Suffix Tree Approach to Discover Conserved Non-Coding Sequences in Plants

Sairam Behera

Department of Computer Science and Engineering
University of Nebraska-Lincoln

2018 International Workshop on String Algorithms in Bioinformatics (StringBio)

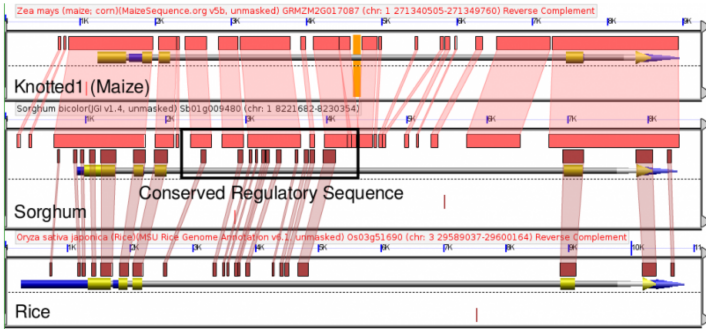
October 25, 2018

Conserved Non-Coding Sequence (CNS)

- A Region of the genome that does not code for proteins.
- Significantly slower rates of mutation than truly nonfunctional sequences.
- Involves in regulating the expression of neighboring genes.
- CNS in plants tend to be much smaller than animals.

Conserved Non-Coding Sequence (CNS)

- A Region of the genome that does not code for proteins.
- Significantly slower rates of mutation than truly nonfunctional sequences.
- Involves in regulating the expression of neighboring genes.
- CNS in plants tend to be much smaller than animals.



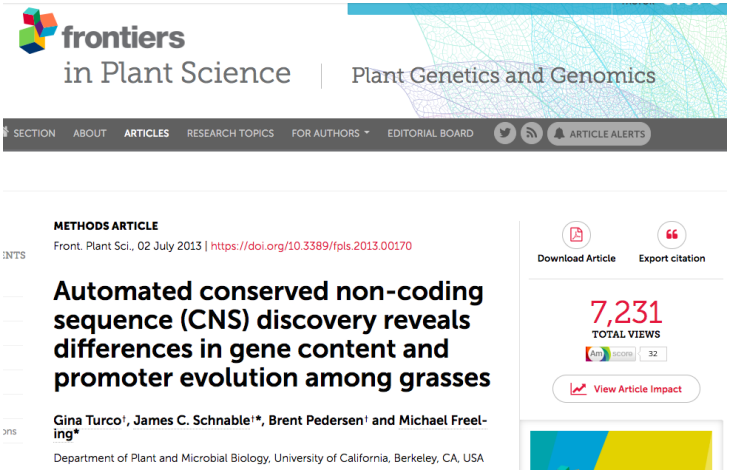
CNSs of Maize, Sorghum and Rice
(Source: `http://genomevolution.org/r/4bv3`)

CNS discovery approaches

- Manual Inspection of syntenic gene pairs.
- Pairwise genome alignment BLAST, LASTZ, QUOTA-ALIGN etc.
- Fast implementation of alignment plot method (Seaweed algorithm).
- Whole genome alignment using progressive alignment programs.
- Phylogenetic footprinting (Comparative motif mapping and alignment).

CNS discovery approaches

- Manual Inspection of syntenic gene pairs.
- Pairwise genome alignment BLAST, LASTZ, QUOTA-ALIGN etc.
- Fast implementation of alignment plot method (Seaweed algorithm).
- Whole genome alignment using progressive alignment programs.
- Phylogenetic footprinting (Comparative motif mapping and alignment).



Our Algorithm (DiCE)

- non-pairwise and non-alignment.
- Process any number of sequences simultaneously.
- Uses the maximal repeats from a generalized suffix tree of sequences to discover the CNSs.
- Identifies both exact matched CNSs as well as CNSs with a given mismatch rate.
- Compared and validated our approach using 17,996 syntenic genes of six grass species.
- Code is available at `https://github.com/srbehera11/DiCE`

Suffix Tree

- Many applications in computational biology
- Linear time construction algorithms

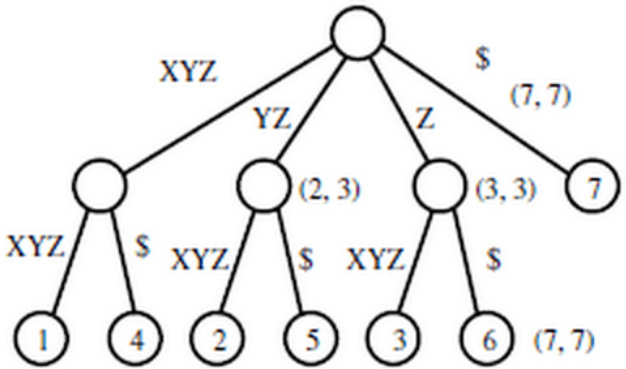
- Linear time solutions to
  - Genome alignment
  - Finding longest common substring
  - All-pairs suffix-prefix matching
  - **Locating all maximal repetitions**
  - And many more $\cdots$

Suffix Tree

Suffix: Given a sequence $S = s_1 s_2 \cdots s_n$, each subsequence ending at last character i.e. s_n is called a suffix.
Suffix Tree: Representation of all suffixes in a trie structure.
Example: S = XYZXYZ

XYZXYZ$
YZXYZ$
ZXYZ$
XYZ$
YZ$
Z$
$

(a)



(b)

(a) Suffixes of S (b) suffix tree

Generalized Suffix Tree

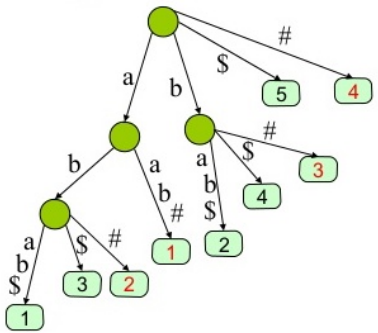
Given a set of sequences $S = \{S_1, S_2, \cdots S_n\}$, the **generalized suffix tree** is a suffix tree representation of all suffixes of all sequences.
Example: S_1 = abab\$, and S_2 = aab#

abab\$
bab\$
ab\$
b\$
\$

(c)

aab#
ab#
b#
#

(d)

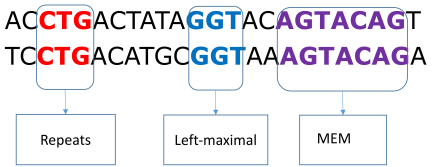


(e)

(a) Suffixes of S_1 (b) Suffixes of S_2 (c) generalized suffix tree

Maximal Exact Matching (MEM)

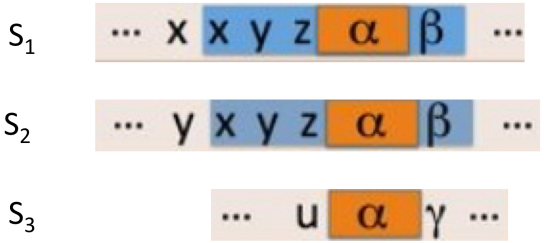
- A repeat is a subsequence if it occurs more than once in a sequence (or a group of sequences)
- A repeat is called left (or right) maximal if it cannot be extended to left (or right)
- Maximal Exact Match (MEM) is both left- and right-maximal
- Example:



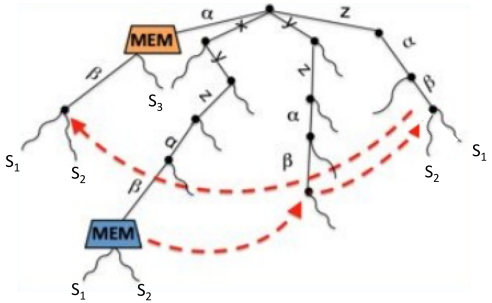
Maximal exact matching.

Identifying MEMs

- MEMs are internal nodes in the suffix tree that have left-diverse descendants. (have descendant leaves that represent suffixes with different characters preceding them).
- Linear-time suffix tree traversal to locate MEMs.



(f)

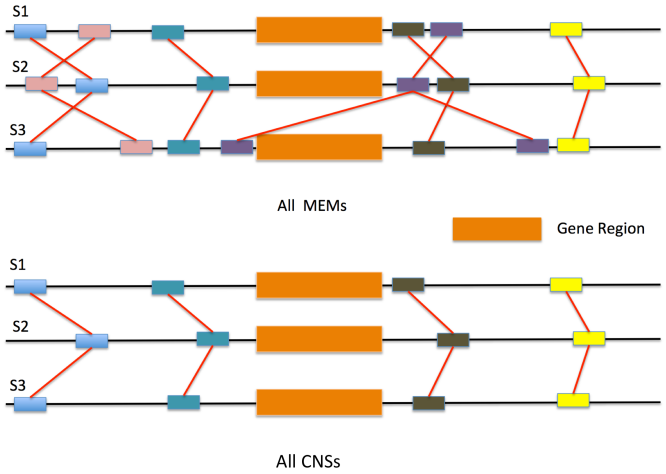


(g)

MEMs in (a) sequences and (b) suffix tree
(Source: SplitMEM; Marcus et al.)

CNSs and MEMs

- The CNSs are conserved regions i.e. repeated regions in all sequences.
- Finding CNSs in a group of sequences is to find order-consistent MEMs in all sequences.



MEMs and CNSs in three sequences

Problem Definition

- $\mathbb{M}_{\geq k,n}$ denotes a MEM that is present in all n sequences with length at least k.
- $\mathcal{M}$ is a set of all $\mathbb{M}_{\geq k,n}$.
- $\mathcal{C}$ is the set of CNSs ($\mathcal{C} \subseteq \mathcal{M}$).
- Define $score(\mathcal{C}) = \sum_{i=1}^{|\mathcal{C}|} length(C_i)$

Problem:
Given $\mathbb{S} = \{S_1, S_2 \cdots S_n\}$ a set of n sequences, find a set $\mathcal{C}$ such that $score(C)$ is maximum.

Algorithm

Algorithm 1: Finding CNS

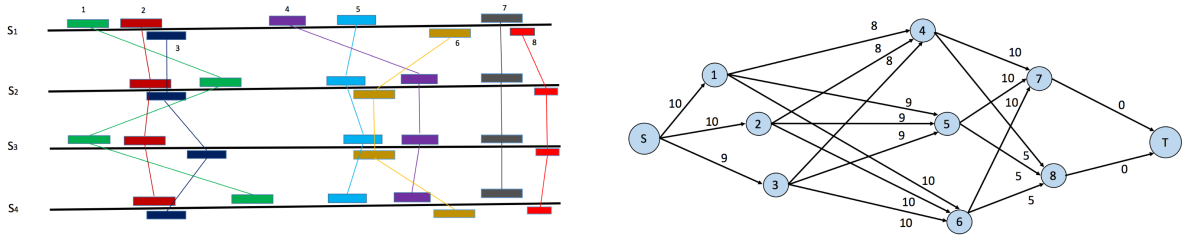
Input: Set of n sequences $S = \{S_1, S_2, \cdots S_n\}$
Output: A set of CNSs

1 Construct Generalized Suffix Tree $\mathscr{T}$ of sequences $S_1, S_2 \cdots S_n$ with suffix links
2 Mark the MEM nodes that represent $\mathbb{M}_{\geq k,n}$ and are not present in genomic regions
3 Extract the MEMs from the marked nodes in $\mathscr{T}$
4 Construct a weighted directed acyclic graph (DAG) using MEMs
5 Perform a topological ordering of the weighted DAG
6 Find the maximum weighted path $\mathbb{P}$ in the DAG
7 Store the MEMs associated with $\mathbb{P}$ in set $\mathscr{C}$
8 Output $\mathscr{C}$

Algorithm for finding CNSs

Directed Acyclic Graph (DAG) from MEMs

- Each MEM is a node in DAG.
- Directed edge between two non-intersecting, non-overlapping MEMs.
- Weight of each directed edge is the length of MEM it is pointing to.
- The start and end node are two imaginary nodes represent start and end of the sequence.

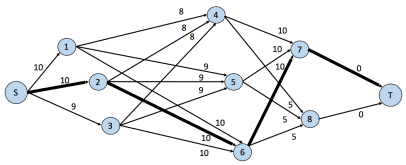


(h) MEMs (i) DAG

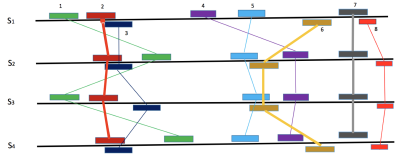
MEMs and corresponding directed acyclic graph

Finding CNSs

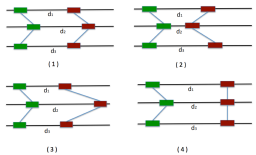
- Each directed path gives a set of independent MEMs with increasing order of start positions.
- The path that gives maximum weight is selected.
- The ties are broken based on smaller distance between two consecutive MEMs.



(j)



(k)



(l)

(a) Maximum weighted path (b) MEMs corresponding to the path (c) breaking the ties

Finding CNSs with mismatches

Algorithm 2: Finding CNS with mismatches

Input: Set of n sequences $S = \{S_1, S_2, \cdots S_n\}$
Output: A set of CNSs

1 Construct Generalized Suffix Tree $\mathscr{T}$ of sequences $S_1, S_2 \cdots S_n$ with suffix links
2 Mark the MEM nodes that represent $\mathbb{M}_{\geq k,l}$, where $2 \leq l \leq n$, and are not present in genomic regions
3 Extract the MEMs from the marked nodes in $\mathscr{T}$
4 Check if MEMs of type $\mathbb{M}_{\geq k,n}$ can be extended to construct pMEMs
5 Construct a weighted directed acyclic graph (DAG) using MEMs and pMEMs
6 Perform a topological ordering of the weighted DAG
7 Find the maximum weighted path $\mathbb{P}$ in the DAG
8 Store the MEMs and pMEMs associated with $\mathbb{P}$ in set $\mathscr{C}$
9 Output $\mathscr{C}$

Algorithm for finding CNSs with mismatches

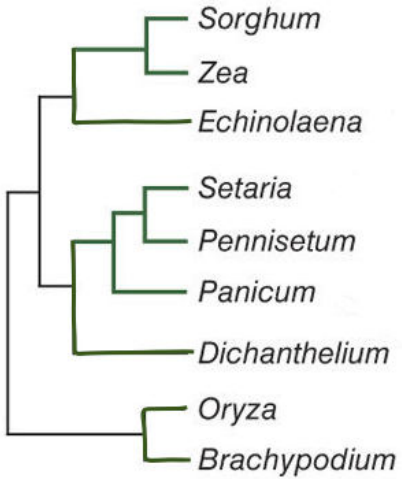
Implementation

- The algorithms are implemented in C++
- Inputs:
  - file containing fasta sequences with gene locations
  - minimum length of CNS
  - mismatch rate
- Output: list of CNSs
- The source code is available at `https://github.com/srbehera11/DiCE`

Results

The algorithms were tested on a single core of a Xeon E5-2697 v4 2.3GHz server with 2 CPU/36 cores per node and a total of 512GB of RAM at Holland Computing Center (HCC), University of Nebraska-Lincoln.

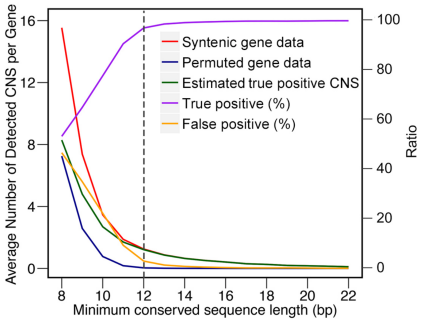
We tested our algorithm on 17, 996 syntenic genes of six grass species.



Grass phylogeny

Accuracy and Sensitivity of DiCE

- 200 genes, conserved at syntenic orthologous in *sorghum*, *rice*, *setaria*, *brachypodium*, *oropetium*, and *dichanthelium*, are selected.
- Exact matched CNSs with lengths 8 to 22 bp are identified at the 1kb upstream and downstream regions.
- The average false positive discovery rates per gene are estimated using 100 random permutations of the dataset.



True positive discovery rate.

Comparison of DiCE and CDP

- CNS Discovery Pipeline (CDP) [1] is one of the pipeline used for finding CNSs in grass species.
- CDP performs pairwise comparisons based on BLASTN, and then identifies CNS present in three or more species through overlap with a single common reference.

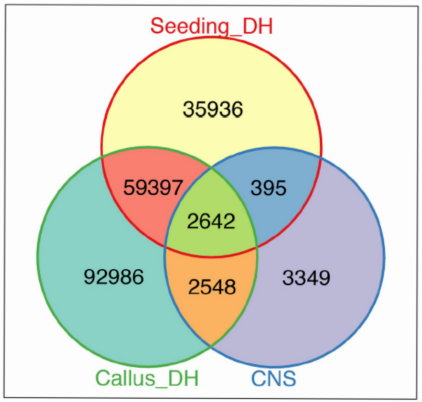
Summary of CNS distributions in 17996 Orthologous syntenic genes			
	CDP	DiCE (p = 0)	DiCE (p = 15)
CNS data	Sb-Si-Os	Sb-Si-Os	Sb-Si-Os
Total number of orthologous CNSs	22139	10498	22590
% of syntenic genes with at least 1 CNS	41.27% (7428)	27.91% (4968)	45.24% (8142)
Average number of CNSs	1.23 CNS/gene	0.58 CNS/gene	1.25 CNS/gene
Mean length of CNSs	35.76 bp	32.66 bp	44.72 bp
Median length of CNSs	27.00 bp	18.00 bp	32.00 bp
Total quantity of CNSs	791640 bp	162250 bp	804817 bp

CDP vs DiCE.

[1] https://genomevolution.org/wiki/index.php/CNS_Discovery_Pipeline

Association of CNSs with DHSs

- 8,934 CNSs identified from the syntenic genes in sorghum, rice, and setaria with the minimum CNS length 12 bp
- Compared with the DNase I hypersensitive sites (DHSs) of rice callus and seedings.
- 34.0% (3,037) and 58.1% (5,190) of CNSs overlapped with DH sites identified in seeding and callus tissues respectively.
- 25.7% and 41.6% of CNSs discovered by CDP overlapped.



Overlap of CNSs with DHS of rice.

Running Time

- Both CDP and DiCE were run on a single core of Xeon E5-2697 v4 2.3GHz server at HCC, University of Nebraska-Lincoln.
- The CDP took almost 24 hours to identify the CNSs among 17, 996 otrhologous syntenic gene sets of sorghum, setraia and rice.
- DiCE takes only 4.5 hours and 5 hours when it is tested with p = 0 and p = 15 respectively.

Conclusion

- Our algorithm, called DiCE
  - Not based on multiple pairwise alignments.
  - Identifies the CNSs from the maximal repeat fragments.
  - Sensitive to smaller CNSs found in plant species
  - Discovers the exact matches CNSs and CNSs with a given mismatch rate
  - 5 times faster than CDP, the best existing CNS discovery pipeline.
- Accuracy and sensitivity of the results are evaluated using a permutation test.
- Results are validated by comparing with previously known DH sites of rice seed and callus that are related to CNSs.

References

- S. Behera, X. Li, J. Schnable and J. S. Deogun, "DiCE: Discovery of conserved noncoding sequences efficiently," 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Kansas City, MO, USA, 2017, pp. 79-82. doi:10.1109/BIBM.2017.8217628
- X. Lai, S. Behera, Z. Liang, Y. Lu, J.S. Deogun, J.C. Schnable, "STAG-CNS: An Order-Aware Conserved Noncoding Sequences Discovery Tool for Arbitrary Numbers of Species", Mol. Plant., vol. 10, no. 7, pp. 990-999, 2017.

Thank You !!

Any Questions ?

@srbehera11

https://github.com/srbehera11