A Parallel, Distributed-memory Framework for Comparative Motif Discovery

Dieter De Witte[1], Michiel Van Bel[2,3], Pieter Audenaert[1], Piet Demeester[1], Bart Dhoedt[1], Klaas Vandepoele[2,3], and Jan Fostier[1]

[1] Department of Information Technology (INTEC), Ghent University - iMinds, Gaston Crommenlaan 8 bus 201, Ghent, Belgium
dieter.dewitte,pieter.audenaert,piet.demeester, bart.dhoedt,jan.fostier@intec.ugent.be
[2] Department of Plant Systems Biology, VIB, Technologiepark 927, Ghent, Belgium
[3] Department of Plant Biotechnology and Bioinformatics, Ghent University, Technologiepark 927, Ghent, Belgium
michiel.vanbel,klaas.vandepoele@psb.vib-ugent.be

Abstract. The increasing number of sequenced organisms has opened new possibilities for the computational discovery of cis-regulatory elements ('motifs') based on phylogenetic footprinting. Word-based, exhaustive approaches are among the best performing algorithms, however, they pose significant computational challenges as the number of candidate motifs to evaluate is very high. In this contribution, we describe a parallel, distributed-memory framework for *de novo* comparative motif discovery. Within this framework, two approaches for phylogenetic footprinting are implemented: an alignment-based and an alignment-free method. The framework is able to statistically evaluate the conservation of motifs in a search space containing over 160 million candidate motifs using a distributed-memory cluster with 200 CPU cores in a few hours. Software available from http://bioinformatics.intec.ugent.be/blsspeller/

Keywords: Motif discovery, phylogenetic footprinting, parallel computing, distributed-memory

1 Introduction

Over the past decade, numerous computational methods have been proposed for the discovery of cis-regulatory elements (so-called 'motifs') in genomic sequences [1]. The most simple approaches are based on the notion of overrepresentation, i.e., the observation that a specific word occurs more often in a DNA sequence than would be expected by chance. However, as motifs are typically short and degenerate, it is difficult to discriminate between true, functional motifs and background noise.

As more and more organisms are being sequenced, methods based on phylogenetic footprinting are becoming increasingly attractive. The underlying idea is that functional regions in the DNA are subjected to selective pressure, conserving

them over long evolutionary distances, in contrast to non-functional DNA which can diverge freely [2]. Such methods can therefore detect cis-regulatory elements with increased sensitivity through the identification of conserved sequences.

We developed a word-based methodology for comparative motif discovery (unpublished). Whereas most other methods restrict themselves to a (promising) subset of candidate motifs [3, 4], our method is exhaustive, i.e., all words (up to a prespecified length and expressed in a degenerate alphabet) that appear in the input sequences are scored for conservation. Additionally, whereas most existing tools rely on pre-generated multiple sequence alignments [5–8] (MSA) to select conserved motif instances, our method has the ability to run in both alignment-based and alignment-free mode. The alignment-free mode has the advantage that it can detect conserved motif instances, even for distantly related species for which the generation of meaningful multiple sequence alignments is difficult.

In this contribution, we discuss the parallel framework for comparative motif discovery in which these methods were implemented. The main motivation for the development of such framework, is to make effective use of a large, distributed-memory cluster in order to deal with the increased computational requirements inherent to word-based phylogenetic footprinting. More specifically, the computational requirements are high because:

1. Our method exhaustively scores *all* words (up to a prespecified length) that are present in the input sequences. The advantage of this exhaustive approach is that optimal and complete results are produced (as opposed to statistical methods which yield a limited number of local optima, e.g. MEME [9]).
2. We allow candidate motifs to be expressed in a degenerate alphabet, leading to a combinatorial increase of the number of words to evaluate. Currently, we allow words to be expressed in a five-letter alphabet (A, C, G, T and N), however, more sensitive results could be obtained by allowing for two-fold degenerate characters or even the full IUPAC alphabet, again at the cost of increased computational requirements.
3. In order to avoid the recomputation of previously generated intermediate results, we keep all words and their conservation information in memory. Because of the large number of words, the memory requirements can exceed what can be provided by a typical workstation, advocating the development of a distributed-memory implementation.
4. In our current studies, we use four related organisms. The current trend is to incorporate more and more organisms, as this may again improve the sensitivity of the method.
5. In order to assess whether a word is significantly conserved, we employ the genome-wide statistical evaluation that was introduced in [10]. This gives rise to data dependencies between processes and hence, significant inter-process communication. By carefully choosing how the data is partitioned across the local memories of the different machines, communication volumes can be minimized.

Additional motivation can be found by looking at the evolution of computer hardware: since the introduction of the first multi-core CPU around 2003, computational power of a CPU chip has mainly progressed by incorporating more and more CPU cores. Additionally, powerful clusters are created by assembling a large number of workstations and connecting them with an intercommunication network. In order to take advantage of such hardware configurations, parallel software methodologies must be developed.

This paper is organized as follows: in Section 2, the framework's workflow is discussed, along with a discussion of how 'conservation' is quantified and how the genome-wide statistical testing is performed. In Section 3, the parallel, distributed-memory implementation is described. Section 4 presents results and current limitations of the proposed framework, followed by a conclusion and future research directions in Section 5.

Our parallel framework is open-source and can be obtained free of charge from http://bioinformatics.intec.ugent.be/blsspeller.

2 Comparative motif discovery framework

Consider a number of S related species for which orthologous genes are grouped into so-called gene families. The promoter sequences upstream of the genes are extracted. The assumption underlying phylogenetic footprinting is that the genes within a family are regulated by the same (set of) transcription factors. In its most simple form, each of the F different gene families contains a single promoter sequence (2 strands) from each organism. The input hence consists of $2SF$ different promoter sequences[1] with a total length of $N = 2SFN_s$ characters, where N_s denotes the length of a single promoter sequence.

The framework consists of two phases: the intra-family and inter-family phase. During the intra-family step, words are scored for conservation *within* each family individually; in the inter-family step, a confidence score is established for each word w by comparing the number of gene families in which w is conserved to a background model. This statistical model was adopted from [10]. Both phases are now described in more detail.

Intra-family phase During this step, each gene family is processed individually. Given a specific gene family, all words with a length between $l_{\min}$ and $l_{\max}$ that occur within that family are exhaustively enumerated and scored for conservation. The outcome for each word w is binary: either it is sufficiently conserved within that family or it is not. The framework imposes no restrictions on what kind of conservation metric is used. The goal of this phase is to count the number of gene families in which each word w is conserved.

In our software, we provide two complementary ways of doing this. In the *alignment-based mode*, we rely on pre-generated MSAs of the promoter sequences in a family. Words are then enumerated by adopting a sliding window approach

[1] Note that we also take paralogous genes into account, hence our dataset is actually slightly larger than described.

and conservation is based on whether a word is aligned in several species within the MSA. Alternatively, in the *alignment-free* mode, a generalized suffix tree (GST) [11] is constructed [12] from the sequences within a family, and the speller algorithm [13, 14] is used to enumerate all words. The degree of conservation of a word w is based only on the presence or absence of w in a promoter sequence, regardless of the (relative) position or orientation of w. The alignment-free mode is especially beneficial for organisms that are more diverged, for which the generation of MSAs is difficult or even impossible.

In both approaches, the degree of conservation of a given word w is quantified in a biologically meaningful way, by means of the branch length score (BLS) [10]. The BLS ranges between 0% (not conserved at all) to 100% (fully conserved in all sequences) and takes the phylogenetic relationships between the organisms into account. If the BLS exceeds a certain threshold T, the word w is assumed to be *conserved* in that gene family.

Every conserved word w is stored in a hash table, along with the number of gene families F_w in which w is conserved. The hash table hence consists of a large number of $< w, F_w >$ key-value pairs. Words that are not conserved are not stored in the table.

Inter-family phase During this step, for each word w that is stored in the hash table, it is established whether or not this word is significantly conserved in the complete dataset, i.e., all gene families. A confidence score C is established by comparing the number of gene families F_w in which w is conserved to the median number of gene families F_{bg} in which random permutations of w are conserved as follows:

$$C = \left[1 - \frac{F_{bg}}{F_w}\right] \tag{1}$$

Stated more precisely, given a word w, the framework generates a large number (default value = 1000) of random permutations of w and establishes the number of gene families in which these random permutations are conserved. F_{bg} then denotes the median (or representative) value. Note that all information needed to calculate the background model has already been generated during the intra-family step and can be retrieved by simple lookup operations in the hash table. The background model F_{bg} can be seen as the expected number of gene families in which a word with the same length and character composition will be conserved. If the candidate motif w is conserved in many more families that what could be expected by chance, a high confidence C will be obtained. All words w with a confidence C that exceeds a threshold (default value = 90%) are retained and are considered true motifs.

The framework allows for the use of several conservation thresholds T_i ($i = 1 \ldots t$) in a single run. In that case, the hash table stores $< w, \overline{F_w} >$ pairs, where $\overline{F_w}$ now denotes a vector that holds the number of gene families in which w is conserved for each of the different thresholds T_i separately. A confidence value C is then obtained as the maximum confidence calculated over all thresholds T_i.

$$C = \max_{i=1\ldots t} \left[1 - \frac{\overline{F_{bg}}[i]}{\overline{F_w}[i]} \right] \tag{2}$$

The use of different thresholds provides for the detection of motifs that are significantly conserved in only a subset of the species (and thus reduced conservation threshold) in a single run and hence computationally efficient manner.

3 Distributed-memory, parallel implementation

For realistic datasets, the sequential algorithm described in the previous section is computationally demanding. Even though the total number of words N_w to consider scales linearly with the total input size ($N_w = O(N)$), the number of words to consider is huge. This results in very large runtimes for the sequential algorithm.

Each word w that is conserved in at least one gene family is stored in random access memory, along with the number of gene families F_w in which it is conserved, and this for a number of BLS thresholds. The clear advantage of this approach is a strong reduction in runtime during the calculation of the background model for each motif (see Section 2). The disadvantage is that the memory requirements exceed what can be typically provided by a single workstation. A parallel, distributed-memory framework (see Fig. 1) was developed to alleviate both the runtime and memory bottlenecks.

In the intra-family phase, the different gene families are uniformly distributed among the different parallel processes. Each process hence handles a subset of the gene families, and has a local hash table in which its $< w, \overline{F_w} >$ pairs are stored. This step is communication-free. At the end of this phase, a given word w can be contained by several processes, each holding only partial values in their respective $\overline{F_w}$ vectors.

In a single communication phase, these partial $\overline{F_w}$ vectors are accumulated. This is achieved by redistributing all words over the different processes, such that corresponding words are sent to the same process. That process can then sum the partial $\overline{F_w}$ vectors for each word w. Additionally, we partition the different words among the local memories of the different processes in such way that a given word and its permutations end up in the same process. For example, both the word w = CACGTG and w' = AGTGCC belong to the same *permutation group* and will end up in the same process. More specifically, for each word w, a hash value h is computed that only depends on the character composition of w, but not on the order of the characters within w. The words CACGTG and AGTGCC hence yield the same hash value h. This value is used to determine the process to which these words will be sent.

In order to obtain a uniform workload distribution during the inter-family phase, we assign a weight W_g to each permutation group g that corresponds to the maximum number of words represented by this permutation group:

$$W_g = \frac{(n_A + n_C + n_G + n_T + n_N)!}{n_A!\, n_C!\, n_G!\, n_T!\, n_N!},$$

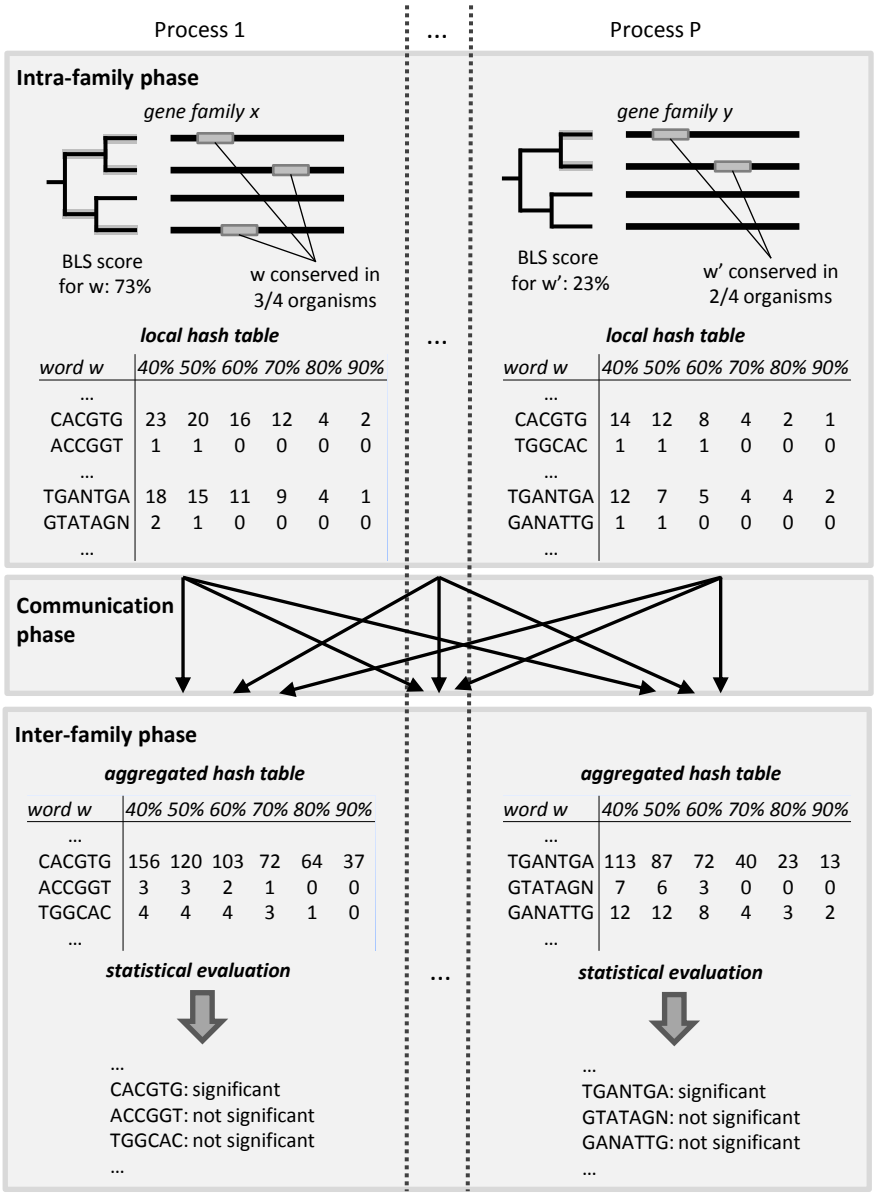

Fig. 1. Schematic overview of the parallel framework. In the intra-family step, the different gene families are distributed among the P different parallel processes and each process independently scores all words within those families for conservation. A local hash table stores all $< w, F_w >$ pairs which indicate in how many gene families w has been conserved. During a single communication step, words are shuffled between parallel processes such that a given word and its permutations end up in the same process. In the inter-family step, partial F_w vectors are aggregated and for each word w, the significance of conservation is determined.

where n_X denotes the number of characters X in a word of the permutation group. This weight is used to attribute roughly the same number of words to each process.

During the inter-family phase, the confidence values C are computed for each word w. This step can again be performed in parallel, as each process now holds different words. Note that because of the particular distribution of words during the previous step, this phase is now communication-free. Indeed, for every word w, all random permutations of w that are conserved are stored in the hash table of the same process. A certain random permutation that is not found in the local hash table, is not conserved in any gene family. To speed up the confidence calculations, only a single background model per permutation group is computed to which the candidate motifs are compared.

4 Results and Current Limitations

The framework was implemented in C/C++ and the Message Passing Interface (MPI) was used to handle the inter-node communication. All runs were performed on a computer cluster consisting of 25 nodes, each node containing two quad-core Intel Xeon L5420 CPUs and 16 GByte of RAM each (200 CPU cores and 400 GByte of RAM in total). The nodes communicate through a QDR Infiniband high-speed interconnection network.

As a dataset, we consider four monocot plant species: *Oryza sativa ssp. indica*, *Brachipodium distachyon*, *Sorghum bicolor* and *Zea mays*. Using the 'integrative orthology viewer' in PLAZA 2.5 [15, 16], the orthology relationships between these grasses were inferred. We extracted two datasets, one with an upstream promoter length $N_s = 500$ bp, and a second dataset with a promoter length $N_s = 2$ kbp. In total, the dataset consists of $F = 17\,724$ gene families and 163 064 regulatory sequences (counting both forward and reverse strands). All words that exist in these datasets were exhaustively enumerated. Words are expressed in a 5-letter degenerated alphabet: the four bases A, C, G, T and the N character. A maximum of three 'N' characters were allowed per word. Both datasets were run using the alignment-based and alignment-free mode. Table 1 provides details for each run.

Clearly, the alignment-free mode is computationally more intensive than the alignment-based mode. This is because the definition of 'conservation' during the intra-family step is much more relaxed in the alignment-free mode, hence yielding a much higher number of words that are found to be conserved and stored in the hash table. However, because the same definition of 'conservation' is used for both the candidate word w and its control motifs (i.e. random permutations used to build the background model), the statistical test during the inter-family phase is consistent, filtering only those motifs that are conserved in a much higher number of families than expected in a random dataset. In the remainder of this Section, we focus on the computational issues. We discuss the alignment-free run on the 500bp dataset.

Table 1. Motif discovery on four monocotyldon species. The dataset consists of 17 724 gene families. AF = alignment-free, AB = alignmnent-based.

Promoter length N_s	Modus	Number of parallel processes P	Runtime (walltime)	Number of words per family (avg.)	Number of sign. motifs
500 bp	AB	96	0h12	18 150	865 512
500 bp	AF	96	0h57	178 000	1 356 004
2 kbp	AB	96	0h30	28 000	902 983
2 kbp	AF	200	4h25	628 000	1 689 998

During the intra-family phase, each process scores the conservation of each word in a subset of all families. Using 96 parallel processes, each process is attributed roughly 185 families. This step takes 48 minutes or 85% of the total runtime. The load was found to be well-balanced, as the required time per family is roughly equal for each family individually. Within this step, only 6% of the time is spent on the initial construction of the GSTs whereas 94% of the time is spent in the discovery algorithm and the computation of the BLS values.

In the communication phase, all words are repartitioned among the different processes in such way that both a given word w and all permutations of w that were found during the intra-family step are attributed to the same process. The frequency vectors $\overline{F_w}$ corresponding to the same words are immediately merged to limit the memory overhead. The total time for this step is 8 minutes time or 14% of the total runtime. The actual time spent redistributing the data is 5 minutes or 9% of the total runtime, while the remaining time is spent on the packing and unpacking of the motif frequency vectors and the merging of corresponding motifs. Note that we use a high-speed Infiniband interconnection network, but that the use of an Ethernet network is also possible, as this step has only a limited contribution to the total runtime.

The inter-family phase is again communication-free and consist of the statistical testing of all words w. It required only 20 seconds or 0.6% of the total runtime.

Because no computations are duplicated in the parallel algorithm, and because the single communication step has only a limited contribution to the total runtime, we expect our algorithm to exhibit a significant speedup, compared to the sequential algorithm. Note that we cannot process the complete dataset on a single node, making it difficult to estimate the exact speedup. For smaller datasets however, we achieve a speedup of up to 120, using 256 parallel processes.

The main limitation of the framework however, lies in the increased memory requirements, compared to the sequential algorithm. Whereas the sequential algorithm requires only a single $< w,\overline{F_w} >$ pair for each individual word w, the parallel algorithm has additional memory requirements because several $< w,\overline{F_w} >$ pairs might be stored in the local memories of different parallel processes. This is the case when w is found to be conserved in several gene families, contained by different processes. Therefore, the aggregated memory requirements at the end of the intra-family step are higher than in the sequential algorithm. For the

largest simulation (2 kbp promoters and alignment-free mode), each of the 200 processes required almost 2 GByte of memory, yielding an aggregated memory requirement of roughly 400 GByte. Currently, this is the main limitation of the framework.

5 Conclusion and Future Research directions

In this contribution, we have presented a parallel framework for comparative motif discovery. The framework is word-based and gene-centric, as it takes a number of orthologous promoter sequences from related species as input. A measure of conservation can be defined in a flexible way. The framework allows for different alphabets (e.g. 4-letter alphabet, 4-letter alphabet + 'N' character, or even the full IUPAC alphabet) and provides for a statistical evaluation of candidate motifs based on count statistics. The framework can take advantage of large distributed-memory clusters in order to deal with high computational requirements.

Within this parallel framework, we have implemented two methodologies. First, an alignment-based approach where conservation is scored based on pregenerated multiple sequence alignments and second, an alignment-free approach where conservation does not depend on the relative position or orientation of the candidate motif. In both cases, the branch length score (BLS) was used to quantify conservation, taking the phylogenetic relationships between the organisms into account. Using this framework, we exhaustively processed four plant species.

The framework is implemented using the Message Passing Interface (MPI), but bears some conceptual resemblance with the map-reduce paradigm, where two compute phases are effectively separated by a single communication step. The framework can undoubtedly be cast in e.g. Hadoop's map-reduce implementation. The advantage of using such scheme, is that Hadoop provides for an automatic load balancing of both map and reduce phase and can recover from node failures. More importantly, map-reduce can operate out-of-core, streaming data to local hard disks if the local memory capacities turn out to be insufficient. This should, in turn allow for the handling of a larger number of organisms, or provide for more sensitive alphabets (e.g. the full IUPAC alphabet), with the ultimate goal of obtaining a comparative motif discovery method with increased sensitivity.

Acknowledgments This work was carried out using the Stevin Supercomputer Infrastructure at Ghent University, funded by Ghent University, the Hercules Foundation and the Flemish Government - department EWI. This research fits in the Multidisciplinary Research Partnership of Ghent University: Nucleotides to Networks (N2N).

References

1. Das, M.K. and Dai, H.-K.: A survey of DNA motif finding algorithms. BMC bioinformatics 8(Suppl 7), S21 (2007)
2. Blanchette, M. and Tompa, M.: Discovery of regulatory elements by a computational method for phylogenetic footprinting. Genome Research 12(5), 739-748 (2002)
3. Elemento, O. and Tavazoie, S.: Fast and systematic genome-wide discovery of conserved regulatory elements using a non-alignment based approach. Genome biology 6(2), R18 (2005)
4. Wu, J., Sieglaff, D.H., Gervin, J., and Xie, X.S.: Discovering regulatory motifs in the Plasmodium genome using comparative genomics. Bioinformatics 24(17), 1843-1849 (2008)
5. Sieglaff, D.H., Dunn, W.A., Xie, X.S., Megy, K., Marinotti, O., and James, A.A.: Comparative genomics allows the discovery of cis-regulatory elements in mosquitoes. Proceedings of the National Academy of Sciences 106(9), 3053-3058 (2009)
6. Kumar,L., Breakspear,A., Kistler,C., Ma,L.J. and Xie, X.: Systematic discovery of regulatory motifs in Fusarium graminearum by comparing four Fusarium genomes. BMC Genomics 11, 208 (2010)
7. Ettwiller, L., Paten, B., Souren, M., Loosli, F., Wittbrodt, J., and Birney, E.: The discovery, positioning and verification of a set of transcription-associated motifs in vertebrates. Genome biology 6(12), R104 (2005)
8. Xie, X., Lu, J., Kulbokas, E.J., Golub, T.R., Mootha, V., Lindblad-Toh, K., Lander, E.S. and Kellis, M.: Systematic discovery of regulatory motifs in human promoters and 3' UTRs by comparison of several mammals. Nature 434(7031), 338-345 (2005)
9. Bailey, T.L., Bodén, M., Buske, F.A., Frith, M., Grant, C.E., Clementi, L., Ren, J., Li, W.W. and Noble, W.S.: MEME SUITE: tools for motif discovery and searching. Nucleic Acids Research 37, 202-208 (2009)
10. Stark, A., Lin, M.F., Kheradpour, P., Pedersen, J.S., Parts, L., Carlson,J.W., Crosby,M.A., Rasmussen,M.D., Roy,S., Deoras,A.N., et al.: Discovery of functional elements in 12 Drosophila genomes using evolutionary signatures. Nature 450(7167), 219-232 (2007)
11. Gusfield, D.: Algorithms on strings, trees, and sequences: computer science and computational biology. Cambridge University Press (1997).
12. Giegerich, R., Kurtz, S., Stoye, J.: Efficient implementation of lazy suffix trees. Software: Practice and Experience 33(11), 1035-1049 (2003)
13. Marsan, L. and Sagot, M.F.: Algorithms for extracting structured motifs using a suffix tree with application to promoter and regulatory site consensus identification. Journal of Computuational Biology 7(3/4), 345-360 (2000).
14. Marschall, T., and Rahmann, S.: Efficient exact motif discovery. Bioinformatics 25(12), 356-364 (2009)
15. Van Bel, M., Proost, S., Wischnitzki, E., Movahedi, S., Scheerlinck, C., Van de Peer, Y. and Vandepoele, K.: Dissecting Plant Genomes with the PLAZA Comparative Genomics Platform. Plant physiology 158(2), 590-600 (2012)
16. Proost, S., Van Bel, M., Sterk, L., Billiau, K., Van Parys, T., Van de Peer, Y., Vandepoele, K.: PLAZA: a comparative genomics resource to study gene and genome evolution in plants. The Plant Cell 21, 3718-3731 (2009)