BIOINFORMATICS ORIGINAL PAPER

Vol. 29 no. 7 2013, pages 837–844
doi:10.1093/bioinformatics/btt049

Sequence analysis

Advance Access publication January 30, 2013

# MosaicFinder: identification of fused gene families in sequence similarity networks

Pierre-Alain Jachiet[1,†], Romain Pogorelcnik[2,†,*], Anne Berry[2], Philippe Lopez[1] and Eric Bapteste[1]

[1]UMR CNRS 7138 Systématique, Adaptation, Evolution, Université Pierre et Marie Curie, 75005 Paris, France and [2]LIMOS, Ensemble Scientifique des Cezeaux, 63173 AUBIERE, France

Associate Editor: Mario Albrecht

## ABSTRACT

**Motivation**: Gene fusion is an important evolutionary process. It can yield valuable information to infer the interactions and functions of proteins. Fused genes have been identified as non-transitive patterns of similarity in triplets of genes. To be computationally tractable, this approach usually imposes an *a priori* distinction between a dataset in which fused genes are searched for, and a dataset that may have provided genetic material for fusion. This reduces the 'genetic space' in which fusion can be discovered, as only a subset of triplets of genes is investigated. Moreover, this approach may have a high–false-positive rate, and it does not identify gene families descending from a common fusion event.

**Results**: We represent similarities between sequences as a network. This leads to an efficient formulation of previous methods of fused gene identification, which we implemented in the Python program *FusedTriplets*. Furthermore, we propose a new characterization of families of fused genes, as clique minimal separators of the sequence similarity network. This well-studied graph topology provides a robust and fast method of detection, well suited for automatic analyses of big datasets. We implemented this method in the C++ program *MosaicFinder*, which additionally uses local alignments to discard false-positive candidates and indicates potential fusion points. The grouping into families will help distinguish sequencing or prediction errors from real biological fusions, and it will yield additional insight into the function and history of fused genes.

**Availability**: *FusedTriplets* and *MosaicFinder* are published under the GPL license and are freely available with their source code at this address: http://sourceforge.net/projects/mosaicfinder.

**Contact**: pogorelc@isima.fr

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

Received on July 30, 2012; revised on January 4, 2013; accepted on January 27, 2013

## 1 INTRODUCTION

### 1.1 Biological and evolutionary motivation for studying gene fusion

Fused genes, which result from the fusion of previously separate genes, or parts of their sequences, are key evolutionary entities (Patthy, 2003). It is generally believed that such events are rare, and that the resulting genes are often deleterious. However, fused genes can also encounter evolutionary success (Rogers and Hartl, 2012). Gene fusions have been reported in the three domains of life, e.g. in (hyper)thermophilic Archaea (Rodrigues *et al.*, 2007), in bacteria (Nie *et al.*, 2011; Pasek *et al.*, 2006) and in Eukaryotes (Durrens *et al.*, 2008; Ekman *et al.*, 2007; Zhou *et al.*, 2008). As a matter of fact, it has been estimated that two-fifths of the prokaryotic genes and more than two-thirds of the eukaryotic genes are composed of several domains (Han *et al.*, 2007), which have been likely combined through fusion events. In the latter taxa, gene fusions have been particularly well documented in animals (Buljan *et al.*, 2010; Marsh and Teichmann, 2010). In particular, it was shown that domain rearrangements occurred in 35.9% of gene families within the *Drosophila* clade (Wu *et al.*, 2011), significantly affecting processes of signalling and development. More problematically, in humans, gene fusions were reported to play a role in cancer. These fusions notably concerned relatively large and conserved genes (Narsing *et al.*, 2009) and members of the rapidly accelerated fibrosarcoma family of protein kinases, recently identified as characteristic aberrations of the most common tumours of the central nervous system in children (Lawson *et al.*, 2011). The widespread occurrence of gene fusion is notably explained by the fact that gene fusion can lead to new functions (Long, 2000). For instance, in the ciliate unicellular eukaryote *Tetrahymena thermophila*, gene fusions contributed to the evolution of processes, such as phospholipid synthesis, nuclear export and surface antigen generation (Salim *et al.*, 2011). Likewise, a gene fusion occurred in the early history of fungi, resulting in cellobiose dehydrogenases involved in the degradation of cellulose and lignin (Zamocky *et al.*, 2004). Later, the fungi *Candida albicans* benefited from the fusion of the 5′ domain of ALS5 (agglutinin-like sequence 5) to the tandem repeat region and 3′ domain of ALS1 producing an original ALS protein, likely involved in the adhesion to host and abiotic surfaces (Zhao *et al.*, 2011). As some of these fused genes increased the fitness of their carrier, they were maintained in genomes and gave rise to new gene families. Thus, various

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

fused globins have been reported, occasionally supplanting the parental gene form, as was the case for the fusion of $\beta/\Delta$ globin, before the radiation of *Paenungulata* (the clade containing elephants, dugongs and manatees and hyraxes) (Opazo *et al.*, 2009). Similarly, some of the toxins exploited by sea anemones to paralyse their preys have evolved by gene fusion, as they improved the transcript stability and secretion of these toxins (Moran *et al.*, 2009). From an evolutionary perspective, fused gene families can convey useful information about the history of life. They can provide valuable markers for phylogenetic analysis. It was suggested that these slow and rare events could be informative for reconstructing the phylogeny of plants (Nakamura *et al.*, 2007). Moreover, Stechmann and Cavalier-Smith (2002) used a derived gene fusion to propose a rooting of the eukaryotic tree. However, convergences and lateral transfers of gene fusions have also been reported, e.g. both in eukaryotes (Abdelnoor *et al.*, 2006; Aleshin *et al.*, 2007; Makiuchi *et al.*, 2007) and in diverse bacterial phyla, where gene fusions of polyamine biosynthetic enzymes *S*-adenosylmethionine decarboxylase (AdoMetDC, speD) and aminopropyltransferase (speE) orthologues, catalysing *de novo* diamine to triamine formation (Green *et al.*, 2011), and fused genes involved in histidine biosynthesis have been laterally transferred (Fani *et al.*, 2007 and so forth), suggesting that fused genes should only be used as phylogenetic markers with great care (Waller *et al.*, 2006). Finally, from a functional perspective, fused gene families serve as precious 'Rosetta stones' (Adai *et al.*, 2004) for the identification of potential protein–protein interactions and metabolic or regulatory networks (Enright *et al.*, 1999; Marcotte *et al.*, 1999). Our purpose in this article is to propose a new method for finding fused genes and to group them into families, which yields additional insight into the function and history of these genes.

## 1.2 Fused gene detection: state of the art

All current *in silico* methods for finding fused genes are based on sequence similarities (Durrens *et al.*, 2008; Enright *et al.*, 1999; Marcotte *et al.*, 1999; Rogers *et al.*, 2009; Salim *et al.*, 2011; Snel *et al.*, 2000; Suhre, 2004). The idea is that a fused gene (or *composite gene*) is similar to two *component* genes, which are not pairwise similar and align on disjoint parts of the fused gene (Fig. 1). In the rest of the article, we will use these terms of *composite* and *component* genes, as proposed in Enright *et al.* (1999). We will designate as a *fused triplet* a triplet of genes that exhibits this non-transitive pattern of similarity. Many variations around this idea have been implemented to identify composite genes and their components since the Marcotte *et al.* and the Enright *et al.* 1999 articles. They encounter four types of issues.

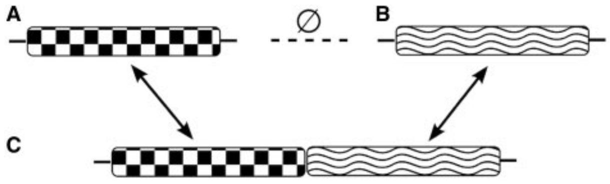

**Fig. 1.** Composite (fused) gene C and its two components A and B. A and B are similar to disjoint parts of C. A and B are dissimilar

First, the number of fused triplets rapidly becomes enormous for big datasets. Previous authors usually distinguished *a priori* between a query dataset (genome), within which composite genes were searched for, and a reference dataset (genomes, Clusters of orthologous groups of proteins (COGs)), in which components could be found. This greatly reduced the number of candidate triplets, with the drawback that some triplets are missed, as only a subset is investigated.

Second, some triplets may not result from a fusion but from distant homologues, i.e. a pair of homologous sequences that display no similarity at the sequence level, but that are both similar to a third intermediate sequence (Park *et al.*, 1997). Two types of tests are usually performed to exclude those false positives. The first test cross-checks that component genes are not similar, either with the same algorithm at a more permissive threshold [most of the time a higher Basic Local Alignment Search Tool (BLAST) E-value (Yanai *et al.*, 2001)] or with a more accurate algorithm (Enright *et al.*, 1999) such as Smith–Waterman (Smith and Waterman, 1981). The second test checks whether component genes align along non-overlapping regions of the candidate composite genes (Enright and Ouzounis, 2001; Yanai *et al.*, 2001). These controls eliminate many false positives.

Third, strongly supported fused triplets may result from sequencing or prediction errors (Pasek *et al.*, 2006), if a gene is artificially split into two separate genes, or if two adjacent genes are artificially fused into a single one. As those errors are presumably random and rare, a control is to identify other occurrences of candidate composite and component genes in closely related genomes.

A fourth and central issue is the grouping of identified component and composite genes *a posteriori* into gene families descending from a common fusion event. This grouping is necessary to count evolutionary events and to perform general functional analyses. First, if one could group composite triplets descending from a common fusion event, it would summarize the information contained in this enormous number of triplets of genes into fewer triplets of gene families, and, therefore, avoid long post-analyses of the results. This is, however, far from obvious and computationally challenging. Second, grouping into families would reduce the risk of distant homologies, as the absence of similarity between any pair of genes from two component families is much more robust than the absence of similarity between two component genes. Third, potentially artefactual composite or component genes would be easily identified, as they are the only representatives of their family (trivial family of size one).

A proxy to achieve a grouping into families has been to map genes on pre-existing family classifications (Suhre, 2004; Yanai *et al.*, 2001), usually Clusters of orthologous groups of proteins (COG)/Clusters of orthologous groups of eukaryotic proteins (Tatusov *et al.*, 2003). This is only partially satisfactory, as by definition, families of composite genes do not match a single COG and, therefore, are overlooked by that approach. Novel gene families (e.g. environmental) that have not been associated to a COG family are likewise difficult to detect. Alternatively, Enright and Ouzounis (2000) grouped composite genes into families by simple linkage. This is straightforward, as similarity between sequences is already computed to look for fused triplets. But simple linkage will aggregate unrelated composite genes if multiple fusion events have occurred in the history of some genes (Supplementary Fig. S1). Moreover, this method does not allow reconstructing component gene families and their relation with composite families.

### 1.3 Fused gene detection: our approach

We propose to explicitly represent similarity between DNA or protein sequences (hereafter called genes) as a network. Sequence similarity networks were first proposed in a study conducted by Tatusov *et al*. (1997) and used for larger scale studies in the study conducted by Enright *et al*. (2002). This approach enables to apply efficient graph theory concepts and tools to mine similarity information (Atkinson *et al*., 2009; Halary *et al*., 2009; Song *et al*., 2008; Tordai *et al*., 2005). We propose a new characterization of families of composite genes, with a robust and fast method of detection, well-suited for the automatic analysis of large datasets, without using an *a priori* distinction on the datasets from which families of composite genes may be identified.

We also unify the existing methods for composite gene detection by transposing them into a sequence similarity network. This enables us to compare our new tool called MosaicFinder with the existing gene-centred approach, which we call FusedTriplets. MosaicFinder not only directly groups composite and component gene families but also reduces the risk of outputting a large number of false positives. In such searches, questions of macro-evolution should be addressed with a carefully selected dataset (e.g. introducing sequences from genomes that are representatives from the many taxonomical groups under comparison).

## 2 METHODS

### 2.1 Preliminary notions

A *graph* $G = (V, E)$ is a set of vertices $V$ and a set of edges $E$ that link some pairs of vertices together (our graphs are undirected). Sequence similarity networks are graphs with sequences (or genes) as vertices, connected by edges when they are found to be similar by a pairwise comparison method (Smith and Waterman, 1981), BLAST (Altschul *et al*., 1990) and BLAST-Like Alignment Tool (Kent, 2002). Two vertices are *adjacent* if they are linked by an edge, i.e. two sequences are adjacent if they are similar. The *neighbourhood* of a vertex $x$ is the set $N(x)$ of vertices that are adjacent to $x$ ($x$ not included). Given a subset $X$ of vertices, we will call *common neighbourhood* of $X$, denoted $CN(X)$, the intersection of the neighbourhoods of all the vertices of $X$ [i.e. $CN(X) = \bigcap_{x \in X} N(x)$]. Hence, the common neighbourhood of a set of genes $F$ (e.g. a gene family) is the set of sequences in the dataset that are similar to every sequence of $F$, sequences of $F$ excluded. A *clique* (also called *complete* subgraph) is a set of pairwise adjacent vertices. A set of genes $F$ is a clique if for every pair of sequence $(u, v)$ in $F$, $u$ and $v$ are similar. It usually means that sequences in $F$ have a conserved homologous region in common. A graph is *connected* if there is a path between any pair of vertices. A *connected component* is a maximal connected subgraph. Note that two sequences in the same connected component may not have any homologous region in common (Fig. 2). A *separator* is a set of vertices whose removal increases the number of connected components. A *clique separator* is a separator that is a clique. A *clique minimal separator* (which we will shorten to CMS) is a clique separator, which is minimal for the separation of two given vertices (the reader is referred to Berry *et al*., 2010 for graph definitions and details on CMSs).

Typically, a sequence similarity network can be reconstructed for a large dataset by connecting genes that are related in a BLAST (Altschul *et al*., 1990) search, with an *E-value* score better than a user-defined threshold. Sequence similarity networks are graphs with sequences (or genes) as vertices, directly connected by edges when they show a similarity greater than a user-defined threshold. For a given comparison between two sequences, the alignment, score and E-value are not symmetric. They can vary depending on which sequence is used as the query. The network is symmetrized by considering the best match of each pairwise comparison. As the greatest asymmetry is found in the better-scoring comparisons [i.e. at a much more stringent threshold than the ones used for network reconstruction (Atkinson *et al*., 2009)], this procedure does not impact the topology. Thus, the structure of this network captures much of the history of gene evolution: not only classical divergence by point mutations but also recombinations, fusions and fission events (Adai *et al*., 2004). Conserved families of genes with a single common ancestor are all connected to each other in a connected component of the graph (unless they evolved beyond recognition by BLAST). They form cliques of sequences in the network, which are aligned over most of their length. Divergent families will form less densely connected groups of vertices because the common ancestry between some pairs of their genes is less frequently detected.

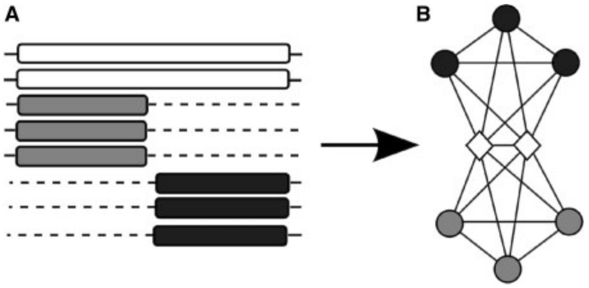

**Fig. 2.** (**A**) Multiple alignment of composite genes (white) and component genes (grey and black). (**B**) Similarity network of those genes. The white vertices form a composite gene family. They are a clique minimal separator of the network. The black vertices and the grey vertices form two separate component families

### 2.2 FusedTriplets: implementation of the gene-centred method

As explained in the introduction and by Figure 1, a composite gene is characterized in the similarity network by connecting two component genes that are not adjacent, and which are similar to disjoint parts of the composite gene. This leads to the following steps to identify fused triplets: enumeration of all non-transitive triplets of genes and cross-check of the absence of similarity between component genes and test of their alignment overlap along the composite gene.

We cross-check the absence of similarity between component genes in the same way as Yanai *et al*. (2001). It simply consists of testing whether they remain dissimilar at a more permissive threshold than the one used for triplet enumeration. For example, if one considered component and composite genes similar if $E - value \leq 1e - 10$, one can test whether component genes have an $E - value \leq 1e - 5$ to make sure that they are dissimilar [see Atkinson *et al*. (2009) for the effect of the threshold on similarity network topology]. This method presents the great advantage that it requires no further computation, as it only uses the similarity network information.

We reject triplets whose component gene alignment along composite gene overlap by >20 amino acids (as in Yanai *et al*., 2001). This small overlap is allowed because BLAST alignments tend to extend slightly beyond homologous regions.

We implemented these steps in the Python script FusedTriplets.

## 2.3 MosaicFinder

When exploring a large dataset including several genomes, there may be several representatives of a given fusion event, which we would want to group into composite and component gene families. Rather than doing this grouping *a posteriori* from the results of a gene-centred approach, an interesting prospect is to identify those families directly in the similarity network (Fig. 2).

A family of composite genes has the particularity to link otherwise non-connected groups of nodes. The characteristic non-transitive pattern of composite genes extends to families. We propose to characterize a composite gene family as a CMS of the sequence similarity network (Berry *et al*., 2010). A composite gene family is a *separator*, as its removal disconnects component gene families. It is *minimal*, as every composite gene is similar to the components. The additional condition that the separator is a *clique* describes the requirement that the family of composite gene is conserved. It should be noted that a composite gene that is the only representative of its family will be identified, as a clique minimal separator of size one.

For all these reasons, CMS is a good model to identify composite gene families. In addition, CMSs present several interesting properties: the number of CMSs bounded by the number of vertices, and an exact polynomial-time algorithm exists to identify them. MosaicFinder works in several consecutive steps, which are detailed later in the text.

**STEP 1: Construction of the similarity network**

MosaicFinder takes the result as input of all-against-all BLAST comparisons between the sequences under study, in the form of a simple flat-table, including information about the region that aligns between pairs of sequences (BLAST qstart, qend, sstart, send). To determine whether two sequences are similar, MosaicFinder relies on a pair of similarity scores, the ‘E-value’ and ‘percentage of identity’ of these two sequences. The results are then represented as an undirected network $G = (V, E)$, where $V$ is the set of sequences, and edge is $(u, v) \in E$ if the similarity score $S_{uv}$ or $S_{vu}$ is higher than a user-defined threshold.

**STEP 2: Identification of fused gene families**

A graph algorithm (Berry *et al*., 2010) is then applied to find clique minimal separators in this network and to propose candidate families of composite genes. This is the central and longest step of MosaicFinder.

**STEP 3: Identification of component families**

The component families are then identified by disconnections in the common neighbourhood of each CMS. This common neighbourhood does not contain the nodes from the separator. It, therefore, contains several connected components, which we defined as component families. Figure 3 illustrates this process.

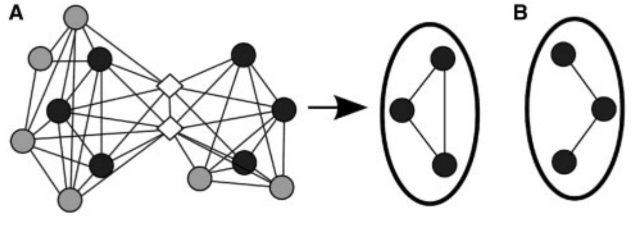

**Fig. 3.** (**A**) White nodes are a clique minimal separator. Black nodes are its common neighbourhood. Note that some grey nodes may be connected to the separator but not be in its *common* neighbourhood. (**B**) The subgraph of the CMS common neighbourhood contains two connected components, which define its component families. Note that component families are not required to be fully connected

**STEP 4: Cross-checking similarity between component families (optional)**

MosaicFinder optionally tests component families for distant homology. A simple way to verify the absence of any similarity between component families is to test whether they remain disconnected with a more permissive threshold of identity than the threshold used to identify the CMS. It should be stressed that the *gap* between the two thresholds, and not the absolute value of the permissive one, ensures that component families are not distant homologs. This test is optional because a disconnection between two component families is already robust. Furthermore, raising the BLAST score can increase the risk of detecting false positives, especially for an E-value beyond 1e-3 (Fokkens *et al*., 2010) and for large datasets.

**STEP 5: Use of alignments to eliminate false positives**

Undetected distant homologues in the common neighbourhood of a CMS may still lead to an overestimation of the number of component families and composite gene families. MosaicFinder further tests for the presence of such distant homologues using information about the regions that align in BLAST comparison between vertices from the CMS and vertices from their common neighbourhood. There is a false positive when the different component families align in the same region of a candidate composite gene because such a significant overlap in an alignment suggests that homology between sequences of different component families was undetected. As different genes from a component family $F$ may align to slightly different parts of a potential composite gene $v$, we used the median alignment of $F$ along $v$, defined as follows. The start position of the median alignment of $F$ along $v$ is the median over all start positions of alignments of $F$ family members along gene $v$, and the end is similarly the median over all end positions. MosaicFinder rejects a candidate composite gene if the median alignment of different component families overlaps on >20 (by default) amino acids. This small overlap is allowed because BLAST extends alignments as far as possible, and small non-homologous flanking regions may artefactually align. Otherwise, the composite gene is accepted, and a ‘fusion point’ is calculated as the middle point between the median alignments of each component families.

**STEP 6: Output**

MosaicFinder outputs a table of genes and gene families involved in fusion events. This table indicates the fusion event that genes are involved in, and their groupings into composite or component families. It additionally indicates a fusion point for composite genes.

# 3 RESULTS

We implemented MosaicFinder and FusedTriplets, which we used to compare the detection of composite gene families with the existing methods for detecting composite genes. As there exists no large manually curated database of composite genes to use as a test bed, we simulated the evolution of composite genes and composite gene families to test the accuracy of MosaicFinder.

We also ran tests on real databases, but we have less information on the validity of our (or other) methods in this context. We focused our attention on the number of composite genes detected.

## 3.1 Test of MosaicFinder on simulated composite gene families

We simulated the evolution of component and composite gene families under various evolutionary circumstances to test and compare the sensitivity and specificity of MosaicFinder and FusedTriplets in their detection of composite genes (Fig. 4).

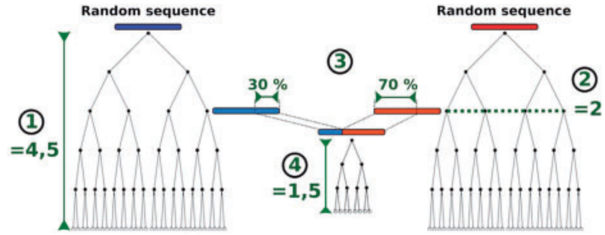

**Fig. 4.** Simulation of the evolution of component and composite gene families. Two random initial sequences are evolved along five-level perfect binary trees to produce two component gene families. Total length of both trees is scaled by the same evolutionary rate (parameter 1). A pair of sequences evolving along these trees is chosen at the same distance from root (parameter 2, *fusion level*). A given percentage (parameter 3) of the first sequence is fused with a fragment of the second sequence to create a new composite sequence of the same length. This sequence is evolved along a perfect binary tree with five fusion levels, scaled by a given evolutionary rate (parameter 4) to produce the composite gene families. In this figure, component families are divergent (tree length $\approx$ 5 MNSS), whereas composite family is conserved (tree length $\leq 2$)

We used Seq-Gen (Rambaut and Grass, 1997) to simulate the evolution of component families, under the Whelan and Goldman model of amino acid substitution and a site-specific rate heterogeneity following a continuous gamma distribution ($\alpha = 1$). Ancestral sequences of 300 amino acids were generated randomly for each component family. These sequences were then evolved along perfect (complete) binary trees with five levels, i.e. symmetric and balanced trees with $2^5 = 32$ leaves at the fifth level, resulting in component families with 32 genes. We explored the effect of gene family divergence on composite gene detection under the hypothesis that the more divergent gene families are, the harder they are to detect. We produced gene families with different degrees of divergence as follows. We scaled these ultrametric phylogenetic trees with Seq-Gen (option -d) so that the total length of a tree can be measured as the distance from the root to any of the leaves in units of mean number of substitutions per site (MNSS). Typically, a tree of length 2 MNSS resulted in conserved families with all pairs of sequences presenting an E-value of $\leq$1e-10 (therefore, corresponding gene families forming cliques in a gene network reconstructed at that threshold). By contrast, trees of length 5 MNSS resulted in divergent families in which homology between many pairs of sequences was no longer detectable by BLAST. In these rapidly evolving trees, 99% of all pairs of maximally distant sequences presented an E-value of $\leq$1e-10 and 90% an E-value of $\leq$1e-5. To cover the range from highly conserved to highly divergent gene families, we explored 14 evolutionary rates, from 0.5 to 7 with a step of 0.5 (parameter 1). We generated simulated fusion events from a pair of component families evolving at the same evolutionary rate. A pair of sequences evolving along these trees was chosen at the same distance from the tree root [*fusion level* from 0 to 5 (parameter 2)]. We used this pair of sequences to create a novel 300 amino acids composite sequence made of 10–50% of the first sequence fused with 90–50% of the second sequence (parameter 3). This ancestral composite sequence was then evolved along a third perfect binary tree with five fusion levels, so that genes from composite and component gene families had undergone the same number of diversification events starting from ancestral component sequences (Fig. 4). The composite family was evolved at the same 14 evolutionary rates (parameter 4) that were used for the component families, thereby producing highly conserved to highly divergent composite families. For recent fusion events (*fusion level* = 0), the composite sequence was left unmodified. This protocol was repeated 10 times for each combination of the four parameters. We, therefore, simulated $10 * 14 * 6 * 5 * 14 = 58.800$ fusion events.

## 3.2 Result on simulation

For each simulated fusion event, we compared all pairs of genes from this dataset with BLASTp (E-value of $\leq$1e-5). We searched the resulting similarity network with MosaicFinder, FusedTriplets and FusedTriplets_E10, i.e. FusedTriplets with a more stringent 1e-10 E-value threshold and a cross-check of the absence of similarity between component genes/families at the original 1e-5 E-value threshold. The main explanatory parameters for composite gene detection results are evolutionary rates of component and composite gene families. We analysed the proportion of edges between composite and component genes that were recovered in the network, for various combinations of evolutionary rates (Supplementary Fig. S2.1). As expected, the majority of connections between fast evolving (>3 MNSS) component and composite families were lost, which defines an 'evolutionary zone' within which both MosaicFinder and FusedTriplets will work best. We compared the three methods with respect to the proportions of detected false positives (component genes that were erroneously identified as composite genes) and the proportions of detected true positives (composite genes that were correctly identified). In our simulations, all methods returned few false positives ($\leq$5%) for all combinations of evolutionary rates. However, FusedTriplets displayed higher proportions of false positives than FusedTriplets_E10 and MosaicFinder, this latter seemed as the method which is the least prone to outputting false positives (Supplementary Fig. S2.2). Such false positives seem to arise for particular combinations of evolutionary rates, leading to triplets in which there is a composite gene, located at one of the extremities of the triplet and two component genes. This topology is obtained when the intermediate component gene (i.e. the false positive) is connected on the one hand to its homologue (the other component gene) because of some sequence similarity that is still detectable for a given region of their sequences, and on the other hand to the composite gene via a different region of its sequence. As gene networks based on real data often connect sequences through partial regions of similarity, the analysis of real data may result in the detection of such false positives. These results strongly suggest that it is generally a good idea to use two thresholds with different stringencies in the detection of candidate composite genes, in analyses with FusedTriplet. Regarding the detection of true positives, there seemed to be 'evolutionary zones' in which all three methods recovered significant proportions of composite genes in our simulations (Supplementary Fig. S2.3.A). However, on closer examination, within these zones, the methods performed differently. First, we compared FusedTriplets with MosaicFinder (Supplementary

Fig. S2.3.B). Logically, MosaicFinder returned less true positives than FusedTriplets because MosaicFinder cannot detect candidate composite genes that are not also proposed by FusedTriplets. However, FusedTriplet_E10 (less sensitive than FusedTriplets to false positives, as described earlier in the text) is less efficient for detecting composite genes than MosaicFinder. Therefore, using MosaicFinder to analyse large datasets seems as a good trend. Overall, MosaicFinder is more robust than FusedTriplet, as it produces almost no false positives and successfully detects composite genes and groups them into families. We also investigated how other parameters (percentage of fused material from the component genes, fusion levels) affected the detection of false positives and true positives by these three methods (Supplementary Fig. S3). We observed that composite genes simulated in more recent events were more frequently detected than composite genes simulated in older events by all methods, and especially by MosaicFinder (Supplementary Fig. S3.3). Likewise, composite genes simulated in more balanced fusion events (e.g. when composite genes received fragments of similar sizes from the component genes) were more frequently detected than composite genes simulated in less balanced fusion events by all methods (Supplementary Fig. S3.5). This was expected because it is harder for any method to detect similarity between composite genes and component genes on shorter fragments, but FusedTriplet_E10 was more affected by this problem than the other methods. Regarding the fusion points, we find that MosaicFinder accordingly estimates the position of the fusion points. In all, 94% of the computed fusion points are <5 amino acids away from the *true* fusion point, and 99% <16 amino acids away. This variation is due to the imprecision of BLASTp alignments. Those numbers validate *a posteriori* the 20 amino acids overlap allowed between component families on composite genes.

### 3.3 Biological results

Our analyses of a real large dataset (591.439 sequences from the three domains of life and from mobile genetic elements, such as viruses and plasmids) with MosaicFinder extended our knowledge on the evolution of composite gene families. First, it showed that all types of genomes, whether they come from cellular organisms or from their mobile genetic elements, are concerned by the process of gene fusion (Supplementary Fig. S4). Eukaryotic genomes are significantly much more affected by this process than prokaryotic genomes and genomes of mobile elements; however, when the focus of the analysis is limited to the evolution of prokaryotes and their mobile genetic elements, these latter, in particular the plasmids, can be showed to be critically involved in that process. An excess of families of composite genes are found on plasmids, suggesting that these important vessels of DNA mobility are involved in the creation and/or the distribution of composite genes. This conclusion is consistent with the literature that claims that genomic evolution cannot be accurately described without taking the role of these infracellular entities into account (Bapteste and Burian, 2010; Bapteste *et al.*, 2012; Halary *et al.*, 2009). Moreover, our implementation allowed us to study the triplets centred on composite genes detected by MosaicFinder (Supplementary Fig. S5), offering an additional way to investigate the respective contribution of cellular entities and mobile genetic elements to the process of gene fusion. When sequences from all genomes are analysed, 53% of these triplets only connect genes from cellular organisms; yet, when the focus is on the genomes of prokaryotes and mobile genetic elements, this proportion logically drops to 42% (because of the removal of intra-eukaryotic fusions). This result is consistent with our previous observations and is remarkable because it means that although a small majority of gene fusions apparently exclusively involves the genetic material from cellular organisms (when eukaryotic genomes are taken into consideration), a very large fraction of gene fusion events detected in that dataset have possibly involved the contribution of at least one mobile genetic element. An analysis of these triplets at a finer scale further suggests that mobile genetic elements were likely providers of some DNA for up to 39% of these fusions (which can be deduced from the sum of the percentages of triplets in which at least one sequence of mobile genetic element is at least at one of the extremities of the triplets), and that mobile genetic elements were possibly carriers of composite genes for up to 20% of these fusions (as can be deduced from the sum of the percentages of the per cent of triplets in which the composite sequence is carried by a mobile genetic element). When only genomes of prokaryotes and mobile genetic elements are considered, mobile genetic elements seem to act as providers of DNA/carriers of composite genes in up to 48/25% of the gene fusions, respectively. The functions of these composite genes are described in Supplementary Figure S6.

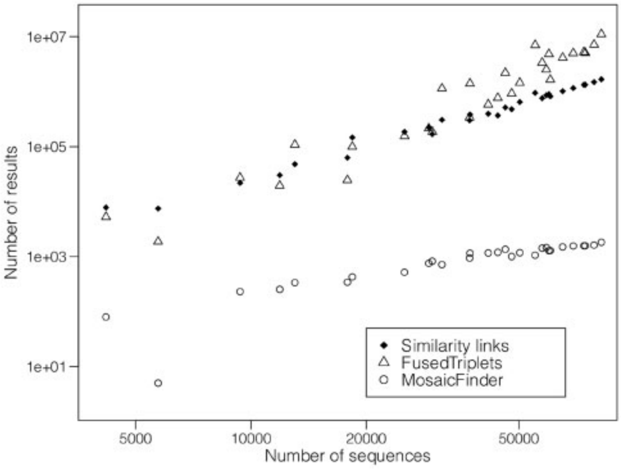

**Fig. 5.** Comparison of the number of edges between sequences, the number of identified composite families by MosaicFinder and the number of identified composite triplets by FusedTriplets (logarithmic scales)

We also built datasets of various sizes, composed of 1–30 prokaryotic complete genomes, to search for composite and component genes. Figure 5 presents the number of composite gene triplets and triplets of families output by FusedTriplets and MosaicFinder, respectively, compared with the number of edges between sequences. FusedTriplets outputs an enormous amount of fused triplets (up to 11 millions), whereas MosaicFinder outputs only up to 1821 fusion events. FusedTriplets outputs up to 5339 potential composite genes for the biggest dataset (note that a given composite gene can correspond to many different triplets), whereas MosaicFinder finds 2490 unique composite genes. Most composite families (1313) identified by MosaicFinder contain only one sequence. Of the remaining 508 fusion events, for 349 composite gene families, we could only detect one representative sequence of one of the two component families involved in the event. Thus, 159 detected fusion events (amounting to 985 composite genes) involved composite and component gene families with more than one sequence. These numbers show the great number of potentially misleading fusion events, and the interest of MosaicFinder is to identify them.

## 4 DISCUSSION

We proposed a new characterization of families of composite genes, as clique minimal separators in sequence similarity networks, and implemented this method into the C++ program MosaicFinder. We showed that on simulated data, MosaicFinder identifies conserved composite gene families well. Even if MosaicFinder was not designed to do so, it also identifies the evolutionary conserved fraction of composite genes from divergent families. In cases where divergent genes have evolved too much to show similarity to both component families, MosaicFinder proves to have a very low false positive rate.

We show that MosaicFinder gives good results quickly, with the advantage that genes are grouped into families, thus avoiding the extra work of regrouping the composite genes after they are output. Moreover, this information may be visualized as an annotated graph using Cytoscape (Shannon *et al*., 2003). Supplementary Figure S7 gives an example.

According to our results from Section 3, MosaicFinder generates few false positives. Since in the real dataset constructed with 30 complete prokaryote genomes, MosaicFinder detected the impressive rate of one fusion gene of 33 genes, we can conjecture that in real data, there are in fact many composite genes.

Future work consists in breaking up long cycles with a local approach, as long cycles may mask fusion families by connecting component families indirectly (Supplementary Fig. S8).

## 5 SOFTWARE

MosaicFinder is based on the graph theoretic tool of clique separator decomposition. MosaicFinder is reliable for studying fusion events for phylogenetic research as well as for functional biology. The program has been developed in C++. FusedTriplets is a Python script that generalizes previous approaches to find composite genes, based on sequence similarity network abstraction. Both programs are freely available with their source code at this address http://sourceforge.net/projects/mosaicfinder/.

## ACKNOWLEDGEMENT

The authors thank Michel Habib for suggesting using clique minimal separators.

*Funding*: P.A.J. was funded by an AMX PhD grant, R.P. and A.B. were funded by the French Agency for Research (DEFIS program TODO) and the ANR-09-EMER-010.

*Conflict of Interest*: none declared.

## REFERENCES

Abdelnoor,R.V. *et al*. (2006) Mitochondrial genome dynamics in plants and animals: convergent gene fusions of a MutS homologue. *J. Mol. Evol*., **63**, 165–173.

Adai,A. *et al*. (2004) LGL: creating a map of protein function with an algorithm for visualizing very large biological networks. *J. Mol. Biol*., **340**, 179–190.

Aleshin,V.V. *et al*. (2007) Do we need many genes for phylogenetic inference? *Biochemistry (Mosc.)*, **72**, 1313–1323.

Altschul,S.F. *et al*. (1990) Basic local alignment search tool. *J. Mol. Biol*., **215**, 403–410.

Atkinson,H.J. *et al*. (2009) Using sequence similarity networks for visualization of relationships across diverse protein superfamilies. *PLoS ONE*, **4**, e4345.

Bapteste,E. and Burian,R.M. (2010) On the need for integrative phylogenomics, and some steps toward its creation. *Biol. Philos*., **25**, 711–736.

Bapteste,E. *et al*. (2012) Evolutionary analyses of non-genealogical bonds produced by introgressive descent. *Proc. Natl Acad. Sci. USA*, **109**, 18266–18272.

Berry,A. *et al*. (2010) An introduction to clique minimal separator decomposition. *Algorithms*, **3**, 197–215.

Buljan,M. *et al*. (2010) Quantifying the mechanisms of domain gain in animal proteins. *Genome Biol*., **11**, R74.

Durrens,P. *et al*. (2008) Fusion and fission of genes define a metric between fungal genomes. *PLoS Comput. Biol*., **4**, e1000200.

Ekman,D. *et al*. (2007) Quantification of the elevated rate of domain rearrangements in metazoa. *J. Mol. Biol*., **372**, 1337–1348.

Enright,A. and Ouzounis,C. (2001) Functional associations of proteins in entire genomes by means of exhaustive detection of gene fusions. *Genome Biol*., **2**, 10034.

Enright,A.J. and Ouzounis,C.A. (2000) GeneRAGE: a robust algorithm for sequence clustering and domain detection. *Bioinformatics*, **16**, 451–457.

Enright,A.J. *et al*. (1999) Protein interaction maps for complete genomes based on gene fusion events. *Nature*, **402**, 86–90.

Enright,A.J. *et al*. (2002) An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res*., **30**, 1575–1584.

Fani,R. *et al*. (2007) The role of gene fusions in the evolution of metabolic pathways: the histidine biosynthesis case. *BMC Evol. Biol*, **7** (**Suppl. 2**), S4.

Fokkens,L. *et al*. (2010) Enrichment of homologs in insignificant BLAST hits by co-complex network alignment. *BMC Bioinformatics*, **11**, 86.

Green,R. *et al*. (2011) Independent evolutionary origins of functional polyamine biosynthetic enzyme fusions catalysing de novo diamine to triamine formation. *Mol. Microbiol*., **81**, 1109–1124.

Halary,S. *et al*. (2009) Network analyses structure genetic diversity in independent genetic worlds. *Proc. Natl Acad. Sci. USA*, **107**, 127–132.

Han,J.-H. *et al*. (2007) The folding and evolution of multidomain proteins. *Nat. Rev. Mol. Cell Biol*., **8**, 319–330.

Kent,W.J. (2002) BLATThe BLAST-like alignment tool. *Genome Res*., **12**, 656–664.

Lawson,A.R.J. *et al*. (2011) RAF gene fusion breakpoints in pediatric brain tumors are characterized by significant enrichment of sequence microhomology. *Genome Res*., **21**, 505–514.

Long,M. (2000) A new function evolved from gene fusion. *Genome Res*., **10**, 1655–1657.

Makiuchi,T. *et al*. (2007) Occurrence of multiple, independent gene fusion events for the fifth and sixth enzymes of pyrimidine biosynthesis in different eukaryotic groups. *Gene*, **394**, 78–86.

Marcotte,E.M. *et al*. (1999) Detecting protein function and protein-protein interactions from genome sequences. *Science*, **285**, 751–753.

Marsh,J.A. and Teichmann,S.A. (2010) How do proteins gain new domains? *Genome Biol*, **11**, 126.

Moran,Y. *et al*. (2009) Fusion and retrotransposition events in the evolution of the sea anemone *Anemonia viridis* neurotoxin genes. *J. Mol. Evol*., **69**, 115–124.

Nakamura,Y. *et al*. (2007) Rate and polarity of gene fusion and fission in *Oryza sativa* and *Arabidopsis thaliana*. *Mol. Biol. Evol*., **24**, 110–121.

Narsing,S. *et al*. (2009) Genes that contribute to cancer fusion genes are large and evolutionarily conserved. *Cancer Genet. Cytogenet*, **191**, 78–84.

Nie,Y. *et al*. (2011) Two novel alkane hydroxylase-rubredoxin fusion genes isolated from a dietzia bacterium and the functions of fused rubredoxin domains in long-chain n-alkane degradation. *Appl. Environ. Microbiol*., **77**, 7279–7288.

Opazo,J.C. *et al*. (2009) Origin and ascendancy of a chimeric fusion gene: the beta/delta-globin gene of paenungulate mammals. *Mol. Biol. Evol*., **26**, 1469–1478.

Park,J. *et al.* (1997) Intermediate sequences increase the detection of homology between sequences. *J. Mol. Biol.*, **273**, 349–354.

Pasek,S. *et al.* (2006) Gene fusion/fission is a major contributor to evolution of multi-domain bacterial proteins. *Bioinformatics*, **22**, 1418–1423.

Patthy,L. (2003) Modular assembly of genes and the evolution of new functions. *Genetica*, **118**, 217–231.

Rambaut,A. and Grass,N.C. (1997) Seq-gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.*, **13**, 235–238.

Rivals,I. *et al.* (2007) Enrichment or depletion of a GO category within a class of genes: which test? *Bioinformatics*, **23**, 401–407.

Rodrigues,M.V. *et al.* (2007) Bifunctional CTP: inositol-1-phosphate cytidylyltransferase/CDP-inositol: inositol-1-phosphate transferase, the key enzyme for di-myo-inositol-phosphate synthesis in several (hyper) thermophiles. *J. Bacteriol.*, **189**, 5405–5412.

Rogers,R.L. and Hartl,D.L. (2012) Chimeric genes as a source of rapid evolution in *Drosophila melanogaster*. *Mol. Biol. Evol.*, **29**, 517–529.

Rogers,R.L. *et al.* (2009) Formation and longevity of chimeric and duplicate genes in *Drosophila melanogaster*. *Genetics*, **181**, 313–322.

Salim,H.M.W. *et al.* (2011) deFuser/detection of fused genes in eukaryotic genomes using gene deFuser: analysis of the *Tetrahymena thermophila* genome. *BMC Bioinformatics*, **12**, 279.

Shannon,P. *et al.* (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.*, **13**, 2498–2504.

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

Snel,B.B. *et al.* (2000) Genome evolution-gene fusion versus gene fission. *Trends Genet.*, **16**, 9–11.

Song,N. *et al.* (2008) Sequence similarity network reveals common ancestry of multidomain proteins. *PLoS Comput. Biol.*, **4**, e1000063.

Stechmann,A. and Cavalier-Smith,T. (2002) Rooting the eukaryote tree by using a derived gene fusion. *Science*, **297**, 89–91.

Suhre,K. (2004) FusionDB: a database for in-depth analysis of prokaryotic gene fusion events. *Nucleic Acids Res.*, **32**, 273D–276D.

Tatusov,R.L. *et al.* (1997) A genomic perspective on protein families. *Science*, **278**, 631–637.

Tatusov,R.L. *et al.* (2003) The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*, **4**, 41.

Tordai,H. *et al.* (2005) Modules, multidomain proteins and organismic complexity. *FEBS J.*, **272**, 5064–5078.

Waller,R.F. *et al.* (2006) Lateral gene transfer of a multigene region from cyanobacteria to dinoflagellates resulting in a novel plastid-targeted fusion protein. *Mol. Biol. Evol.*, **23**, 1437–1443.

Wu,Y.-C. *et al.* (2011) Evolution at the subgene level: domain rearrangements in the *Drosophila* phylogeny. *Mol. Biol. Evol.*, **29**, 689–705.

Yanai,I. *et al.* (2001) Genes linked by fusion events are generally of the same functional category: a systematic analysis of 30 microbial genomes. *Proc. Natl Acad. Sci. USA*, **98**, 7940–7945.

Zamocky,M. *et al.* (2004) Ancestral gene fusion in cellobiose dehydrogenases reflects a specific evolution of GMC oxidoreductases in fungi. *Gene*, **338**, 1–14.

Zhao,X. *et al.* (2011) ALS51, a newly discovered gene in the Candida albicans ALS family, created by intergenic recombination: analysis of the gene and protein, and implications for evolution of microbial gene families. *FEMS Immunol. Med. Microbiol*, **61**, 245–257.

Zhou,Q. *et al.* (2008) On the origin of new genes in *Drosophila. Genome Res.*, **18**, 1446–1455.