

Exploring the solution space of 1-Dollo Problem

Sarthak Mishra

April 24, 2023

Abstract

Cancer has a growth of cells that leads to disruption of normal cell functions. The changes in the genes that control the cell functions cause these uncontrolled growths. These cells mutate, causing them to divide faster than usual or preventing them from dying usually. These growth anomalies are called tumors. A cell can have different mutations occurring within the same cell, called intra-tumor heterogeneity. One of the evolutionary models used to understand these mutations is k - *Dollo*. From a theoretical perspective, the hardness of the k - *Dollo* problem, where $k \geq 1$, seems still open. This project has proposed a graph-based algorithm to understand and determine the solution space of 1 - *Dollo* phylogeny completion. The method is based on constructing a complete directed graph based on losses and gains required to reach one cell to another in the phylogenetic tree.

1-Dollo phylogeny .

1 Introduction

Cancer is a genetic disorder characterized by uncontrolled cell growth that disrupts the function of normal cells. Changes in the genes that regulate how cells work trigger it. These cells mutate, causing them to divide faster than average or preventing them from dying normally. Tumors are the term for these extra growths. Intra-tumor heterogeneity occurs when several distinct mutations occur within the same tumor. Phylogenetic trees are often used to model and explain cancer progression, with clinical consequences [8]. T is a phylogeny tree based on characters. The sequenced records, referred to as taxa, are the tree's leaves. Each taxon represents the collection of characters that have mutated as a result of our observations. A gain of character c indicates that any cell corresponding to the leaves under that edge has a character c mutation, assuming they do not back mutate later.[3] A back mutation, or loss of character c , means that any cell corresponding to the leaves under that edge did not have a character c mutation when it was sequenced. On the same edge, several characters can be obtained and lost. All sequenced data is assumed to begin with a single normal cell[3]. The cell mutates and divides as it grows, resulting in new cells with these mutations. These cells repeat the process, mutating on their

independence of the others, causing the intra-tumor heterogeneity [10]. For this project I am considering restrictive 1 – *Dollo* model where a SNV/character can be gained at most once and lost at most once[3].

I am introducing a Directed graph-based method. Each row of the input matrix is constructed as a vertex and has the edge on each other row. The weights are assigned to the edge such that they depict the losses and gains required to reach one cell from another. The method relies on modified DFS to check if all the vertices of the constructed graph are reachable from some vertices $v \in V$.

1.1 1-Dollo

[4] A rooted tree T is a 1-Dollo phylogeny for a matrix $B \in 0, 1^{m \times n}$ provided that

1. T has m leaves that uniquely correspond to rows of B .
2. The root r of T is labeled by vector $b_r = [0, 0, \dots, 0]^T$
3. For each character $c \in [n]$, there is at most one gain edge (u, v) in T such that $b_{u,c} = 0$ and $b_{v,c} = 1$.
4. For each character $c \in [n]$, there is at most one loss edge (u, v) in T such that $b_{u,c} = 1$ and $b_{v,c} = 0$.

1.2 Perfect Phylogeny

[11]Perfect Phylogeny Theorem: A binary matrix $B \in 0, 1^{m \times n}$ is a perfect phylogeny matrix if and only if it does not contain the submatrix or any permutation of its rows.

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

2 Unique SNVs

We are given a $m \times n$ binary matrix, B , where m represents the cells, and n represents the SNVs.

Let $P = \{\text{set of characters present in } p\}$ and $Q = \{\text{set of characters present in } q\}$

1. $\text{unique}_p(q)$ is the set of characters that taxon p possesses but that are not possessed by taxon q .
2. $\text{unique}_p(q) = P - Q$, i.e. the set of characters uniquely possessed by taxon p not by taxon q .

2.1 Relation of taxon P and Q

There can be three possible scenarios in tree T , for P and Q .

1. P is above Q : If P is above Q , then, along the line, there must be some edge where there is a loss of unique SNVs of P with respect to Q -unique_P(Q). Also, along the line, there must be some edge where there is a gain of unique SNVs of Q with respect to P -unique_Q(P).
2. Q is above P : If Q is above P , then, along the line, there must be some edge where there is a loss of unique SNVs of Q with respect to P -unique_Q(P). Also, along the line, there must be some edge where there is a gain of unique SNVs of P with respect to Q -unique_P(Q).
3. P and Q are on different branch of T : The third scenario is P , and Q is on a different branch in Tree, T . In this case, unique SNVs of P with respect to Q -unique_P(Q) are equal to P , and the unique SNVs of Q with respect to P -unique_Q(P) are equal to Q .

From this point I am going to use cells instead of taxon.

2.2 Relation between gain and loss

In order for P to be on top of Q , there must be an edge with a loss of unique_P(Q) and an edge with a gain of unique_Q(P). And in order for Q to be on top of P , there must be an edge with a loss of unique_Q(P) and an edge with a gain of unique_P(Q). The SNVs lost while moving from P to Q will be the SNVs gain while moving from Q to P .

2.3 Forming the complete directed Graph

Using the methods described above, we can compute all the Unique SNVs between pairs of cells and use the values of Uniques to create a complete directed graph $G = (V, E)$, $V \in Cells$, $(i, j) \in E \iff i \neq j$ and $(i, j) \in Cells$; weight $W(e) = -\text{unique}_i(j) + \text{unique}_j(i)$ where $e \in (i, j)$

For Example: If we are given B with 4 cells/rows

$Cell_1$
$Cell_2$
$Cell_3$
$Cell_4$

From	To	Weight $W(e)$	Represented by
Cell 1	Cell 2	$-\text{unique}_{\text{cell}_1}(\text{cell}_2) + \text{unique}_{\text{cell}_2}(\text{cell}_1)$	A
Cell 2	Cell 1	$-\text{unique}_{\text{cell}_2}(\text{cell}_1) + \text{unique}_{\text{cell}_1}(\text{cell}_2)$	-A
Cell 1	Cell 3	$-\text{unique}_{\text{cell}_1}(\text{cell}_3) + \text{unique}_{\text{cell}_3}(\text{cell}_1)$	B
Cell 3	Cell 1	$-\text{unique}_{\text{cell}_3}(\text{cell}_1) + \text{unique}_{\text{cell}_1}(\text{cell}_3)$	-B
Cell 1	Cell 4	$-\text{unique}_{\text{cell}_1}(\text{cell}_4) + \text{unique}_{\text{cell}_4}(\text{cell}_1)$	C
Cell 4	Cell 1	$-\text{unique}_{\text{cell}_4}(\text{cell}_1) + \text{unique}_{\text{cell}_1}(\text{cell}_4)$	-C
Cell 2	Cell 3	$-\text{unique}_{\text{cell}_2}(\text{cell}_3) + \text{unique}_{\text{cell}_3}(\text{cell}_2)$	D
Cell 3	Cell 2	$-\text{unique}_{\text{cell}_3}(\text{cell}_2) + \text{unique}_{\text{cell}_2}(\text{cell}_3)$	-D
Cell 2	Cell 4	$-\text{unique}_{\text{cell}_2}(\text{cell}_4) + \text{unique}_{\text{cell}_4}(\text{cell}_2)$	E
Cell 4	Cell 2	$-\text{unique}_{\text{cell}_4}(\text{cell}_2) + \text{unique}_{\text{cell}_2}(\text{cell}_4)$	-E
Cell 3	Cell 4	$-\text{unique}_{\text{cell}_3}(\text{cell}_4) + \text{unique}_{\text{cell}_4}(\text{cell}_3)$	F
Cell 4	Cell 3	$-\text{unique}_{\text{cell}_4}(\text{cell}_3) + \text{unique}_{\text{cell}_3}(\text{cell}_4)$	-F

Table 1: Representation for all weights with letters. From this point we will use the represented letters as weights.

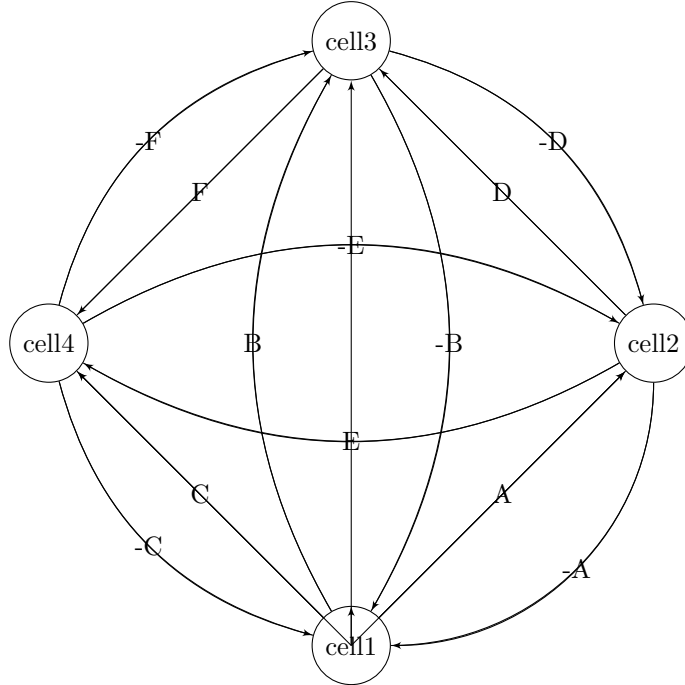


Figure 1: Complete directed graph created from table 1.

The above Graph can be represented as the matrix below.

	$Cell_1$	$Cell_2$	$Cell_3$	$Cell_4$
$Cell_1$	0	A	B	C
$Cell_2$	-A	0	D	E
$Cell_3$	-B	-D	0	F
$Cell_4$	-C	-E	-F	0

3 Restriction for 1 – *Dollo* Evolution Model

In 1 – *Dollo* evolution model, a SNV can be lost only once and gained only once[3].

3.1 Lost Constraint

As 1 – *Dollo* evolution model, says a SNV can be lost only once[3]. We are introducing a lost counter variable, L , that will be used to track the number of losses.

Initialization:

$$L = \{0, \dots, 0\} \text{ of size } n$$

Update:

1. If we are moving from $Cell_1$ to $Cell_2$ then L needs to be updated with $\text{unique}_{Cell_1}(Cell_2)$.
2. But the Loss Constraint for 1 – *Dollo* model restrict only one loss per SNV. Hence, we need a mechanism to check if the SNV has been lost or not.
3. To do that we can use check this condition X: if $\exists i \in \text{unique}_{Cell_1}(Cell_2)$ and $i \in L$
4. If the X is True, then the SNVs that are some SNV present in $\text{unique}_{Cell_1}(Cell_2)$ are already used in L . So this move will violate loss constraint.
5. If the X is False, then the SNVs that are present in $\text{unique}_{Cell_1}(Cell_2)$ are not used in L . So we union the elements of $\text{unique}_{Cell_1}(Cell_2)$ with L - $L = \text{unique}_{Cell_1}(Cell_2) \cup L$.

3.2 Gain Constraint

As 1 – *Dollo* evolution model, says a SNV can be gained only once[3]. This means the SNV cannot gained if it has been lost before.

1. If the SNVs is to be gained while going from $Cell_1$ to $Cell_2$ there can only one way the gain constraint can be broken. If there was a loss of the same SNV in the path then the gain constraint will be broken.
2. We can check it using condition X from 3.1. If X is True, then there has to be some SNVs that are required to be gained $Cell_1$ and $Cell_2$ have already been lost .Hence, making this move will violate the constraint.
3. If X is False, then are no SNVs lost that are required to be gained in $Cell_1$ and $Cell_2$. Hence, we can make the move from $Cell_1$ to $Cell_2$.

4 Type of Losses

4.1 Non-Conflicting losses

If all the uniques required to lose from a cell, for example, $cell_1$ to other cells in the matrix, follow the definition of perfect phylogeny, then the loss is called non-conflicting loss.

This means if the matrix created out of uniques from $cell_1$ i.e $unique_{Cell_1}(Cell_2)$, $unique_{Cell_1}(Cell_3)$, $unique_{Cell_1}(Cell_4)$ have no conflicting sub-matrix $\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$ or any of its permutations then this matrix will have no-conflicting losses.

4.2 Conflicting losses

Conflicting losses occur when a set of unique from a reference cell, for example, $cell_1$, to other cells in the matrix has conflict.This means there is a presence of conflicting sub-matrix $\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$ or any of its permutations in matrix created of $unique_{Cell_1}(Cell_2)$, $unique_{Cell_1}(Cell_3)$ and $unique_{Cell_1}(Cell_4)$.

4.3 Relation with 1 – *Dollo* Completion

For the input matrix B to have 1 – *Dollo* Completion, it has to have at least one configuration of cells where there are non-conflicting losses, and it also must have adhered gain constraint. Having non-conflicting losses is one of the necessary conditions for 1 – *Dollo* completion.

In order for matrix B to have 1 – *Dollo* completion , there must me at most one loss and one gain per SNV[3]. This means if matrix B has 1 – *Dollo* completion

it has non conflicting losses. i.e it cannot have a pair of losses: $\text{unique}_{\text{Cell}_1}(\text{Cell}_j)$ and $\text{unique}_{\text{Cell}_1}(\text{Cell}_h)$ such that there exists

1. $\text{Unique}_{\text{unique}_{\text{Cell}_1}(\text{Cell}_j)}(\text{Cell}_1(\text{Cell}_h)) \neq \emptyset$
2. $\text{Unique}_{\text{unique}_{\text{Cell}_1}(\text{Cell}_h)}(\text{Cell}_1(\text{Cell}_j)) \neq \emptyset$
3. $k \in [m]; k \neq j$ and $k \neq h$ and $k \neq i$ such that $\text{unique}_{\text{Cell}_1}(\text{Cell}_k)$ contains the some SNVs from both 1. and 2. then there is conflicting loss.

This means to reach j , h , and k from i , there needs to be a loss of at least two Unique SNVs, given above by in 1. and 2., and there needs to be a loss of both SNVs again in $\text{unique}_{\text{Cell}_1}(\text{Cell}_k)$. Hence this cannot have a 1-*Dollo* completion. This can be achieved by checking for conflicting submatrixes as mentioned in the definition of perfect phylogeny. We can check for conflicting losses by looking for conflicting sub-matrixes. Gussfield, used a greedy constructive algorithm in $O(mn)$ time to check this [11][3].

5 Algorithm

Now that we have defined all the constraints needed for the problem, we can explain the proposed algorithm.

1. First of all we calculate all the uniques between all the pairs of cells. We can achieve this by subtracting the set of SNVs of a cell from another.
2. Once we get all the uniques we check from non-conflicting losses. From section 4 we concluded that having non-conflicting losses is one of the necessary conditions for 1-*Dollo* completion. Hence, We use Gussfield's constructive approach to check for non-conflicting losses [11]. We need to check this for all the $i \in \text{Cells}$. This means we assume cell i is at the top and use the unique of i with respect to k , $\text{unique}_i(k)$ for all $k \in \text{Cells}$ and $k \neq i$. This will ensure one of the necessary conditions- loss constraint. Now we need to check for another necessary condition - grain constraint.
3. If there exists non-conflicting loss for some $i \in \text{Cells}$. Then we Form the complete directed graph using the uniques calculated in step 1 to check for gain constraint. We now form the Graph, $G = (V, E)$, $V \in \text{Cells}$, $(i, j) \in E \iff i \neq j$ and $(i, j) \in \text{Cells}$; *weight* $W(e) = -\text{unique}_i(j) + \text{unique}_j(i)$ where $e \in (i, j)$.
4. Now that we have constructed the Graph, $G = (V, E)$, we need to check the reachability of each vertex. In the Tree, T , one of the $\text{cells} = \text{Cell}_1, \text{Cell}_2, \text{Cell}_3, \text{Cell}_4$ has to be the topmost, or all of the cells must be siblings of each other. By now we have verified in step 2 that with cell $i \in \text{cells}$ there is a non-conflicting losses. This means the cell $i \in \text{cells}$ have set of losses that does not conflict while forming a tree with other cells. We need to use the same cell $i \in \text{cells}$ from step 2 to check for

reachability.

If we need to start from i and use a DFS to check if all the vertices are reachable under loss and gain constraint, we can reach the following conclusions:

- (a) if there is a vertex that is not reachable, we can be sure that there is no 1 – *Dollo* completion for that matrix.
- (b) If we find that all the vertex are reachable, then we can be sure there is a 1 – *Dollo* completion.

Algorithm 1 Main Algorithm

Input: Binary Matrix B with m – *cells* and n – *SNVs*

Output: If B has 1 – *Dollo* Completion

1: Calculate the

$\text{unique}_i(j)$

for all $(i, j) \in \text{rows/cells}$ and $i \neq j$

2: For all $m \in \text{cells}$ use Gussfield’s construction method [11] to check for non-conflicting losses/conflicting sub-matrix in uniques.

1. IF there is a non-conflicting loss condition for $m \in \text{cells}$ then check for Reachability, call Reachability checker, with Start state= m

(a) If Reachability checker returns True ,then there is a 1 – *Dollo* completion. Hence, Return True

(b) If Reachability checker returns False, then there is no 1 – *Dollo* comepletion. Hence, Return False.

2. Return False

Algorithm 2 Reachability checker

Input: Binary array *Unique* with size n [$Unique_{i,j_1} \dots Unique_{i,j_n}$] and start state

Output: IF all the cells are reachable from a single cell.

- 1: Initialize L - the loss counter variable as $\{0..0\}$ of size n and *visited* as a dictionary with keys as *cells* and values as 0
 - 2: Construct the graph $G = (V, E)$, $V \in Cells, (i, j) \in E \iff i \neq j$ and $(i, j) \in Cells$; *weight* $W(e) = -unique_i(j) + unique_j(i)$ where $e \in (i, j)$.
 - 3: Use DFS from startstate to search for reachability. The termination step of DFS is when all the vertices are reached. Constraint the search on Gain and Losses
 - 4: IF all vertices are not reached and the DFS completes Return False.
 - 5: If all vertices are reached then Return True.
-

5.1 Proof of Correctness

1. For the B to have a valid completion, there must be a way to accommodate each row/cell of B such that there can only be $1 - loss$ per SNV and $1 - gain$ per SNV throughout the tree, T .
2. In the tree, T , the edges connecting two cells have gains and losses.
 - (a) If $cell_1$ is on top of $cell_2$ in a tree, T , then there must be *edge/s* with losses of SNVs present in $cell_1$ but not in $cell_2$.
 - (b) And there must be *edge/s* with gains of SNVs present in $cell_2$ but not in $cell_1$.
 - (c) And there must be *edge/s* with gains of SNVs present in $cell_2$ but not in $cell_1$.

Hence graph $G = (V, E)$ of *figure1* will depict this translation correctly.

3. In order for a $cell_i \in cells$ to have a valid configuration in Tree T , there must not be any conflicting losses.
 - (a) if $Cell_i$ has to loose certain $x \in SNV$ to reach $cell_j$ and $cell_i$ has to looses certain $y \in SNV$ to reach $cell_k$ and where $cell_i, cell_j, cell_k \in cells$; $cell_i \neq cell_k$; $cell_i \neq cell_j$; $cell_j \neq cell_k$ and $x \neq y$
 - (b) And there is a $cell_h$ where $cell_h \in cells$; $cell_i \neq cell_h$; $cell_h \neq cell_j$; $cell_h \neq cell_k$ such that $cell_i$ has to loose both x and y to reach $cell_h$.

This means a set of losses need to be lost in two separate branches of the tree. Gussfield's construction can easily check this for checking conflicting sub-matrixes[11]. As conflicting sub-matrixes depict this condition. As, x needs to be lost while moving from $cell_i$ to $cell_j$ and $cell_i$ to $cell_h$ whereas it is not required to loose x while moving from $cell_i$ to $cell_k$. Similar is the case with y but it is only lost while moving from $cell_i$ to $cell_k$ and $cell_i$ to $cell_h$ whereas it is not required to loose y while moving from $cell_i$

to $cell_j$. Hence in order to construct a tree with $cell_i$, $cell_j$, $cell_k$ and $cell_h$ there needs to be a loss of x or y in multiple branches. This violates the loss constraint. This is similar to 2-state perfect phylogeny[11]. Hence, checking for conflicting sub-matrix in uniques of each cell to other cells can capture this condition. If there is a non-conflicting losses from $cell_i$ to other cells than one of the necessary conditions of 1 – *Dollo* completion is validated.

4. We have assured that there are non-conflicting losses from $cell_i$ to other cells we need to check from another necessary condition reachability to check for gain constraints.

- (a) Gain Constraint: We can only gain one SNV once throughout the tree.

There can be three scenarios:

- i. If an SNV is present in the $V = Cell_i$ (current cell). Then there is no need to gain it. Hence, we will not get a gain of that SNV while traversing from $V = Cell_i$.
- ii. If an SNV was never gained, then it can be gained without breaking the constraint.
- iii. If an SNV was gained in vertex before $V = Cell_i$ (current cell) and it is not present in the current cell, it cannot be gained again.

We can ensure the gain constraint by checking scenario (iii), by checking if the SNVs to be gained is present in Loss counter, L .

- (b) Loss Constraint: We can only lose one SNV once throughout the tree.

There can be three scenarios:

- i. If $V = Cell_i$ (current cell) does not contain the SNV. Then we cannot get a loss of that SNV while traversing from $V = Cell_i$.
- ii. If an SNV has never been lost, then it can be lost without breaking the constraint.
- iii. If an SNV has been lost before, then it cannot be lost.

We can ensure the loss constraint by checking scenario(ii) by checking if the SNVs to be lost is present in Loss Counter, L .

5. Reachability: We have ensured that the Graph, G , and constraints proposed to depict 1 – *Dollo* problem are correct. We have also proved that non-conflicting losses are one of the sufficient conditions for 1 – *Dollo* completion and connected the presence of non-conflicting losses with conflicting sub-matrix. Now, we can check if each vertex $v \in V$ can be reached from the vertex i , which satisfies the non-conflicting loss condition. We can use a DFS type search to ensure this. If we cannot find a sequence of paths such that all vertices in V are reached from start vertex i under constraints, then we can be sure that there is no 1 – *Dollo* completion. The proof to this is trivial. We are searching for a path in Graph G such

that all vertices are traversed in that path. This means we are checking for all the combinations of vertices such that the simultaneous gains and losses, weights required to traverse from one vertex to another are not breaking the constraint. If there is no path covering all the vertices, then there is no cell arrangement such that it has a $1 - Dollo$ completion. If we find a path such that all vertices in V are reached from i , then the resulting path is the tree T .

6 Run time Analysis

The unives can be calculated and the weight matrix for Graph G can be constructed in $m(m-1)/2$. The non-conflicting losses can be checked by Gussfield's greedy construction [11] in $O(nm)$ for each m cells. Suppose there are non-conflicting losses from $cell_i$ then reachability can be checked by going through all the combinations of cells where $cell_i$ is the first cell. Hence, a $(m-1)!$ possible combination of cells such that $cell_i$ is on the top. The run time for DFs is $O((m-1)! + m)$. This method can be improved to $O(nm^3)$ by greedily sorting the number of gains and losses from $cell_i$ in ascending order and choosing the $cell_j \in cells$ which has the least number of SNVs to be changed. Once we reach $cell_j$ we can again sort for a number of gains and losses from $cell_j$ to the remaining cells and find another cell with the least SNVs to change. The intuition behind this approach is that the cells with the least number of changes required should be closest together than the ones with more changes required in Tree T . I am exploring this approach.

7 Results and Conclusion

I have completed checking for 4x4 and 5x4 matrices. The test was generated by generating all 16 bits binary strings and partitioning it into 4 arrays of size 4. I have compared the results of 4x4 with (Xie's results)[3]. You can find it in Github repository https://github.com/smishra677/cs598TermProject_experiments I am looking forward to applying this approach further on matrices of any sizes.

8 Further works

Further work can be done to engineer a way to check for gains similar to the method used for conflicting losses. Another direction could be on to implement this approach for on completion of matrices with $k > 1$.

9 Reference

1. Bouckaert, R., Fischer, M., Wicke, K. (2020). Combinatorial perspectives on Dollo- k characters in phylogenetics. arXiv preprint arXiv:2011.13479.
2. Von Bell, M. (n.d.). POLYA'S ENUMERATION THEOREM ' AND ITS APPLICATIONS. Retrieved from <https://helda.helsinki.fi/bitstream/handle/10138/159032/GraduTiivistelma.pdf?sequence=3>
3. Xie, S. (2019, August 23). IDEALS @ Illinois: Towards characterizing the solution space of the 1-Dollo Phylogeny problem. <https://www.ideals.illinois.edu/handle/2142/104916>
4. Mohammed El-Kebir, SPhyR: tumor phylogeny estimation from single-cell sequencing data under loss and error, *Bioinformatics*, Volume 34, Issue 17, 01 September 2018, Pages i671–i679, <https://doi.org/10.1093/bioinformatics/bty589>
5. Modjtaba Ghorbani Mahin Songhori (2013) The Enumeration of Hetero-Fullerenes by Polya's Theorem, *Fullerenes, Nanotubes and Carbon Nanostructures*, 21:6, 460-471, DOI: 10.1080/1536383X.2011.629759
6. Read, R. (1987). Polya's Theorem and Its Progeny. *Mathematics Magazine*, 60(5), 275-282. doi:10.2307/2690407
7. Germundsson, Roger; Weisstein, Eric. "XOR". *MathWorld*. Wolfram Research. Retrieved 03 May 2021.
8. Russell Schwartz and Alejandro Schaffer. The evolution of tumour phylogenetics: principles and practice. *Nature Reviews Genetics*, 18, 02 2017. doi: 10.1038/nrg.2016.170.
9. James S. Farris. Phylogenetic Analysis Under Dollo's Law. *Systematic Biology*, 26(1):77–88, 03 1977. ISSN 1063-5157. doi: 10.1093/sysbio/26.1.77. URL <https://doi.org/10.1093/sysbio/26.1.77>.
10. Peter C. Nowell. The clonal evolution of tumor cell populations. *Science*, 194 (4260):23–28, 1976. ISSN 00368075, 10959203. URL <http://www.jstor.org/stable/1742535>.
11. Gusfield(1991)] Dan Gusfield. Efficient algorithms for inferring evolutionary trees. *Networks*, 21(1):19–28, 1991. doi: 10.1002/net.3230210104. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/net.3230210104>.