# CompositeSearch: A Generalized Network Approach for Composite Gene Families Detection

Jananan Sylvestre Pathmanathan,[1] Philippe Lopez,[1] François-Joseph Lapointe,[2] and Eric Bapteste*,[1]

[1]Institut de Biologie Paris-Seine (IBPS), UPMC Université Paris 06, Sorbonne Universités, Paris, France

[2]Département de Sciences Biologiques, Université de Montréal, Montréal, QC, Canada

***Corresponding author:** E-mail: eric.bapteste@upmc.fr.

**Associate editor:** Tal Pupko

## Abstract

**Genes evolve by point mutations, but also by shuffling, fusion, and fission of genetic fragments. Therefore, similarity between two sequences can be due to common ancestry producing homology, and/or partial sharing of component fragments. Disentangling these processes is especially challenging in large molecular data sets, because of computational time. In this article, we present CompositeSearch, a memory-efficient, fast, and scalable method to detect composite gene families in large data sets (typically in the range of several million sequences). CompositeSearch generalizes the use of similarity networks to detect composite and component gene families with a greater recall, accuracy, and precision than recent programs (FusedTriplets and MosaicFinder). Moreover, CompositeSearch provides user-friendly quality descriptions regarding the distribution and primary sequence conservation of these gene families allowing critical biological analyses of these data.**

***Key words:*** **bioinformatics, evolution, molecular evolution, network analysis, protein sequence analysis.**

Genetic sequences evolve through multiple processes beyond point mutations. In particular, the remodeling of genes by shuffling of genetic fragments, fusion, and fission, as well as de novo gene emergence, contributes to the creation, and diversification of gene families (Kawai et al. 2003; Moore et al. 2008; Kaessmann 2010; Marsh and Teichmann 2010; Wu et al. 2012; Promponas et al. 2014; Bornberg-Bauer et al. 2015; McLysaght and Guerzoni 2015; Ruiz-Orera et al. 2015; Guerzoni and McLysaght 2016; Lees et al. 2016; Meheust et al. 2016). Therefore, genetic sequences show similarity with one another for diverse reasons, that is, common ancestry producing homology, and/or partial sharing of component fragments (Song et al. 2008; Haggerty et al. 2014). These processes must be disentangled to understand the rules and constraints on genes evolution. Although gene remodeling has been especially studied in eukaryotes (Kawai et al. 2003; Patthy 2003; Ekman et al. 2007; Nakamura et al. 2007; Meheust et al. 2016) and in cultured prokaryotes (Enright et al. 1999; Marcotte et al. 1999; Enright and Ouzounis 2000, 2001; Snel et al. 2000; Jachiet et al. 2013), analyses of large molecular data sets remain a computational bottleneck (Salim et al. 2011; Jachiet et al. 2013). For instance, a large scale investigation of how remodeled genes evolved in prokaryotes would require comparing millions of coding sequences from the thousands of complete genomes available, but previous detection methods are unable to handle such large data sets.

In this article, we present CompositeSearch, a memory-efficient, fast, and scalable method to detect composite gene families in large data sets, typically in the range of several million sequences. Composite genes are the result of the fusion of partial or complete nonhomologous DNA fragments, called components, or as a result of fission from a larger gene into dissociated persistent fragment (fig. 1A). CompositeSearch generalizes the use of similarity networks to detect composite and component gene families with a greater recall, accuracy, and precision than recent programs, FusedTriplets and MosaicFinder (Jachiet et al. 2013). Moreover, it provides user-friendly quality descriptions regarding the distribution and primary sequence conservation of these gene families allowing critical biological analyses of these data, and it is used as an input for the reconstruction of multirooted gene networks (Haggerty et al. 2014).

## New Approach

Here, we present CompositeSearch, a memory-efficient, fast, and scalable method, implemented in C++, which detects composite gene families in large data sets (typically in the range of several million sequences). Composite genes are traditionally defined based on their apparent modularity: they are composed of segments (i.e., components) that have evolved separately in distinct gene families (Patthy 2003; Song et al. 2008; Jachiet et al. 2013). Under this definition, composite genes can be the result of fusion of components, or involved as progenitors in fission events, after which associations of components are split in separate gene families. CompositeSearch generalizes the use of sequence similarity networks (SSN) to detect composite and component gene families. SSN are undirected graphs, where each node represents a unique sequence and each edge represents the similarity between connected sequences (given similarity criteria, such as a minimum percentage identity, BLAST *E* value; Altschul et al. 1990 and minimum mutual coverage, i.e., the

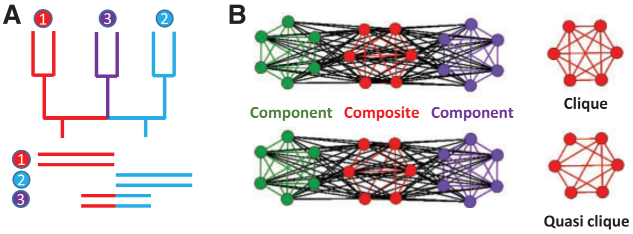

FIG. 1. (*A*) Top: Example of a composite gene. Gene family 3 evolved from a composite of families 1 and 2. Bottom: Sequences from family 3 partially align with sequences from families 1 and 2. (*B*) Similarity network of a composite gene family (red) and its component gene families (green and purple). MosaicFinder will detect only the top case where composite genes form a clique, whereas CompositeSearch detects composite gene families forming a clique (top) or quasi-clique (bottom).

minimal length covered by the matching parts with respect to the total length of each compared sequence) (Jachiet et al. 2013; Corel et al. 2016). 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. Thus, the network is first 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.

This network's structure captures much of the history of gene evolution: not only divergence by point mutations but also recombinations, fusions, and fission events (Adai et al. 2004; Jachiet et al. 2013). Typically, gene families form subgraphs with high connectivity, in which connected sequences display significant BLAST *E* values $\leq 1E^{-5}$, mutual covers $\geq 80\%$, and %ID $\geq 30\%$. By contrast, superfamilies (Atkinson et al. 2009) and composite gene families (Song et al. 2008; Jachiet et al. 2013, 2014; Haggerty et al. 2014; Meheust et al. 2016) introduce more complex informative patterns in SSNs.

Using these graphs to identify composite genes and gene families, CompositeSearch shows a greater recall, accuracy, and precision than recent programs FusedTriplets (FT) and MosaicFinder (MF). In short, these two programs are helpful but limited in scope. FT cannot handle large data sets and does not define composite gene families. MF is also unable to analyze large data sets (due to memory and speed limitations). Although it identifies composite and component gene families, MF is only meant to find highly conserved composite gene families that form minimal clique separators in sequence similarity network. The "clique" condition implies that MF misses divergent (e.g., ancient or fast evolving) composite gene families (whose members do not necessarily connect all together in sequence similarity networks) (fig. 1*B*). The "separator" condition implies that composite genes will remain undetected for data sets with highly remodeled genes by MF. Indeed, the repeated use of gene components introduces cyclic paths in sequence similarity networks, which turns composite families into local, but not global separators.

Beyond its larger scope and better performance, CompositeSearch can also provide quality descriptions (absent from MF and FT) regarding the size and primary sequence conservation of composite and component gene families, easing critical biological analyses of these data. CompositeSearch is available at https://github.com/TeamAIRE/CompositeSearch, last accessed November 2, 2017. For a detailed description of the algorithm, see supplementary Materials and Methods, Supplementary Material online.

# Results

## Benchmark on Simulated Data

We tested and compared CompositeSearch with FT and MF (Jachiet et al. 2013) on 100 replicates of simulated data, covering a large range of parameters and simulating 2-components and 3-components composites (supplementary fig. S4 and Materials and Methods, Supplementary Material online). We explored the effect of gene family divergence and multiple component reassortments on composite gene detection under the hypothesis that the more divergent gene families are, the harder they are to detect. The sensitivity and specificity of each program were summarized in supplementary table S1, Supplementary Material online. In terms of detection of composite genes, CompositeSearch performs as well as FT, with identical True Positive Rate (TPR) and False Positive Rate (FPR), but, unlike FT, CompositeSearch returns composite gene families. However, CompositeSearch has higher TPR than MF, especially for divergent composite sequences, with a similar 1% FPR. Therefore, CompositeSearch will find additional composite genes with respect to MF, thanks to the detection of composite genes forming quasi-cliques. As CompositeSearch is able to detect the number of components for each composite, we created a more detailed table (supplementary table S2, Supplementary Material online) showing the sensitivity and specificity of CompositeSearch to detect the exact number of components.

## Benchmark on Real Data

We also used a data set of 204,894 viral proteins from (Jachiet et al. 2014) to benchmark our software against real data. CompositeSearch detected 21,623 composite genes clustered in 5,532 families, vastly outperforming MF (5,845 composites in 1,718 families). FT found slightly more composites (23,305), but did not return any families. This slight increase in the number of composites detected by FT was mainly due to BLAST overextending matches on real data, thus producing false positives.

## Performances

Because its algorithm uses a dichotomous search to browse the network and because it is multithreaded, CompositeSearch outperforms both FT and MF in terms of speed and memory use, when these parameters are contrasted on a Linux machine with Intel Xeon CPU E5-2630 v2 2.60-GHz processors and 256 GB RAM, even on one CPU. This is especially noticeable for large metagenomic data sets (table 1). By contrast, construction the SSN

**Table 1.** CompositeSearch, FusedTriplets, and MosaicFinder Performances Comparison.

| Data | Nodes | Edges | Software | #CPU | Runtime | Memory (GB) |
|---|---|---|---|---|---|---|
| 1 | 338,868 | 71,946,457 | MosaicFinder | 1 | 548 h 27 min | 82 |
| | | | FusedTriplets | 1 | 70 h 47 min | 18 |
| | | | CompositeSearch | 1 | 00 h 12 min | 2.5 |
| | | | CompositeSearch | 10 | 00 h 06 min | 2.5 |
| 2 | 3,166,706 | 282,789,792 | MosaicFinder | 1 | — | — |
| | | | FusedTriplets | 1 | — | — |
| | | | CompositeSearch | 10 | 08 h 48 min | 32 |

NOTE.—We compared the performance of CompositeSearch, FusedTriplets, and MosaicFinder on the same Linux machine with Intel Xeon CPU E5-2630 v2 2.60-GHz processors and 256 GB RAM. The data (1) are an SSN from plasmids complete genomes (NCBI December 2014) and (2) HCH metagenomes (Sangwan et al. 2012). CompositeSearch outperform FusedTriplets and MosaicFinder even with one CPU as shown for data (1). On the data (2), FusedTriplets and MosaicFinder stop by running out of memory, which was not the case for CompositeSearch.

composite genes and composite gene families detection runs in a few second to few minutes depending on the network's size.

## Discussion

CompositeSearch is an efficient tool that detects composite genes and composite gene families. It allows investigating the process of gene remodeling in large data sets, for example metagenomes and/or thousands of complete genomes. Although CompositeSearch is faster than currently available software, like FusedTriplets and MosaicFinder, it still can be improved. We observed that in CompositeSearch, the most time consuming step is the detection of gene families, using a DFS algorithm than runs on a single CPU. Parallelized algorithms that detect connected components are available (Kang et al. 2009; Iverson et al. 2015), but they usually require high computational resources. As CompositeSearch was developed with maximum portability in mind, these algorithms are not implemented yet could be in a future version.

## Supplementary Material

Supplementary data are available at *Molecular Biology and Evolution* online.

## Acknowledgments

We thank Raphaël Méheust for his helpful and constructive discussion during the development of CompositeSearch. We also thank our beta testers James O. McInerney, Mary J. O'Connell, Raymond Moran, and Rob Leigh. J.S.P. and E.B. are funded by the European Research Council (FP7/2017-2013 Grant Agreement #615274).

## References

Adai AT, Date SV, Wieland S, Marcotte EM. 2004. LGL: creating a map of protein function with an algorithm for visualizing very large biological networks. *J Mol Biol.* 340(1):179–190.

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. 1990. Basic local alignment search tool. *J Mol Biol.* 215(3):403–410.

Atkinson HJ, Morris JH, Ferrin TE, Babbitt PC. 2009. Using sequence similarity networks for visualization of relationships across diverse protein superfamilies. *PLoS One* 4(2):e4345.

Bornberg-Bauer E, Schmitz J, Heberlein M. 2015. Emergence of de novo proteins from 'dark genomic matter' by 'grow slow and moult'. *Biochem Soc Trans.* 43(5):867–873.

Corel E, Lopez P, Meheust R, Bapteste E. 2016. Network-Thinking: graphs to analyze microbial complexity and evolution. *Trends Microbiol.* 24(3):224–237.

Ekman D, Bjorklund AK, Elofsson A. 2007. Quantification of the elevated rate of domain rearrangements in metazoa. *J Mol Biol.* 372(5):1337–1348.

Enright AJ, Iliopoulos I, Kyrpides NC, Ouzounis CA. 1999. Protein interaction maps for complete genomes based on gene fusion events. *Nature* 402(6757):86–90.

Enright AJ, Ouzounis CA. 2000. GeneRAGE: a robust algorithm for sequence clustering and domain detection. *Bioinformatics* 16(5):451–457.

Enright AJ, Ouzounis CA. 2001. Functional associations of proteins in entire genomes by means of exhaustive detection of gene fusions. *Genome Biol.* 2(9):RESEARCH0034.

Guerzoni D, McLysaght A. 2016. De novo genes arise at a slow but steady rate along the primate lineage and have been subject to incomplete lineage sorting. *Genome Biol Evol.* 8(4):1222–1232.

Haggerty LS, Jachiet PA, Hanage WP, Fitzpatrick DA, Lopez P, O'Connell MJ, Pisani D, Wilkinson M, Bapteste E, McInerney JO. 2014. A pluralistic account of homology: adapting the models to the data. *Mol Biol Evol.* 31(3):501–516.

Iverson J, Kamath C, Karypis G. 2015. Evaluation of connected-component labeling algorithms for distributed-memory systems. *Parallel Comput.* 44:53–68.

Jachiet PA, Colson P, Lopez P, Bapteste E. 2014. Extensive gene remodeling in the viral world: new evidence for nongradual evolution in the mobilome network. *Genome Biol Evol.* 6(9):2195–2205.

Jachiet PA, Pogorelcnik R, Berry A, Lopez P, Bapteste E. 2013. MosaicFinder: identification of fused gene families in sequence similarity networks. *Bioinformatics* 29(7):837–844.

Kaessmann H. 2010. Origins, evolution, and phenotypic impact of new genes. *Genome Res.* 20(10):1313–1326.

Kang U, Tsourakakis CE, Faloutsos C. 2009. PEGASUS: A Peta-Scale Graph Mining System – implementation and observations. 2009 9th IEEE International Conference on Data Mining. p. 229–238. IEEE.

Kawai H, Kanegae T, Christensen S, Kiyosue T, Sato Y, Imaizumi T, Kadota A, Wada M. 2003. Responses of ferns to red light are mediated by an unconventional photoreceptor. *Nature* 421(6920):287–290.

Lees JG, Dawson NL, Sillitoe I, Orengo CA. 2016. Functional innovation from changes in protein domains and their combinations. *Curr Opin Struct Biol.* 38:44–52.

Marcotte EM, Pellegrini M, Ng HL, Rice DW, Yeates TO, Eisenberg D. 1999. Detecting protein function and protein-protein interactions from genome sequences. *Science* 285(5428):751–753.

Marsh JA, Teichmann SA. 2010. How do proteins gain new domains? *Genome Biol.* 11(7):126.

McLysaght A, Guerzoni D. 2015. New genes from non-coding sequence: the role of de novo protein-coding genes in eukaryotic evolutionary innovation. *Philos Trans R Soc Lond B Biol Sci.* 370(1678):20140332.

Meheust R, Zelzion E, Bhattacharya D, Lopez P, Bapteste E. 2016. Protein networks identify novel symbiogenetic genes resulting from plastid endosymbiosis. *Proc Natl Acad Sci U S A.* 113(13):3579–3584.

Moore AD, Bjorklund AK, Ekman D, Bornberg-Bauer E, Elofsson A. 2008. Arrangements in the modular evolution of proteins. *Trends Biochem Sci.* 33(9):444–451.

Nakamura Y, Itoh T, Martin W. 2007. Rate and polarity of gene fusion and fission in *Oryza sativa* and *Arabidopsis thaliana*. *Mol Biol Evol.* 24(1):110–121.

Patthy L. 2003. Modular assembly of genes and the evolution of new functions. *Genetica* 118(2–3):217–231.

Promponas VJ, Ouzounis CA, Iliopoulos I. 2014. Experimental evidence validating the computational inference of functional associations from gene fusion events: a critical survey. *Brief Bioinform.* 15(3):443–454.

Ruiz-Orera J, Hernandez-Rodriguez J, Chiva C, Sabido E, Kondova I, Bontrop R, Marques-Bonet T, Alba MM. 2015. Origins of de novo genes in human and chimpanzee. *PLoS Genet.* 11(12):e1005721.

Salim HM, Koire AM, Stover NA, Cavalcanti AR. 2011. Detection of fused genes in eukaryotic genomes using gene deFuser: analysis of the *Tetrahymena thermophila* genome. *BMC Bioinformatics* 12:279.

Sangwan N, Lata P, Dwivedi V, Singh A, Niharika N, Kaur J, Anand S, Malhotra J, Jindal S, Nigam A, et al. 2012. Comparative metagenomic analysis of soil microbial communities across three hexachlorocyclohexane contamination levels. *PLoS One* 7(9):e46219.

Snel B, Bork P, Huynen M. 2000. Genome evolution. Gene fusion versus gene fission. *Trends Genet.* 16(1):9–11.

Song N, Joseph JM, Davis GB, Durand D. 2008. Sequence similarity network reveals common ancestry of multidomain proteins. *PLoS Comput Biol.* 4(4):e1000063.

Wu YC, Rasmussen MD, Kellis M. 2012. Evolution at the subgene level: domain rearrangements in the Drosophila phylogeny. *Mol Biol Evol.* 29(2):689–705.