically transfering annotations from a known one. However, efficiency, flexibility and objectives of current methods do not suit the whole spectrum of applications, genome sizes and organizations. Innovative approaches are still needed. Hence, we propose an alternative way of comparing multiple genomes based on segmentation by similarity. In this framework, rather than being formulated as a complex optimization problem, genome comparison is seen as a segmentation question for which a single optimal solution can be found in almost linear time. We apply our method to analyse three strains of a virulent pathogenic bacteria, Ehrlichia ruminantium, and identify 92 new genes. We also find out that a substantial number of genes thought to be strain specific have potential orthologs in the other strains. Our solution is implemented in an efficient program, QOD, equipped with a user-friendly interface, and enables the automatic transfer of annotations betwen compared genomes or contigs (Video in Supplementary Data). Because it somehow disregards the relative order of genomic blocks, QOD can handle unfinished genomes, which due to the difficulty of sequencing completion may become an interesting characteristic for the future. Availabilty: http://www.atgc-montpellier.fr/qod.

INTRODUCTION

The unprecedented sequencing capacity offered by high throughput sequencing technologies allows whole genome sequencing in short times and at low costs. Many projects aim at sequencing several genomes of a species or of a genus to infer functional and evolutionary knowledge from their genomic variability, among which the 1000 human genomes project (http://www.1000genomes.org) or the Microbial Genome Program of US Department of Energy (http://microbialgenomics.energy.gov). With the genomes at hand, the projects step ahead with a comparative genomic analysis to annotate genes, infer their homology/orthology relationships or reveal syntenic regions and rearrangement events. Broadly summarized, comparative genomics aims at identifying the genomic regions or organization that are either 'shared' among or 'specific' to individuals, strains and species. It is fundamental to the understanding of genomes structure and evolution. In bacteriology, the genomic part shared among all strains, the 'core' genome, represents the essential genic component, while the specific part, or 'dispensable' genome, abound in genes associated with genomic exchanges, which promote bacterial evolution (1). Core and dispensable genomes are computed using whole ORFeome/proteome comparisons (1,2) or whole genome alignments (3). In eukaryotes, large syntenic regions are detected either on DNA sequences using whole genome alignments computed with genome aligners (4,5) or more specialized programs (6), or by considering genomes as permutations of known orthologous genes and computing rearrangement distances (7). In both eukaryotic and prokaryotic cases, regulatory or chromosome maintenance motifs conserved across species are sought in aligned, shared genomic regions (8,9).

Among multiple applications, comparative genomics served to identify species-specific genes that could explain the capacity to cause disease. For instance, the comparison of *Candida dubliniensis* with the virulent

*To whom correspondence should be addressed. Tel: +33 4 67 41 86 64; Fax: +33 4 67 41 85 00; Email: rivals@lirmm.fr

Candida albicans exhibited specific expansion of some families of proteins, which became potential virulence-associated factors (10). Comparative analysis represent an important step in post-genomic vaccine design (11,12), in which shared, variable genes of multiple strains are selected for further immunization tests and may lead to design broadly protective vaccines (13). Similarly, comparison of three strains of the ruminants' pathogen, *Ehrlichia ruminantium*, underlined the importance of tandem duplication as a source of diversity and annotated some strain-specific genes (14). Not only may strain-specific genomic regions include virulence-related genes, but they can also serve to refine the diagnostic.

A wealth of computational methods have been designed to meet the needs of comparative genomics; most attempt to find 'shared' and 'specific' regions in the compared genomes. The two commonly used bioinformatic solutions for this sake rely on different information levels: the genome or proteome levels. The first involves comparing the ORFeomes/proteomes, as sets of proteins, to predict orthology with the reciprocal BLAST-hit approach, which is time consuming (15). Moreover, this requires the proteins to be correctly annotated. The second solution, 'whole genome alignment', is a computationally difficult optimization problem (7). Hence, heuristic alignment tools (4,5) usually build a highest scoring chain of local alignments and try to infer the evolution of each genome in terms of rearrangements (duplication, inversion, transposition), another NP-hard problem (7). The output alignment is sensitive to the method's parameters, the setting of which requires trained users. Consequently, even with a whole genome alignment at hand, which could be long and complex to obtain, it is not straightforward to determine the core genome of a bacterial species (3).

Note that neither the chaining nor the rearrangement inference steps solved during a whole genome alignment (and other methods) are necessary to determine the 'shared' and 'specific' parts of the genomes under consideration. Hence, whole genome alignment may be too involved for some goals in comparative genomics. This is the rationale that led us to propose a new formulation of multiple genome comparison to meet only this goal. We introduce a novel concept, 'maximum common intervals': a genome region that cannot be extended and is shared, i.e. alignable, across all genomes [MCI: not to be confounded with common intervals taken as subset of shared genes that colocalize in a region (7)]. The formulation is: given sets of pairwise local similarities between a 'target' genome and each other genome as input, compute all MCI of the target. Apart from avoiding the optimization of a numerical criterion (which often turns out to be NP-hard), and having few parameters, this formulation has another nice property: it can be solved exactly with a fast algorithm, which moreover yields a unique solution. The number of MCI covering a region also indicates whether several possible alignments exist for that region. Hence, the target genome segmentation induced by the MCI allows to partition regions into: unshared, shared with only one putative alignment and shared with several putative alignments, where the second category indicates possible orthology relationships.

Hence, we implemented an almost linear time algorithm to compute all MCI and the corresponding partition in QOD, a software equiped with a 'graphical user interface' (GUI), which provides a graphical overview of the multiple similarities on the target genome. If provided with the target's annotations, QOD intersects the partition with annotations and deduces from the MCI the pairwise alignment of each annotated feature. In potentially orthologous regions, QOD automatically selects well-conserved features and proposes them as potential annotation transfers to the user. QOD, which runs on all major computer platforms, is further equipped with many user-friendly options like word search, graphical/textual results or annotation transfers export, etc. As case study, we investigated the sequence relationships among all three strains of the ruminant pathogen bacteria, *E. ruminantium*, which have already been extensively compared using both proteome and whole genome comparisons (14,16). QOD's results enabled a deep revision of the set of genes annotated as strain specific and provided supporting information for annotating 92 novel genes altogether.

ALGORITHM AND METHODS

Here, we describe a novel approach to genome sequence comparison based on segmentation. We first define the key concept of the segmentation, the notion of 'maximum common interval', and formulate the segmentation algorithm that computes all MCI. We then expose the computation of annotation transfer and describe the tool implementing this algorithm, QOD and its pratical features (Figure 1 and Supplementary Figure S1).

Segmentation algorithm and maximum common intervals

The algorithm description requires a formal statement of the problem. We are given a 'target' genome T, which we need to compare with k other 'reference' genomes: $G_1,\ldots,G_k$. For each reference genome G_j with $1 \leq j \leq k$, we compute between T and G_j all local pairwise similarities whose statistical significance lies above a user-defined threshold. Each local alignment represents a pair of genomic intervals, one from T and one from G_j, that are aligned with each other. We consider the intervals on T of all those local pairwise similarities: they can be disjoint, overlap or even include each other on T. The intervals corresponding to the T versus G_j comparison form the collection $\mathcal{C}_j$ of 'base' intervals on T. The collections $\mathcal{C}_j$ for all $1 \leq j \leq k$, i.e. one per reference genome, make the input of our algorithm. We assume that the n_j intervals of $\mathcal{C}_j$ are ordered first by increasing beginning position in T, second by decreasing length. For $1 \leq i \leq n_j$, the i-th interval of $\mathcal{C}_j$ is denoted I_i^j (the superscript indicates the collection, and the subscript the interval index). The 'beginning' and 'end positions' of an interval I are denoted by $b(I)$ and $e(I)$, respectively. From the k input collections, $\mathcal{C}_j$ with $1 \leq j \leq k$, we want to compute all MCI, which we define below. This is the question answered by our algorithm.

Definitions. An interval J is 'common' to all $\mathcal{C}_{1 \leq j \leq k}$ if and only if for any collection $\mathcal{C}_j$ there exists an interval say I_i^j

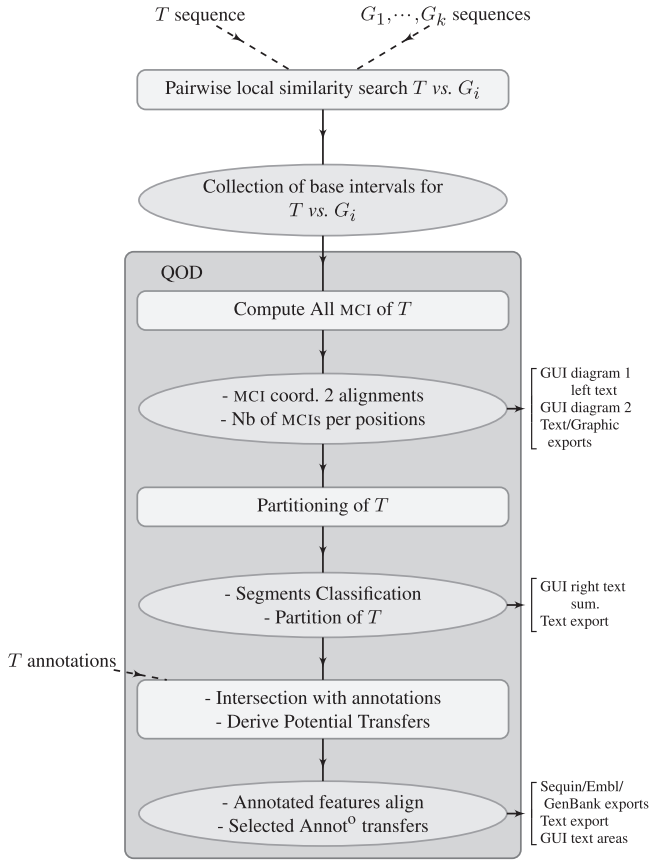

Figure 1. Workflow of a typical QOD analysis. Procedures are drawn in rectangles and intermediate results in ellipses, output locations and formats are shown right of the workflow. The first analysis part is performed outside QOD: local similarities are searched between the target genome T and each reference genome G_i for $1 \leq i \leq k$. This results is k collections of base interval pairs, since each local similarity puts in correspondance one interval of T with one of G_i. The interval pairs coordinates with the corresponding local alignments make up the input of QOD, and can be obtained using BLAST, BLAT or YASS. The second part of the analysis takes place inside QOD and includes 1/computing the MCI, 2/ partitioning T, 3/ intersecting the partition with input annotations and infering potential transfers. A dynamic version of the workflow with pointers to the output diagrams and text areas in the GUI is given in Supplementary Figure S1.

from $\mathcal{C}_j$ such that $J \subseteq I_i^j$. Assume $J = [p,q]$ with $p < q$ is a common interval. J is said to be a 'maximal' if neither $[p-1,q]$ nor $[p,q+1]$ are common intervals.

Basic properties. Several properties of MCI underlie our algorithm.

(1) It can be shown that an MCI beginning, respectively end, position is the beginning, respectively end, position of some base interval.
(2) As MCI are intersections of base intervals, which can be determined from the interval endpoints, it follows that one only needs to consider the base interval endpoints, not the base intervals themselves, to compute MCI.
(3) An MCI is the intersection of some base intervals, one from each collection. First, the MCI beginning (respectively end) position will be the largest beginning (respectively the smallest end) position of the base intervals overlapping the beginning position.
(4) Moreover, for a given position, if at least an interval of each collection includes it, there exists an MCI covering this position.
(5) Last, because of maximality any two MCI can be disjoint or overlap, but cannot include each other. As a corrollary, all MCIs the are totally ordered by increasing beginning position.

Algorithm overview. The basic properties lead to Algorithm 1, which computes all MCI in the order of increasing beginning position, and is illustrated dynamically in Supplementary Data. We scan in parallel (while loop lines 4–16) the base intervals of each $\mathcal{C}_{j:1\leq j\leq k}$ from left to right (in order) and consider a current 'reference base interval' (variable R). We determine whether its 'reference beginning position' (variable r) can be that of an MCI, and if such is the case, we compute the MCI end position (lines 12–14), then we iterate with the next possible reference interval. Meanwhile, we maintain in a 'heap' data structure (variable H) one current base interval of each $\mathcal{C}_{j:1\leq j\leq k}$ that is possibly overlapping the reference position; this structure allows to determine their intersection in constant time. The reference beginning position is initialized to the maximum among the starting positions of the first current interval in each collection, since before that at least one collection has no interval, and thus no MCI can start. The reference interval is set to the corresponding interval (line 2). The scan works as follows. Assume the current reference interval is I_m^j, which belongs to $\mathcal{C}_j$ for some $1 \leq j \leq k$ and $1 \leq m \leq n_j$. We need to determine whether in the other collections, $\mathcal{C}_{l:l\neq j;1\leq l\leq k}$, there is an interval overlapping $b(I_m^j)$ (for loop, lines 5–11). For this, we simply skip intervals until one satisfies this condition or lies at the right of the current reference interval (lines 6–10), and we update the heap appropriately (line 11; procedure `updateBranch` changes the interval of leaf l and updates the intersections in its branch). If (case 1) no interval of $\mathcal{C}_l$ for some $l{:}l\neq j$; $1 \leq l \leq k$ satisfies the condition, then we know that there exists no MCI overlapping $b(I_m^j)$, the current reference beginning position (Property 4). Moreover, the beginning position of the current interval of $\mathcal{C}_l$, say I_i^l, becomes the next current 'reference' position (line 9). Otherwise (case 2), if for each other collection one interval overlaps $b(I_m^j)$, we know that $b(I_m^j)$ is the MCI beginning position, and we determine its end position with the heap (Property 3; line 13). If I_i^l now denotes the interval with the smallest end position among all current overlapping base intervals, then $e(I_i^l)$ is the MCI end position. In case 2, we show that the current interval of $\mathcal{C}_l$ must be updated, and the next interval in $\mathcal{C}_l$, i.e. I_{i+1}^l, becomes the next reference interval/position. (The proof of this property is given in Supplementary Data.) Several collections may need to be updated; the procedure `updateHeapEndPoint` computes the MCI endpoint, determines these collections, updates their leaf and stores their indices in list L. Afterwards, we update the current interval of these

Algorithm 1: QOD's main algorithm

Input: k: number of reference genomes; $\forall 1 \leq j \leq k$: $\mathcal{C}_j$ the sorted collection of n_j base intervals, $\mathcal{C}_j := \{I_i^j : 1 \leq i \leq n_j\}$

Output: All MCI shared between the target and all reference genomes

Variables: R: current reference base interval; r: index of the collection to which R belongs; i_j: interval index in $\mathcal{C}_j$; H: heap of interval intersections

```
1  begin
2    r ← argmax_{1≤j≤k}(b(I_1^j));   R ← I_1^r;
3    i_j ← 1 for all 1 ≤ j ≤ k;
4    while AND_{1≤j≤k}(i_j ≤ n_j) do
5      forall l in [1,k] \ {r} do
6        while ((i_l ≤ n_l) and (e(I_{i_l}^l) < b(R))) do i_l++;
7        if (i_l > n_l) then return ;
8        if (b(I_{i_l}^l) > b(R)) then
9          r ← l;   R ← I_{i_l}^l;
10         break;
           // restart for loop from l=1
         // Invariant: I_{i_l}^l overlaps b(R)
11       H.updateBranch(l, I_{i_l}^l) // set leaf l to I_{i_l}^l
             & updates its branch
       // Invariant: there exists an MCI
       // starting in b(R)
12     b(M) ← b(R); L ← empty list;
13     e(M) ← H.updateHeapEndPoint(L, r);
14     output MCI M;
15     forall l in L do i_l++;
16     r ← argmin_{1≤j≤k}(b(I_{i_j}^j));   R ← I_{i_r}^r;
17 end
```

collections (line 15), and the reference interval (line 16). Note that except at the extremities, no beginning position can be skipped since all of them can be the endpoint of an MCI, as shown by the upper bound property in Supplementary Data.

To update the heap, we may face two alternatives. Denote by q the end of the last computed MCI, and I_{i+1}^l by I for simplicity. Either $b(I) \leq q$ then in all other collections the current interval overlaps $b(I)$, updating the heap to account for the change of current interval in $\mathcal{C}_l$ will assign the next MCI to the root node. Otherwise when $b(I) > q$, the updated heap indicates the collections whose current interval does not overlap $b(I)$ and require an update (line 13). The scan will resume for any of these collections (in any order), and may either show that an MCI starts in $b(I)$ or find a new reference position $> b(I)$.

Changing one leaf in the heap induces an update of internal nodes in $O(\log k)$ time. Hence, the algorithm has a complexity of $O((\sum_{1 \leq j \leq k} n_j)\log(k))$ time and $O(\sum_{1 \leq j \leq k} n_j)$ space, where n_j denotes the number of base intervals in $\mathcal{C}_j$.

Partitioning algorithm and regions classification

Once all MCI have been computed, QOD partitions the target genome into 'common' (classes 2 and 3) versus 'unshared' (class 1) regions. In a partition, every base belongs to a single region; hence, unlike MCI, the regions of the partition cannot overlap. To partition the target genome, QOD simply scans the endpoints of MCI from left to right and starts a new region at each position where the set of MCI covering this position changes compared to the previous one. The partitioning is fast: it takes a time proportional to the number of MCI. At the same time, QOD records all alignments associated with a region, which will serve for annotation transfer.

By combining the number of MCI covering a region and the number of possible multiple alignments of an MCI, QOD classifies the partition regions of the target regarding their similarity to the reference genomes into three categories: (i) 'unshared', those that cannot be aligned with all other genomes (ii) similar but with a unique possible alignment or (iii) similar with several possible alignments with at least another genome. As partition regions do not overlap, the genome coverage of each category delivers an overall measure of the target genome similarity with the reference genome set. Clearly, the coverage of similar regions (class 2 + 3) measures the core genome size, while that of class 2 regions indicate how much of the genome is shared and likely orthologous, provided the similarity level is high enough to ensure homology. Note the difference between unshared and 'genome-specific' regions. For instance, if three genomes are considered, a genome-specific region must be unshared in a three-way and in all two-way comparisons with that genome as a target. Both notions are nevertheless relative to the set of genomes under consideration.

Annotation transfer and visualisation

QOD is equipped with a structured GUI: Supplementary Figure S1 shows for each step of QOD's workflow where the results appear on the GUI. Beyond the capacity to give an overall measure and view of the target genome similarity with a set of reference genomes, the partition offers the possibility to determine which functional sequence elements are shared or genome specific (which may require several comparisons). If provided with the annotations of the target genome, QOD computes the inclusion/overlap of annotations with each partition segment and gathers its related annotations in those two categories (included/overlapping). Moreover, it extracts from the segment possible alignments the subalignments corresponding to annotated features and displays them on the GUI (cf. Supplementary Figure S1). All features located in uniquely aligned segments are marked as 'transferable' from the target to any other reference genome. The user can then select according to a minimal per cent of identity or from a feature list, which annotations he wants to transfer, and export the list in various formats (GenBank/Embl/Sequin, see Supplementary Figure S3).

The annotations related to a segment are displayed with the segment information and can be easily browsed with the GUI or output in tabular format.

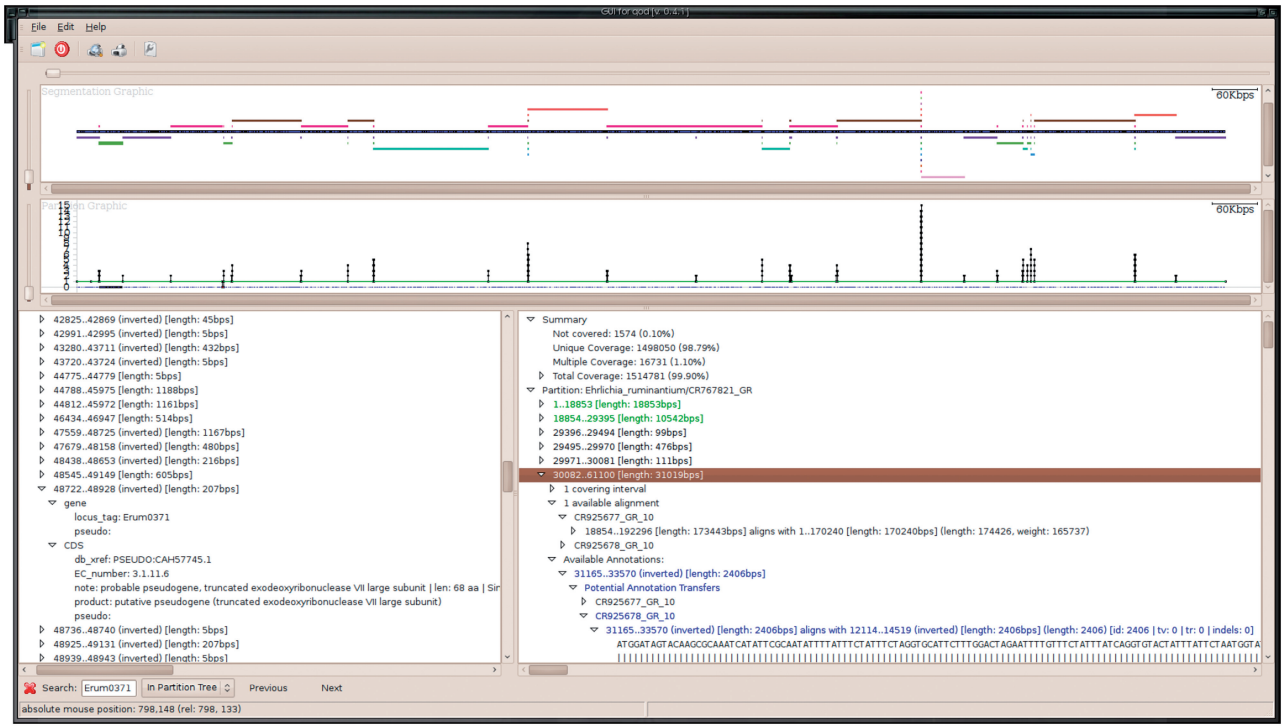

Figure 2. QOD GUI. It is divided in four parts: a upper and lower diagrams at the top, and below that two text subwindows: one left for the MCI description and one right for the partition description. The upper diagram displays the MCI relatively to their genomic coordinates and shows how they intersect. The lower diagram plots the number of overlapping MCI for each position along the genome. Both can be browsed and zoomed in parallel. In both the MCI and partition descriptions, the text is structured like a tree and each item can be displayed or hidden when browsing. The partition description includes a summary of the comparison at the top.

Pratical features and graphical interface of QOD

QOD implements efficient algorithms in ANSI C++ and is available under Cecill license for academic use. It incorporates a sophisticated but user-friendly GUI with many relevant features as described below. QOD can run several processes in parallel on the computer, which lets it exploit in a transparent manner the multi-core computer architectures for an improved efficiency. Moreover, it runs on multiple platforms: Mac OSX, Windows and Linux. When sets of comparisons are performed, the data (sequences, annotations, alignments) occupy large disk/storage resources, which can be reduced using data compression. To avoid the user time-consuming compression/decompression processes, QOD can directly read (gzip) compressed files. Moreover, QOD can export both images representing the genome maps (in a wide variety of image formats) and the MCI/segmentation information together with corresponding annotations (in text/PDF formats). The former can be used e.g. for illustrating the findings of a comparison, while the latter can be further analysed *in silico*. QOD incorporates a text search facility to look for some specific annotation.

Visualization in the GUI

QOD GUI is divided in four sections (Figure 2): two horizontal diagrams at the top give two parallel, zoomable and explorable maps of the target genome and two text sections. The upper diagram shows how the MCI cover the target genome, while the lower one shows how many MCI cover a given genomic interval; in both, the *x*-axis is the genome position. The left structured text details the MCI positions and related information (alignments, annotations), while the right one shows the genome segmentation and related information. In the upper diagram, the thick, black bar represents the target genome, while colour bars above and below it represent the MCI. The MCI position above or below is unrelated to the strand on which similarity was detected. MCI are drawn in lines such that all MCI on a line share the same colour, and two overlapping MCI are displayed on different lines. In the MCI text window, MCI keep the same colour as in the diagram. In the lower diagram, the *y*-axis gives the number of MCI covering a position. For a line at level 1, only a single alignment may be available for the unique MCI. In which case, this absence of ambiguity for homology is displayed by a green coloured line. When the line reaches level 0, meaning the absence of similarity, it is coloured in red. Again, this colour scheme is respected in the segmentation description (right text section). At any time, the interval corresponding to the current selection (MCI or feature) appears as a thick black line on the *y*-axis of the second diagram.

Evaluation method and data

We compare QOD with whole genome aligners widely used in the field, MAUVE and PROGRESSIVEMAUVE (4,17).

All approaches detect pairs of homologous regions and align them. On this basis, it is possible to determine potential annotation transfer of genes or coding regions (CDS) according to their alignment's percentage of identity (%id.). We apply the same procedure on the results of each tool to compute which annotations could be transferred from the 'target' onto a 'reference' genome. If the alignment of the feature's region reaches a percentage of identity higher than a predefined threshold, the feature is transferred. Then, we checked whether the transferred annotation falls in a true annotated feature of the other genome by comparing their genomic positions and allowing a small proportional difference. We consider that the transfer 'matches' the annotation if both the distances between their start, respectively end, positions is less than 2% of the gene/CDS size. The outcome of the test is as follows. If the gene is aligned, we encounter four situations: if the alignment reaches the %id. threshold, the gene is 'transferred', then if there is a matching gene on the reference, we count it as a true positive (TP), otherwise, either no feature or a non-matching gene, it is a false positive (FP); if the alignment %id. is below the threshold, the gene is not transferred, then if there is a matching gene on the reference, we count it as false negative (FN), and otherwise as a true negative (TN). Arbitrarily, we consider unaligned genes as TN (except for the Human–chimpanzee and simulated data sets, see below). The 'sensitivity' or TP rate (TPR) is the ratio TP/(TP + FN), while the FP rate is FP/(FP + TN). For each tool, we plot the receiver operating characteristic (ROC) curve, that is the TPR versus FPR for all %id. threshold values in [0, 100]. We applied these procedures on five data sets: strains of 1/ *Acinetobacter baumannii* with AC numbers CP000521, CP001182, CP000863, CU459141, CU468230, 2/ of *Lactococcus lactis* with AC numbers CP000425, AE005176, AM406671, 3/ of *Buchnera aphidicola* with AC numbers AE013218, AE016826, BA000003, CP001161, CP001158, CP000263, 4/ the longest contigs of Human and chimp chromosomes 21 (AC NT_011512, NT_106996) and 5/ four bacteriophages of the *Siphoviridae* family that infects *L. lactis* (P335, TP901, Tuc2009 and ul36 with AC DQ838728, AF304433, NC_002703, AF349457).

We also compared QOD, MAUVE and PROGRESSIVEMAUVE on data sets obtained by simulating random genome evolution from an ancestral genome along a known tree and allowing random substitutions, indels and inversions. For the simulation, we use the SIMALI software (18) with four genomes and parameters estimated on the above-mentioned bacteriophage data. In these cases, the test records whether the output local alignment blocks above a given %id. threshold coincide with the true alignments obtained by simulation with 2% tolerance on their positions. Results are shown on ROC curves as above. More details are available in Supplementary Data.

RESULTS

To demonstrate the utility of our approach, we first investigate a case study on bacterial strains for which genome annotation and comparison have been published, and compare QOD to well-known multiple genome aligners in terms of accuracy and running times.

As a case study, we evaluated QOD by comparing the three available strains of the bacterium *E. ruminantium*. The strains' genome sequences have been annotated and compared using 1/ whole proteome comparisons to determine orthology/paralogy relationships between proteins, and 2/ whole genome alignments to study the variability at the DNA level (14,16). Hence, this case study offers the opportunity to judge QOD relevance compared to standard methods in the field.

This obligate intracellular bacterium from the Rickettsiales order causes Heartwater disease in wild and domestic ruminants in the sub-Saharan Africa, in African and some Carribean islands. When affected by this fatal tick-borne disease, up to 90% of susceptible animals die within three weeks (19). The spread of *E. ruminantium* and Heartwater severely impacts the production of livestock in Africa, making it an important economical issue (20). Since current diagnostic tools and vaccines show a limited efficiency, partly due to genotypic variations, new targets need to be discovered (14). For this sake, a sequencing program was completed to determine and annotate the protein-coding repertoire (16), and then a comparative genomic analysis of three phenotypically different strains investigated the genomic evolutionary mechanisms of this *Rickettsia* (14). The genomes of three strains are now available: two Welgevonden, denoted Erwo (embl:CR767821; 1 516 355 bp) and Erwe (embl:CR925678; 1 512 977 bp), and a Gardel strain denoted Erga (embl:CR925677; 1 499 920 bp). Both Welgevonden strains originated from South Africa, but Erwe was maintained in the Guadeloupe Island for 18 years in a different cell environment (a naive goat). The Gardel strain was isolated in Guadeloupe from an infected goat (14). Both studies pointed out the surprisingly important proportion of tandem repeats (TRs) in both coding and non-coding regions, and suspected that repeat variation contributes greatly to genome adaptation in *E. ruminantium*. The comparative approach revealed that the three genomes under consideration are highly similar and mostly colinear, and it pointed out a number of strain-specific genes, which could be considered as potential targets for strain-specific diagnostic and/or vaccine design (14).

Comparisons of *E. ruminantium* strains: an overview

To assess QOD's capacity to reveal strain-specific genomic features, we compared three *E. ruminantium* strains. We first sought local similarities between any pair of strains using YASS (*E*-value threshold of 10) (21). We completed the following searches: Erwo versus Erwe, Erwo versus Erga and Erwe versus Erga. We then ran QOD to perform each two-way comparison and three multiple comparisons once with each strain as target: 1/ Erwo versus (Erwe, Erga), 2/ Erwe versus (Erwo, Erga) and 3/ Erga versus (Erwo, Erwe). First, the two-way Erwo versus Erwe (and converse) comparisons report that common segments cover 100% of either genome, and

Table 1. Overview of genome coverages for all two- and three-way comparisons

Comparison	Uncov.	Single align.	Multiple align.	Cov.	No. of uncov. regions
Erwo versus Erwe	0	99.36	0.64	100.00	0
Erwe versus Erwo	0	99.37	0.63	100.00	0
Erwo versus Erga	0.10	99.35	0.55	99.90	1
Erga versus Erwe	0.13	99.28	0.60	99.88	3
Erga versus Erwo	0.13	99.31	0.56	99.87	3
Erwo versus Erwe,Erga	0.10	98.79	1.10	99.89	1
Erwe versus Erwo,Erga	0.10	98.40	1.06	99.46	2
Erga versus Erwo,Erwe	0.13	99.21	0.67	99.88	4

Columns from left: (1) comparison, (2–5) genome percentages (%) in uncovered, single alignment, multiple alignment and covered regions, respectively, (6) number of uncovered intervals. The overall percentages of covered and of single alignment regions indicate the high degrees of similarity and synteny between the three strains. The low percentages and numbers of uncovered regions denote the few 'specific' regions in the genomes.

that 99.3% of the genome has a unique correspondence in the other one. Moreover, the genomes are mainly covered by 16 large, unique segments ranging in size between 11 and 237 kb, meaning that these are syntenic regions and suggesting that both strains likely shared their complete genomic repertoire (Supplementary Figure S2). The detailed coverages for all comparisons are summarized in Table 1.

In the three-way comparison with Erwo as a target (i.e. comparison #1), QOD yields a coverage by common segments of 99.9%, in complete agreement with the high similarity reported in Ref. (14) and the strains' evolution. In this shared genome part, unique segments covered 98.9%, while duplication accounts for 1.1%. This already gives a good insight into where the contour of *E. ruminantium* core genome may lie. However, the counterpart of unshared genomic regions, as estimated by QOD, contains a single 1540 bp long segment (0.1% of the genome), which is clearly insufficient to enclose all specific genes detected in Ref. (14). We thus inspected closely the regions where those genes lie and their feature alignments.

Investigation of strain-specific genes predicted by Frutos *et al.*

According to Ref. (14), Erwo, Erwe and Erga feature, respectively, 7, 28 and 22 strain-specific genes. We found that among the Erwo and Erwe supposedly specific genes, none is strain specific since all aligned well in the sister strain. More exactly, their DNA sequence aligns to 100% identity for 34 out of 35(=7+28) genes, and one gene aligns with a single difference. Thus, all those genes are shared and nearly identical in Erwo and Erwe genomes. Out of these 35 genes, 29(=4+25) align in Erga genome over their whole length with >90% identity, meaning that they are still present in Erga but may not be functional. We thus checked whether the corresponding DNA region in Erga genome encodes an open reading frame (ORF) that is longer than 80% of the original Erwe/Erwo's protein length. Out of these 29 shared genes, 19(=16+3) encode an ORF satisfying this criterion. The analysis of the 19 Erga's supposedly specific genes exhibits a similar situation: 17 genes align in both Erwe/Erwo genomes completely with >90% identity, and 11 encode a putative CDS satisfying our criterion. A summary of this analysis is given for each strain in Table 2, while all genes alignments and predicted ORFs are gathered in Supplementary Data.

Table 2. Strain-specific genes according to Ref. (14) and their homology at DNA level according to QOD

Erwo	Nb	Erwe	Not annot.	Erga	Not annot.	ORF >80%
100% id.	7	7	6			
>90% id.				4	6	3
Erwe	**Nb**	**Erwo**	**Not annot.**	**Erga**	**Not annot.**	**ORF >80%**
100% id.	28	27	20			
>90% id.		28		26	22	18
Erga	**Nb**	**Erwo**	**Not annot.**	**Erwe**	**Not annot.**	**ORF >80%**
100% id.						
>90% id.	22	17	18	17	18	10

For each strain, the columns list the number (Nb) of strain-specific genes, how many can be aligned with either 100 or >90% identity at DNA level in the other strain, how many regions lack annotation (not annotated) and last for those genes not 100% identical, how many encode an ORF longer than 80% of the original protein. For a vast majority of genes annotated as strain specific, QOD finds a homologue in the other strains.

Our results suggest that a large majority of the genes that were annotated as strain specific by Ref. (14) indeed have a homologue, and most of the time a likely ortholog, in the compared strains. Precisely, only three Erga genes appear to be unalignable and thus seem truly strain specific. Generally, the corresponding homologue is not anotated in the other strains. However, for instance between Erwe and Erwo, 7 of these genes were described on both strains but their homology was nevertheless missed. Altogether, only four genes in these subsets share an alignment with <50% identity between Erwo/Erwe and Erga strains; all other genes are still present in all three strains, either as pseudogenes or most of the time as likely functional genes. Moreover, we observed that the homologue alignments are included in the alignment of a single MCI that is tens of kb long, denoting a syntenic region, which provides a high confidence (see Supplementary Tables S3-1, S3-2 and S3-3). Altogether, our results suggest that based on highly significant alignments, 40, 24, 28 genes or pseudogenes could be newly annotated in Erwo, Erwe and Erga, respectively.

Comparison with multiple genome aligners

The *E. ruminantium* study shows the benefits of using QOD compared to standard approaches deployed in genome annotation and comparison projects. However, whole genome aligners represent another way of comparing genome sequences, and it is thus natural to assess how QOD compares to these tools. The assessment is not

trivial since the approaches differ and benchmarks are missing. QOD can exhibit several local alignments covering the same region, while an aligner chooses only one according to some criterion. Nevertheless, both types of tools seek to determine which pairs of regions are orthologous between the genomes and align them. Thus, if the reported orthologies are correct, the gene structure annotation could be potentially transferred from one genome to the other based on the reported alignments. We compared QOD with two widely used genome aligners, MAUVE (4) and PROGRESSIVEMAUVE (17), on their ability to correctly transfer annotations based on the DNA similarities they detect.

To consider a wide spectrum of species and several levels of divergence, we selected three bacterial, one viral, one eukaryotic real data sets and completed these with data sets of simulated genomes (for which we know the correct alignments).

In the case of the human–chimpanzee comparison, we could use the set of orthologous genes given by BioMart (22), while in the other comparisons a prediction was considered correct if the transfer matches an annotated feature at the predicted positions on the other genomes (cf. 'Results' section).

Figure 3 presents the comparison of QOD, MAUVE and PROGRESSIVEMAUVE on real (Figure 3a–e) and simulated data sets (Figure 3f). Results are presented as three ROC curves, one per tool. Each ROC curve plots one minus the specificity of the method or FPR, on the *x*-axis, versus its 'sensitivity' or TPR, on the *y*-axis, for varying thresholds of minimum per cent of identity. Of course, the lower the %id. threshold, the larger the number of considered alignments, the higher the sensitivity can be. ROC curves can be compared globally on the surface lying below the curve: the larger the surface the better the prediction. A point is interpreted as follows: for QOD, the point for 60% identity on Figure 3c is located at (0.03, 0.98) meaning that for this threshold QOD yields a specificity of 97% (i.e. 1–0.03) and a sensitivity of 98%. An optimal prediction would yield a point in (0, 1), while the worst would lie in (1, 0).

On bacterial and viral data sets, QOD performs well and better than both genome aligners. Its ROC curves cover larger areas and are almost vertical, meaning that its predictions remain valid whatever the %id. threshold. The transfer is performed pairwise, from one genome onto the other. However, the choice of the genome as origin of annotations has little effect on the ROC curves, even if some genomes are longer or contain more genes than others (see Supplementary Data). One sees that QOD gains in sensitivity without loosing much specificity when decreasing the %id. threshold, which is not the case of MAUVE and PROGRESSIVEMAUVE. For genomes with higher divergence levels (Figure 3c and e), MAUVE's and PROGRESSIVEMAUVE's results degrade rapidly with the %id. threshold, while QOD achieves near perfect prediction. It is noteworthy that in all cases, MAUVE and PROGRESSIVEMAUVE yield very low specificity ratio with low %id. thresholds (see the points at 20 or 0% identity), while QOD is robust to this parameter. Likely, MAUVE's and PROGRESSIVEMAUVE's alignments include regions with low percentage identities that match non-homologous regions of the genomes, and impact drastically their performances. At such thresholds, QOD outperforms both genome aligners. In practise, it means that with QOD almost all detectable orthologs are correctly transferred and these are polluted by very few false positive transfers. QOD's accuracy remains superior whatever the level of genomic divergence; comparatively its best results are obtained for the most divergent cases: on *B. aphidicola* and bacteriophages.

Even if all genomes shared an orthologous protein in a region, a transfer may not match the corresponding item on the other genome, and therefore be counted as a false positive, if their sequences have evolved. In the bacteriophages comparison, it is the case of `ORF44` in TP901 and `ORF47` in Tuc2009, which are transferred on all other genomes, but are counted as false positives on ul36 and P335. Indeed, the genes exist in the corresponding regions in all genomes, all four proteins are orthologous and similar in sequence, except that those encoded by `ORF40` in P335 and `ORF89B` in ul36 lack 20 amino acids in their N-terminal region. Hence, the transferred gene does not match the features on these genomes because the start positions are too distant. However, the DNA alignment underlying the transfer detects the similarities over the region corresponding to the longest genes of TP901 and Tuc2009, and can therefore predict that mutations have altered the start codons and resulted in shorter proteins. Thus, DNA comparison provides useful information for understanding the evolutionary and functional differences between these proteins.

On the Human versus chimpanzee comparison, see Figure 3d, all tools deliver moderate results, which can be due to the fact that genes and orthology relationships are more variable and difficult to annotate than in prokaryotes. Nevertheless, QOD yields the best results (its ROC curve departs the most from the diagonal).

We also compare these tools on four simulated data sets in which the compared genomes have evolved along a tree from an ancestral genome, and where we know the correct alignments. Results obtained on all simulated data sets indicate that MAUVE and QOD offer similar performances, and perform generally better than PROGRESSIVEMAUVE, as illustrated in Figure 3f where the curves of MAUVE and QOD are nearly superimposed.

On real and simulated data sets, QOD superiority proves to be robust to the genome used as source of annotations, as well as to the tolerance threshold (see Supplementary Figures S1-3, S1-4, S1-5 and S1-6).

For the Human versus chimpanzee comparison, MAUVE takes 9 min, PROGRESSIVEMAUVE 12 min, while YASS takes 66 min and QOD takes alone 4 min. Highly sensitive local alignment search is time consuming for long eukaryotic genomes. However, it can be drasctically increased by choosing somehow less sensitive, but much faster tool (such as BLAT).

Finally, Figure 3g reports the running times of the different methods on all bacterial data sets. For our approach, the time includes both searching the local alignments for all pairwise comparisons (YASS), changing their format (YASS2AXT) and the computation made by QOD.

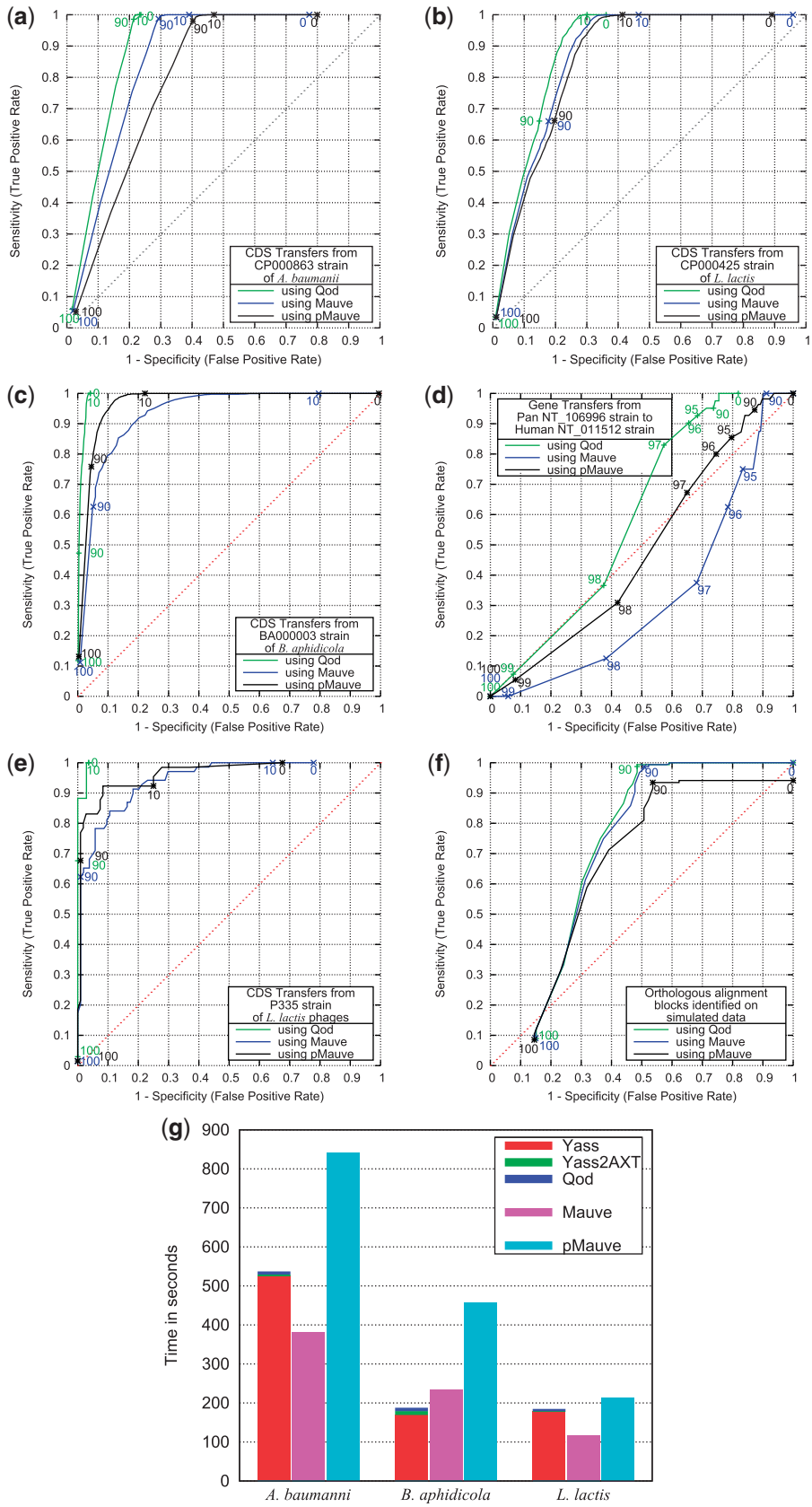

Figure 3. Comparison of QOD with multiple genome aligners. Whole genome aligners and QOD ability to identify orthologous regions on bacterial (**a**)–(**c**), eukaryotic (**d**), viral (**e**) and simulated data sets (**f**). All results are plotted as ROC curves: plotting (1 − specificity) versus the sensitivity for varying threshold of minimum percentage identity of the considered alignments. In all cases, the ROC curves of QOD cover a larger surface than that of MAUVE and PROGRESSIVEMAUVE, denoting its superior performance in identifying orthologous regions. Notably, with low per cent of identity thresholds, MAUVE and PROGRESSIVEMAUVE show a degraded specificity [see points at 0% in (a)–(e)], while QOD remains very accurate. At those %id. values, QOD outperforms these genome aligners. In (**g**), we plot the total running times for all methods on all bacterial data sets.

In general, all methods run in reasonable and similar times, with QOD ranging between MAUVE and PROGRESSIVEMAUVE. For instance, the comparison of five genomes of *A. baumannii* (average length of 3.85 Mbp) MAUVE takes 380 s., PROGRESSIVEMAUVE 820 s. and QOD 530 s. in average depending on the reference genome. For all data sets, about 95 percents of the time counted for QOD is in fact taken by the local similarity detection done with YASS. The running times for MAUVE and PROGRESSIVEMAUVE seem to be most influenced by the number of genomes.

DISCUSSION

Summary and advantages of QOD

Here, we proposed a novel approach for multiple genome comparison based on segmentation. Given sets of pairwise local similarities between a target genome and each of k reference genomes, it determines which regions of the target are both shared among all genomes and maximal (in the sense that they cannot be extended neither to the left nor to the right). Such regions are called MCI and any base may belong to one or more MCI, or to none. Not being covered by an MCI means that QOD did not detect a k-way homology for the region considered (class 1). A target region covered by a single MCI that has a single possible alignment indicates an unambiguous k-way homology, i.e. a likely orthologue (class 2). In all other cases, the region is involved in several possible multiple alignments indicating that it was duplicated in some reference genome and that additional investigation is required to determine orthology/paralogy (class 3). QOD outputs the genome partition and regions classification, which are highly informative.

Our approach is fast, independent of annotations and does not require phylogenetic information. For they process interval bounds, the MCI and partition procedures run in few seconds or minutes. Compared to whole genome multiple alignment (4,5,7) or rearrangement distance approaches (7), it avoids to solve some questions that render these computationally difficult (7): choosing a score optimal multiple alignment or sequence of rearrangements. Two main reasons underlie this choice. First, solving these questions is usually done by optimizing some criterion that may not give the biologically most relevant solution. Second, this information may not be needed for certain applications, for instance when focusing on genome-specific regions or inferring orthologs.

QOD approach is based on local similarities, which gives it another advantage over multiple alignment and rearrangement distances: it can compare unfinished or incompletely assembled genomes whose relative order of contigs is unknown. Genome finishing is a long, complex, and expensive step, which may be avoided in future genome projects. Hence, the capacity to compare an unfinished genome is an issue, to which QOD brings a novel solution. An illustration of such a comparison is shown in Supplementary Data. Consider the comparison of a complete genome as target and the concatenation in an arbitrary order of the contigs of an unfinished genome as reference. For example, two MCI covering adjacent (or slightly overlapping) target regions may help ordering two different contigs. It can also tell whether a gene from the target is partially or completely conserved in one or more contigs of the unfinished genome, and whether the latter encodes some paralogs. The reverse comparison (i.e. the unfinished genome as target and complete genome as reference) can help locating a rearrangement boundary if adjacent MCI on the target match distant regions in the reference. Additionally, QOD is used to annotate single contigs of a new genome by a comparative approach (data not shown).

Our results suggest that QOD compared favourably to two widely used genome aligners, especially in the cases of highly divergent genomes. QOD outputs alignments between genomic regions that allow accurate and sensitive transfers of annotations whatever the chosen threshold of identity percentage. Comparatively, the outputs of MAUVE and PROGRESSIVEMAUVE include alignments with low %id. between non-orthologous regions that induce false annotation transfers. In the case of PROGRESSIVEMAUVE, this is surprising since it includes a filtration step that removes low-quality alignments (17). The better performance of QOD observed on bacterial, eukaryotic and viral genomes is partly, but not only, due to the use of spaced seeds in the local alignment search (21), since PROGRESSIVEMAUVE also relies on spaced seeds (17). We claim that the advantage comes from combining maximum common intervals (i.e. the presence of a local similarity) and the number of overlapping MCI for predicting that two regions are orthologous. This is consistent with the better specificity obtained on prokaryotic versus eukaryotic genomes, since the former are less repetitive than the latter. In conclusion, QOD offers satisfactory performance with diverse data sets: eukaryotic or prokaryotic, pairwise versus multiple, closely or more distantly related genomes.

Considering its user-friendly interface, QOD may prove a handy tool to practise comparative genomics for both research and educational purposes. Future development will aim at adapting QOD interface to easily handle genomes with multiple contigs or chromosomes, especially for large eukaryotic genomes.

Novel gene homologies among *E. ruminantium* strains

Our analysis of three strains of the pathogenic bacteria *E. ruminantium* delivered a more precise view of their genome similarities. The pairwise comparison of Erwo with Erwe strongly suggests that they share all their genomic repertoire. Moreover, for most genes reported as strain specific (14), we exhibit significant conservation at the DNA level among all strains (35/35 for the Erwe/Erwo pair, 30/35 for Erga versus Erwe/Erwo, 17/22 for Erwo/Erwe versus Erga). A majority of those still give rise to a putative, long ORF in the strains supposed to lack those genes. Notably, the homology/orthology status of all these genes could be clarified or corrected based on this analysis. Knowing that strain-specific genes are potentially involved in host–pathogen interactions, and are further investigated as diagnostic or potential drug

targets, we believe that the new homology relationships identified here are of theoretical as well as practical relevance. Moreover, a total of 92 yet unknown genes could be newly annotated in those strains.

It is noticeable that most of genes that were annotated as strain specific by Ref. (14) are short (<300 bp, see Supplementary Data). Like many others, this study mainly based homology/orthology inference on comparisons at the protein level, from which ORF not exceeding an arbitrary threshold length are often excluded. Despite the use of whole genome comparisons at the DNA level, this might explain why those DNA similarities were missed and why annotations lack mention of these homologies. Another reason lies in the fact that some ORFs were probably missed in either genome due to the parameter used when processing shorter ORFs (14,16). The sensitivity of YASS and its ability to report long alignments also partly explains why we could detect those homologies with QOD. Only the comparisons at both the DNA and proteome levels allow to distinguish between absent genes and pseudogenes, both of which are not translated into protein and thus not considered at the proteome level. Our report illustrates the pitfalls of using only proteome comparisons for orthology prediction, even in a case of highly similar genomes. Altogether, it suggests that QOD is a practical, sensitive and complementary tool for annotation and comparison of whole genomes, and may suit future needs of comparative genomics.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We gratefully thank H. Chiapello and C. Lemaitre for careful reading.

FUNDING

This work, AM and RU, were supported by the French National Research Agency (CoCoGen project) [BLAN07-1_185484]. AM and RU were also supported respectively by CNRS (Centre national de la recherche scientifique) and ReNaBi (French bioinformatics platforms network) fundings and a grant from French Minister for Research and Education. Funding for open access charge: French National Research Agency (CoCoGen Project) [BLANO7-1 185484].

Conflict of interest statement. None declared.

REFERENCES

1. Tettelin,H., Masignani,V., Cieslewicz,M.J., Donati,C., Medini,D., Ward,N.L., Angiuoli,S.V., Crabtree,J., Jones,A.L., Durkin,A.S. *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.
2. Huynen,M. and Bork,P. (1998) Measuring genome evolution. *Proc. Natl Acad. Sci. USA*, **95**, 5849–5856.
3. Chiapello,H., Bourgait,I., Sourivong,F., Heuclin,G., Gendrault-Jacquemard,A., Petit,M.-A. and El Karoui,M. (2005) Systematic determination of the mosaic structure of bacterial genomes: species backbone versus strain-specific loops. *BMC Bioinformatics*, **6**, 171.
4. Darling,A.C., Mau,B., Blattner,F.R. and Perna,N.T. (2004) Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome Res*., **14**, 1394–1403.
5. Brudno,M., Malde,S., Poliakov,A., Do,C.B., Couronne,O., Dubchak,I. and Batzoglou,S. (2003) Glocal alignment: finding rearrangements during alignment. *Bioinformatics*, **19**, i54–i62.
6. Lemaitre,C., Tannier,E., Gautier,C. and Sagot,M.-F. (2008) Precise detection of rearrangement breakpoints in mammalian chromosomes. *BMC Bioinformatics*, **9**, 286.
7. Pevzner,P. (2000) *Computational Molecular Biology*. MIT Press, Cambridge, MA.
8. Kolbe,D., Taylor,J., Elnitski,L., Eswara,P., Li,J., Miller,W., Hardison,R. and and Chiaromonte,F. (2004) Regulatory potential scores from genome-wide three-way alignments of human, mouse, and rat. *Genome Res*., **14**, 700–707.
9. Halpern,D., Chiapello,H., Schbath,S., Robin,S., Hennequet-Antier,C., Gruss,A. and El Karoui,M. (2007) Identification of dna motifs implicated in maintenance of bacterial core genomes by predictive modeling. *PLoS Genet*., **3**, e153.
10. Jackson,A.P., Gamble,J.A., Yeomans,T., Moran,G.P., Saunders,D., Harris,D., Aslett,M., Barrell,J.F., Butler,G., Citiulo,F. *et al*. (2009) Comparative genomics of the fungal pathogens Candida dubliniensis and Candida albicans. *Genome Res*., **19**, 2231–2244.
11. Serruto,D., Serino,L., Masignani,V. and Pizza,M. (2009) Genome-based approaches to develop vaccines against bacterial pathogens. *Vaccine*, **27**, 3245–3250.
12. Serruto,D. and Rappuoli,R. (2006) Post-genomic vaccine development. *FEBS Lett*., **580**, 2985–2992.
13. Maione,D., Margarit,I., Rinaudo,C.D., Masignani,V., Mora,M., Scarselli,M., Tettelin,H., Brettoni,C., Iacobini,E.T., Rosini,R. *et al*. (2005) Identification of a universal group B streptococcus vaccine by multiple genome screen. *Science*, **309**, 148–150.
14. Frutos,R., Viari,A., Ferraz,C., Morgat,A., Eychenie,S., Kandassamy,Y., Chantal,I., Bensaid,A., Coissac,E., Vachiery,N. *et al*. (2006) Comparative genomic analysis of three strains of Ehrlichia ruminantium reveals an active process of genome size plasticity. *J. Bacteriol*., **188**, 2533–2542.
15. Tatusov,R.L., Koonin,E.V. and Lipman,D.J. (1997) A genomic perspective on protein families. *Science*, **278**, 631–637.
16. Collins,N.E., Liebenberg,J., deVilliers,E.P., Brayton,K.A., Louw,E., Pretorius,A., Faber,F.E., vanHeerden,H., Josemans,A., vanKleef,M. *et al*. (2005) The genome of the heartwater agent Ehrlichia ruminantium contains multiple tandem repeats of actively variable copy number. *Proc. Natl Acad. Sci. USA*, **102**, 838–843.
17. Darling,A.E., Mau,B. and Perna,N.T. (2010) progressiveMauve: multiple genome alignment with gene gain, loss and rearrangement. *PLoS ONE*, **5**, e11147.
18. Blanchette,M., Green,E., Miller,W. and Haussler,D. (2004) Reconstructing large regions of an ancestral mammalian genome in silico. *Genome Res*., **14**, 2412–2423.
19. Totte,P., McKeever,D., Martinez,D. and Bensaid,A. (1997) Analysis of T-cell responses in cattle immunized against heartwater by vaccination with killed elementary bodies of Cowdria ruminantium. *Infect. Immun*, **65**, 236–241.
20. Mukhebi,A.W., Chamboko,T., O'Callaghan,C.J., Peter,T.F., Kruska,R.L., Medley,G.F., Mahan,S.M. and Perry,B.D. (1999) An assessment of the economic impact of heartwater (Cowdria ruminantium infection) and its control in Zimbabwe. *Prev. Vet. Med*., **39**, 173–189.
21. Noé,L. and Kucherov,G. (2005) YASS: enhancing the sensitivity of DNA similarity search. *Nucleic Acids Res*., **33**, W540–W543.
22. Smedley,D., Haider,S., Ballester,B., Holland,R., London,D., Thorisson,G. and Kasprzyk,A. (2009) BioMart - biological queries made easy. *BMC Genomics*, **10**, 22.