Reversal Distance Problem

Jayasurya Marasani
Department Of CSE [AI]
Amrita Vishwa Vidhya Peetham
Kerala, India
jayamarasani1729@gmail.com

***Abstract*— The Reversal Distance Problem is a complex process to determine the relationship between two genomes of two different species. Given two permutation genomes we need to find the shortest sequence of reversals to transform one into another. The shortest sequence can be found by sorting by reversals method and based on different sorting arrangements we need to take smallest possible reversals as the distance of the reversals. There are many algorithms, but the basic algorithm approach is Greedy Algorithm where it sorts the genome by best possible way, but it has larger time complexity so here comes the Breakpoints and Adjacencies where we find the those checking every element of the genome and count the number of break points and adjacencies. Then using sorting by reversals will reduce so many steps to compute the shortest distance.**

Keywords—Sorting by Reversals, Reversal, Permutations, Genomes, Breakpoints, Adjacencies.

I. INTRODUCTION

In genetics, computational molecular biology is mostly being used because it can help scientists to make prominent and noticeable connections in genes which would be impossible for humans to recognize or to find out. Often it is seen that individual nucleotide mutations have been the target of most classical algorithms, but other global rearrangements like shifts of full genomic fragments have been avoided. Genome arrangement by reversals thought of a worthy study to understand evolutionary distance of different species at the chromosome level. Algorithmic study of genome rearrangement by reversals has been widely mentioned since Watterson, Ewens, Hall, and Morgan introduced the first definition of the reversal distance problem in 1982.Genome rearrangement by reversals provides a decent technique for studying evolutionary history between two species.The reversal distance estimation between two genomes has been modeled as sorting by reversals problem and the problem has been proven to be NP-hard by Caprara in 1997.

A mouse genome is cut into about 300 large genomic fragments called synteny blocks. Different order of these synteny blocks results in human genome. Different shuffling of the ancient mammalian genome results in mouse and human genome. For example, in humans the chromosome 2 is built from fractions that are similar to mouse DNA residing on chromosomes 1,2,3,4,5,6,7,10,11,12,14, and 17. Changes in gene ordering is a result of genome rearrangement, a series of rearrangements can alter the genomic architecture of a species. The biggest challenge is analysing the history of these rearrangements in the mammalian genomes, a recent study of human and mouse genome infers that less than 250 genomic rearrangements have occurred since the parting of humans and mice approximately 80 million years ago. The study of genome rearrangement includes unveiling the conjunctional puzzle of finding a series of rearrangement that changes one genome into another. Genome rearrangements can happen through a variety of ways [2].

Genome rearrangement event is the most common mode of molecular evolution process in biological species. Although the rearrangement process is very complex and complicated. There are three basic operations in Genome rearrangements. They are reversal, translocation and transposition. In this paper, the most focus is on reversals and sorting the genome.The most common rearrangement events that happened in the evolution of mammalian species are Reversal and Translocation. Reversals reverse both the order and the direction of the genes in a segment inside a chromosome. Translocations exchange tails between two chromosomes and it is considered as always reciprocal, i.e., none of the two tails is empty.

Intense study has been performed over tough computational problems of reversal distance and transposition distance. Biologists are mainly interested in reversals scenario where minimum number of flipping is involved relying on the Principle of Parsimony: which tells us that when many possible explanations are available, the simplest approach has a higher probability of being right. When trying to construct an evolutionary relationship tree between species, they want to have the lowest possible number of ancestors for a species achievable because mutations are rare[2].

II. PROBLEM

A. Problem Introduction

There are two variations of this problem, signed permutation and unsigned permutation. For unsigned permutation, a genome is modeled as a permutation π with order n (i.e. a permutation of $\{1,2,\ldots,\Pi\}$), where n is the number of gene blocks in the genome. Let the permutation $\pi=\pi[1]\pi[2]\ldots\pi[n]$, the reversal operation $\rho(i,j)$ rearrange π into $\pi[1]\ldots\pi[i-1]\pi[j-1]\ldots\pi[i]\pi[j]\ldots\pi[n]$. For signed permutation π', each $\pi'[k]$ has either a positive or a negative sign. Each reversal operation p(i.j) not only rearrange π' but also negate the sign of $\pi'[k]$ for $i \leq k < j$. The problem of estimating reversal distance between two genomes is formulated as sorting permutation by reversal operation. That is, given π (or π', we want to find a reversals (Bafna and Pevzner; 1996). sorting sequence that uses minimum number of reversal to sort π(or π') into identity permutation (i.e. the permutation, 1 2 ... 11 for unsigned-permutation, and +1 +2 ... +n for signed permutation). We called the minimum number of reversal the reversal distance.

B. Problem Statement

A reversal of a permutation creates a new permutation by inverting some interval of the permutation; (5,2,3,1,4), (5,3,4,1,2), and (4,1,2,3,5) are all reversals of (5,3,2,1,4). The reversal distance between two permutations π and σ, written $drev(\pi,\sigma)$, is the minimum number of reversals required to transform π into σ (this assumes that π and σ have the same length).

Given: A collection of at most 5 pairs of permutations, all of which have length 10.

Return: The reversal distance between each permutation pair.

C. Definitions

Genomic rearrangement can be described as events that convert one given genomic permutation into another by a series of reversals. The order of genes in any given permutation can be expressed as Λ, where all the genes (elements) are uniquely represented as numbers[8].

$$\pi = \pi_1 \pi_2 \pi_3 \ldots \pi_n$$

where $\pi_i \in \{1, 2 \ldots n\}$ and $\pi_i \neq \pi_j \Leftrightarrow i \neq j$

A given permutation is said to be an identity permutation 'e' if all the elements are in order, i.e., e = 1 2 3 . . . n [9].

Padding of permutation: Addition of 0 and n + 1 in a given permutation of n length is called as padding of permutation [10]. A padded permutation is represented by Π_p

$$\Pi_p = (0\ \Pi_1\ \Pi_2\ \Pi_3 \ldots \Pi_n\ \Pi_{n+1})$$

Signed and Unsigned permutation: Gene orientation in any given sequence can be represented using + and - symbol; e.g. (-1, +2, -3 .... +n). Genome sequences carrying this information along with them are called signed permutation; otherwise, they are called unsigned [10]. By default, number 0 is considered as a positive value. A signed permutation can be converted into an unsigned permutation. Each positive element is replaced by the ordered pair (2x-1, 2x) and all the negative elements by the ordered pair (2x, 2x-1) and then padding the obtained permutation. The resulting permutation is called the extended unsigned permutation [7].

Adjacency and Breakpoints: In the given permutation, a pair(i, i+1) is said to be an adjacency if $|\pi(i)- \pi(i+1)|=1$, where $\pi(i)$ is the index of i^{th} element in π such that $1 \leq i \leq n$, i.e., they are consecutive otherwise the pair is said to be a breakpoint. An identity permutation has no breakpoints. For example: {1 2 4 6 3 5 7 8} in the given permutation pair (1,2) and (7,8) are adjacent. The pair (2, 4), (4, 6), (6, 3), (3, 5) and (5, 7) represent breakpoint in the given permutation.

D. Algorithm Implemented: Greedy Algorithm

- takes all possible reversals $\rho\iota$ on two breakpoints
- $\rho\iota$ = reversals that result the smallest $b(\pi)$
- $\pi\iota = \pi \cdot \rho\iota$
- while $b(\pi\iota) > 0$
- $\rho\iota$ = reversals that result the smallest $b(\pi\iota)$
- $\pi\iota = \pi\iota \cdot \rho\iota$

E. Code

```
def _get_reverse_array(s):
    reverse_arrays = []
    for i in range(len(s)-1):
        for j in range(i+1, len(s)):
            r_list = s[i:j+1]
            r_list.reverse()
            reverse_arrays.append(s[:i] + r_list + s[j+1:])
    return reverse_arrays

def _get_reversal_distance(s1, s2, distance):
    if s1 & s2:
        return distance
    new_s1 = set()
    for s in s1:
        reverse_arrays = _get_reverse_array(list(s))
        for r in reverse_arrays:
            new_s1.add(tuple(r))
    new_s2 = set()
    for s in s2:
        reverse_arrays = _get_reverse_array(list(s))
        for r in reverse_arrays:
            new_s2.add(tuple(r))
    distance += 2
    if s1 & new_s2:
        return distance-1
    if s2 & new_s1:
        return distance-1
    if new_s1 & new_s2:
        return distance
    distance = _get_reversal_distance(new_s1, new_s2, distance)
    return distance

if __name__ == "__main__":
    with open("C:\\Users\\JAYASURYAMARASANI\\Downloads\\rosalind_rear.txt", "r") as f:
        lines = [line.strip().split(" ") for line in f.readlines() if line.strip()]
    for i in range(0, len(lines), 2):
        a = [l for l in lines[i]]
        b = [l for l in lines[i+1]]
        print('The reversal distance for these given genomes is:')
        print (a)
        print (b)
        distance, s1, s2 = 0, set(), set()
        s1.add(tuple(a)), s2.add(tuple(b))
        d = _get_reversal_distance(s1, s2, distance)
        print(d)
    print("done!")
```

F. Output:

```
The reversal distance for these given genomes is:
['7', '3', '8', '5', '1', '6', '9', '4', '10', '2'
['3', '6', '4', '1', '10', '2', '9', '7', '8', '5'
5
The reversal distance for these given genomes is:
['10', '1', '9', '7', '8', '5', '6', '2', '4', '3'
['7', '1', '3', '4', '2', '10', '5', '8', '6', '9'
5
The reversal distance for these given genomes is:
['1', '3', '4', '6', '5', '8', '2', '7', '9', '10'
['4', '6', '2', '1', '8', '9', '5', '3', '10', '7'
7
The reversal distance for these given genomes is:
['4', '8', '5', '1', '3', '2', '10', '9', '6', '7'
['5', '4', '3', '8', '10', '1', '6', '2', '7', '9'
9
The reversal distance for these given genomes is:
['7', '4', '5', '10', '3', '8', '1', '6', '2', '9'
['10', '3', '9', '7', '8', '6', '2', '1', '5', '4'
5
done!
```

III. Conclusion

The Algorithm used is Greedy Algorithm. It is a natural algorithm where it goes step by step reversal which consumes

lot of time but other algorithms uses better approach with breakpoints and adjacencies. Those things reduce time complexity and works efficiently.

REFERENCES

[1] Rusu, Irena., Fertin, Guillaume., Tannier, Eric., Labarre , Anthony., Vialette, Stéphane. Combinatorics of Genome Rearrangements. United Kingdom: MIT Press, 2009.

[2] .N. Garg, M. Jain, S. Singhania, P. Prateek, P. Biswas and I. Chawla, "Sorting by Reversals: A Faster Approach for Building Overlap Forest," 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), 2019, pp. 436-441, doi: 10.1109/SPIN.2019.8711775.

[3] X. Yin and D. Zhu, "Sorting Genomes by Reversals and Translocations," 2009 Asia-Pacific Conference on Information Processing, 2009, pp. 391-394, doi: 10.1109/APCIP.2009.233.

[4] X. Qi, G. Li, S. Li and Y. Xu, "Sorting Genomes by Reciprocal Translocations, Insertions, and Deletions," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 7, no. 2, pp. 365-374, April-June 2010, doi: 10.1109/TCBB.2008.53.

[5] Caprara. “Sorting by reversals is difficult”. in Proceedings of the 1st Conference on Computational Molecular Biology (RECOMB97), pages 75–83, Santa Fe, NM,1997. ACM Press.

[6] Bergeron. “A very elementary presentation of the Hannenhalli–Pevzner theory”, Discrete Applied Mathematics 146, 134-145, 2005.

[7] D.A. Bader, B.M.E. Moret and M. Yan, “A Linear-Time Algorithm for Computing Inversion Distance between Signed Permutations with an Experimental Study”, Journal of computational biology: a journal of computational molecular cell biology, 8 5, 483-91, 2001

[8] V. Bafna and P.A. Pevzner. “Genome rearrangements and sorting by reversals”. SIAM Journal on Computing, 25:272– 289, 1996.

[9] W.H. Gates and C.H. PAPADIMITRIOU, “Bounds for Sorting by Prefix Reversal” Discrete Mathematics, volume 27 issue 1, 47-57,1979

[10] S. Hannenhalli and P.A. Pevzner, “Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals)” in Proceedings of the27th Annual Symposium on Theory of Computing (STOC95), pages 178–189, Las Vegas, NV, 1995. ACM Press.

[11] Park, Euna, "Exact and Approximation Algorithms for Computing Reversal Distances in Genome Rearrangement" (2008). Master's Projects. 104.
DOI:https://doi.org/10.31979/etd.qm9e-d3gt
https://scholarworks.sjsu.edu/etd_projects/104

[12] Park, Euna, "Exact and Approximation Algorithms for Computing Reversal Distances in Genome Rearrangement" (2008). Master's Projects. 104.

[13] rosalind-solutions/rosalind_rear.py at master · zonghui0228/rosalind-solutions · GitHub

[14] http://rosalind.info/problems/rear/

[15] https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1103&context=etd_projects

[16] https://link.springer.com/chapter/10.1007/978-3-030-42266-0_3

[17] https://arxiv.org/ftp/cs/papers/0405/0405014.pdf

[18] http://cs.brown.edu/courses/csci2950-c/Fall2008/LectureSlides/Lecture4.ppt

[19] http://www.cs.unc.edu/~prins/Classes/555/Media/Lec06.pdf

[20] https://www.geeksforgeeks.org/program-for-array-rotation-continued-reversal-algorithm/

[21] https://www.youtube.com/channel/UCKSUVRs2N2FdDNvQoRWKhoQ