An Application of Combinatorial Problems to Build up Simulating Models of Rare Diseases: A Case of Multiple Pathogenic Variants

Nguyen Thi Ngoc Nhi
Lancaster Mennonite School,
Lancaster, Pennsylvania 17602, United States

Nguyen Dinh Lau
University of Education and Science, University of Da Nang, Vietnam

ABSTRACT
Combinatorics is a fundamental field in the high school math curricula of most countries worldwide, including Vietnam and the United States. It stems from basic concepts of sets of elements, mapping functions, combinations, permutations, and others to problems of listing all binary sequences of *n* elements, listing all *r*-combinations of n elements, listing all permutations of n elements. These combinatorial problems assisted with programming languages and machine learning models (emerging machine learning (ML)) have processed numerous applications in various areas of modern life, especially rare diseases as defined by the United States and the European Union. There are different disease-causing variants of rare diseases, so they need to be simulated to come up with appropriate treatment plans. In this paper, a specific synthetic mathematical model which is the problem of listing permutations on many pathogenic variants to simulate pathogenic factors is constructed. It is hoped that the result of this study can provide some suggestion for rare disease prevention and treatments.

General Terms
Simulating, Model, Algorithms, Combinatorial problems, rare diseases.

Keywords
Permutation, pathogen, Blocks, Variants, Client, Server.

1. INTRODUCTION
Rare diseases are defined as "any disease or condition means any disease or condition which (a) affect less than 200,000 persons in the U.S or (b) affects more than 200,000 persons in the U.S but for which there is no reasonable expectation that the cost of developing and making available in the U.S" (p.120) [1]. In other words, rare diseases are diseases with very low incidence in the total global population. There are approximately 6,000 rare diseases in the world that affect 300 million people - equivalent to the total population of the United States. Of these rare diseases, up to 80% are caused by genetics, of which fewer than 200,000 people are diagnosed with the disease [2]. Therefore, to solve the problem of rare diseases and improve the effectiveness of rare disease management, the participation of the entire social system is extremely urgent and needed, because these issues require huge resources and enormous efforts from many parties. With the idea of simulating all different cases of pathogenic variants, this combinatorial mathematical model is applied to list all cases where pathogenic variants are present in rare diseases. One of challenges faced the world is that there are abundant rare diseases with different causative factors and these factors are increasingly diverse and complex in both structure and quantity. To identify which pathogenic factors are root or have a large impact on rare diseases, it is crucial to list all the pathogenic variants to simulate and examine, analyze, evaluate, compare and finally suggest appropriate treatments. By studying and researching the problem of listing all permutations of *n* elements, the problem of listing all permutations of *n* elements (*n* pathogenic variants) have been applied to simulate all different cases of all variants. It is evident that if there are *n* pathogenic factors with n<=10, the problem can be solved feasibly because listing all the permutation arrays of *n* elements will give *n!* arrays. However, if n>10, the number of arrays is too large. In this paper. In this paper, the block division model is exploited to solve n>10 [3], [4], [5].

2. COMBINATORIAL MODEL AND THE PROBLEM OF LISTING ALL PERMUTATIONS

2.1 Definitions
- Loop arrangement:

Definition 1. A *k*-loop arrangement of *n* elements is an ordered set of *k* elements selected from a given *n* elements. The elements may be repeated.

A k-loop arrangement of *n* elements can be viewed as an element of the Cartesian power X^k, where X is a set of *n* elements. Thus, the total number of a *k*-loop arrangement of *n* elements is $\boldsymbol{n^k}$.

Example 1. Let $X=\{x_1,x_2,\ldots,x_n\}$ be a set with n elements. Find the number of subsets of set X.

Represent each subset A of the set X by a binary array of length *n*, denoted as $b=(b_1, b_2, \ldots, b_n)$. In this array, if $b_i=1$, then $x_i \in A$; if $b_i=0$, then $x_i \notin A$. The number of such binary arrays is 2^n, so the number of subsets A is also 2^n.

- Arrangement without a loop

Definition 2. A *k* arrangement without the loop of *n* elements is an ordered set of *k* elements selected from a given *n* elements. The elements cannot be repeated.

A *k* arrangement without the loop of *n* elements can be constructed through the following *k* steps:

- Step 1: Select the first section: *n* possibilities.
- Step 2: Select the second section: *n-1* possibilities.

...

- Step *k*: Select the k^{th} section: *n - k + 1* possibilities.

Thus, according to the multiplication principle, the total number of arrangements without the loop of *k* elements chosen from *n* elements is:

$$A(n,k) = n.(n-1).\ldots.(n-k+1) = \frac{n!}{(n-k)!}$$

Example 2. Determine the injective function from the set X with *k* elements to the set Y with *n* elements (k≤n).
Indeed, there is a single injective function from X to Y corresponding to a *k* arrangement without loop of *n* elements of Y. Hence, the number of injective function is A= n.(n-1).....(n-k+1).

- Permutations

Definition 3. A permutation of *n* elements is a way to arrange these elements in order.
Permutations can be considered as special cases of *k* arrangements without the loop of *n* elements where k = n. The number of permutations as *P(n) = n!*.

Example 3. There are ten people lining up in a row to take portraits. How many arrangements are there for the orders in which each person stands in the line?
Each portrait is a permutation of 10 people so the total number of portraits is 10!.

- Combination

Definition 4. A *k*-combination of *n* elements is a disordered set of *k* elements selected from a given *n* elements.
In other words, a *k*-combination of *n* elements can be considered as a *k*-element subset from the given n elements.
Let the number of *k*-combinations of n elements be C(n, k) and show that:

*A(n,k) = C(n,k) * k! (k!* is number of permutations*)*.

Therefore:

$$C(n,k) = \frac{n!}{k!\,(n-k)!}$$

Example 4. There are *n* football teams competing in an all-play-all tournament. How many football matches must be held in total?
Each match corresponds to a 2-combination of *n*. Hence, there are C(n,2) matches.

2.2 The problem of listing all perpetuations

According to definition 3, permutations are arrangements of the orders of certain elements, so the number of permutations is n! For example, if n=3, then there are 6 permutation arrays of 3!=6 as follows: P(1,2,3); P(1,3,2); P(2,1,3); P(2,3,1); P(3,1,2); P(3,2,1).
When n=4 then there are 24 permutation arrays 4!=24 as in Table 1:

Table 1. Permutation arrays of *n* elements

No	Permutations	No	Permutations
1	P(1,2,3,4)	13	P(3,1,2,4)
2	P(1,2,4,3)	14	P(3,1,4,2)
3	P(1,3,2,4)	15	P(3,2,1,4)
4	P(1,3,4,2)	16	P(3,2,4,1)
5	P(1,4,2,3)	17	P(3,4,1,2)
6	P(1,4,3,2)	18	P(3,4,2,1)
7	P(2,1,3,4)	19	P(4,1,2,3)
8	P(2,1,4,3)	20	P(4,1,3,2)
9	P(2,3,1,4)	21	P(4,2,1,3)
10	P(2,3,4,1)	22	P(4,2,3,1)
11	P(2,4,1,3)	23	P(4,3,1,2)
12	P(2,4,3,1)	24	P(4,3,2,1)

When *n* is large, the number of permutation arrays is highly huge, so it is needed to build an algorithm to run on some languages such as C, Python, Java, and others to determine the permutation arrays. Figure 1 is a block diagram to list the *n*-element permutations.

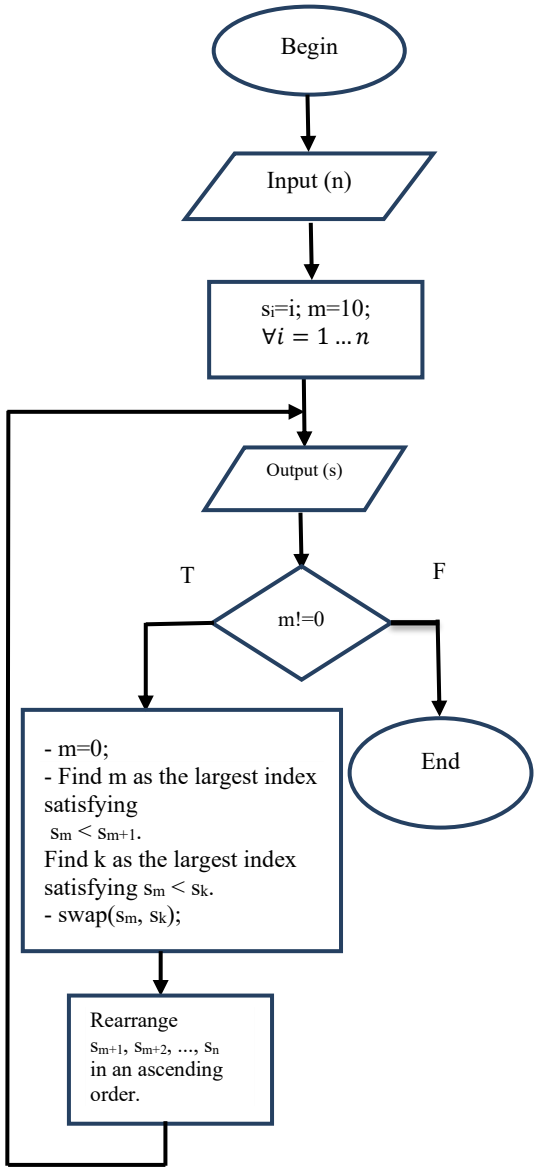

Fig 1: A block diagram listing the *n*-element permutations

2.3 Finding permutation blocks

In section 2.2, an algorithm is presented to find all permutations of *n* elements. But it is highly time-consuming because there are *n!* different permutation arrays. In section 2.3, there is an algorithm for a block of different permutation arrays with input s = [s_1 , s_2 ,... ,s_n] and g = [g_1 , g_2 ,... ,g_n], with the condition s<g in lexicographic order. The algorithm will list all permutations t = [t_1 , t_2 ,... ,t_n] in ascending order from s to g (s<t<g) in lexicographic order.
Given the permutation s = [s_1 , s_2 ,... ,s_n], the next permutation t = [t_1 , t_2 ,... ,t_n]. Moving from right to left, the first element s_m satisfying $s_m < s_{m+1}$. Then it is the largest index k (k > m) that satisfies $s_m < s_k$.
Then let:

$t_i = s_i$ for all i = 1,2,...,m-1;
$t_m = s_k$;

n-m the next elements are the remaining numbers arranged in ascending order.

Algorithm 1. Listing permutation blocks

==========

Input: Two border arrays s = [$s_1, s_2, \ldots, s_n$], g = [$g_1, g_2, \ldots, g_n$]
Output: all arrays t = [$t_1, t_2, \ldots, t_n$]: s<t<g
Steps:

```
While (t<>g)
        //Find m.
        for(int i=n; i>=1;i--)
                if(s[i]>s[i-1])
                {
                        m=i-1;
                        break;
                }
        //Find k.
        for(int i=n; i>=1;i--)
                if(s[m]<s[i])
                {
                        k=i;
                        break;
                }
        //swap the positions of s[m], s[k].
        swap(s[m], s[k]);
// arrange the arrays of elements in ascending order.
        int j=n;
        for(int i=m+1;i<j;i++)
                {
                        swap(s[i], s[j]);
                        j--;
                }
        t=s;
        Print the array.
}//end while.
```

==========

Consider the problem of listing all permutations of n elements {*1,2,...,n*}. Each permutation will be represented as the sequence $s_1, s_2, \ldots, s_n$. Thus the first permutation is [*1,2,...,n*] and the last permutation is [*n,n-1,...,1*].

Suppose it is required to find the permutation $t = [t_1, t_2, t_3, t_4, t_5, t_6]$ that follows the permutation s = [1, 6, 3, 5, 4, 2].

The first 4 numbers of *t* cannot be 1, 6, 3, 5 Because in the permutations starting with 1, 6, 3, 5 (there are only s and [1, 6, 3, 5, 2, 4]), s has the largest order.

The first 3 numbers of *t* also cannot be 1, 6, 3. Indeed, the permutations starting with 1, 6, 3 are:

s=[1, 6, 3, 5, 4, 2], [1, 6, 3, 5, 2, 4], [1, 6, 3, 4, 2, 5], [1,6,3,4,5,2], [1,6,3,2,5, 4], [1, 6, 3, 2, 4, 5].

- And among those permutations, it is obvious that *s* is the permutation with the largest order number.
- *t* cannot start with 1, 6, 3, 5 or 1, 6, 3 because in both cases the remaining numbers of *s* form a descending array.

So starting from the right, the first number *d* is smaller than the number to its right. In this example it is 3. And *t* starts with 1, 6. The next number of *t* must be greater than 3. Since let *t* to be the smallest permutation of all permutations greater than s, the next number must be the smallest of the remaining numbers (ie minus 1 and 6) greater than 3. That number must be 4. The remaining three numbers must be in ascending order. So let t = [1, 6, 4, 2, 3, 5]. The above process repeats until t=g then ends.

3. RARE DISEASES AND PATHOGENIC VARIANTS

As mentioned in the introduction, there are numerous rare diseases, also known as "orphan diseases", and thousands of different types. Diverse nature of pathogenic variants that are pivotal in causing diseases leads to great complexity in diagnosis and treatments. There is an ever-expanding list of mutations in about 4,000 genes causing about 6,500 different diseases [6].

Unfortunately, many suspected cases are not diagnosed or misdiagnosed due to the lack of appropriate clinical and diagnostic resources in many countries. As a result, patients face a major psycho-social-economic crisis and often suffer from the disease for life [7].

Although there are advanced drugs in the world to treat the root cause of a few rare disease due to genetics, these drugs have not yet been approved for use in Vietnam, or even if patients still have difficulty accessing approved drugs because they are not covered by national health insurance.[8]

Pathogenic variants of rare diseases are diverse and complex, but are primarily genetic. The phenotypic, genetic, and environmental characteristics that define a particular disease are often established in different demographic groups, regions, or settings. To better diagnose and care for patients with rare diseases, there should be an agreement on a definition of the unique combination of genetic, phenotypic, and environmental attributes associated with each rare disease [8].

According to the website of Da Nang General Hospital [9]. In Vietnam, there are currently about 100 rare diseases and 6 million people suffering from rare diseases, of which up to 58% of rare diseases appear in children, 30% of children with rare diseases die before the age of 5. Access to rare disease treatment is not only challenging in Vietnam but also in many other developing countries.

Due to their rarity and complexity, rare diseases pose a major challenge to medicine in diagnosis and treatment. Only 5% of rare diseases have FDA-approved drugs. The cost of treating rare diseases is often high-priced, and most patients cannot afford it without support from the state budget or from non-profit organizations and society, especially for long-term treatments.

Below is a table of pathogenic variants of rare diseases.

Table 2. Pathogenic variants of rare diseases

No	Detailed names	Symbols	Groups	Clarification
1	Genetic Mutation	GM	Genetic	Alternations in DNA sequence
2	Chromosomal Abnormalities	CA		Numerical or structural chromosomal anomalies
3	Non-Genic Mutations	NGM		Mutations occurring outside canonical gene regions
4	Monogenic Disorders	MD		Mutations affecting a single gene
5	Toxic Exposure	TE	Environmental	Exposure to chemical toxins, heavy metals, or radioactive agents
6	Infectious Agents	IA		Disease caused by viral, bacterial, or parasitic infections
7	Genetic Susceptibility	GS	Gene-Environment Interaction	Mutated genes that manifest disease only under specific environmental triggers
8	Allergic and Immune Responses	AIR		Aberrant immune responses leading to rare autoimmune conditions
9	Idiopathic	I	Other factors	Rare diseases with unknown etiology

10	Metabolic Dysregulation	MD		Disorders caused by enzyme or protein deficiencies
11	Daily Diet	DD		Disease influenced by nutrition and dietary patterns
12	Age-Related Factors	ARF		Disease risk associated with specific age groups

4. PROPOSED SIMULATION MODEL

Suppose there are four main disease-causing factors (1, 2, 3, 4) that influence the development of a rare disease, specifically Retinitis Pigmentosa (RP).

Retinitis Pigmentosa, also known as night blindness, is a group of disorders affecting the retina. RP typically leads to the destruction of rod cells in the retina, causing a gradual loss of vision and, in severe cases, eventual blindness. The disease is characterized by degeneration of the retinal pigment epithelium, which may begin at birth or later in life. The progressive deterioration of rod cells narrows the visual field, followed by degeneration of cone cells, resulting in severe vision impairment [9], [10].

The four main factors are:

1. Genetic Mutation (GM).
2. Chromosomal Abnormalities (CA).
3. Toxin Exposure (TE).
4. Daily Diet (DD).

Identification of possible permutations:

For the four variants 1, 2, 3, 4, there is a likelihood to enumerate all possible permutations in the simulation model. A permutation represents any possible arrangement of these variants.

The possible permutations are:

Table 3. Arrays of Four variants Through All Permutations

No	Permutation	Variants
1	(1,2,3,4)	(GM-->CA-->TE-->DD)
2	(1,2,4,3)	(GM-->CA-->DD-->TE)
3	(1,3,2,4)	(GM-->TE-->CA-->DD)
4	(1,3,4,2)	(GM-->TE-->DD-->CA)
5	(1,4,2,3)	(GM-->DD-->CA-->TE)
6	(1,4,3,2)	(GM-->DD-->TE-->CA)
7	(2,1,3,4)	(CA-->GM-->TE-->DD)
8	(2,1,4,3)	(CA-->GM-->DD-->TE)
9	(2,3,1,4)	(CA-->TE-->GM-->DD)
10	(2,3,4,1)	(CA-->TE-->DD-->GM)
11	(2,4,1,3)	(CA-->DD-->GM-->TE)
12	(2,4,3,1)	(CA-->DD-->TE-->GM)
13	(3,1,2,4)	(TE-->GM-->CA-->DD)
14	(3,1,4,2)	(TE-->GM-->DD-->CA)
15	(3,2,1,4)	(TE-->CA-->GM-->DD)
16	(3,2,4,1)	(TE-->CA-->DD-->GM)
17	(3,4,1,2)	(TE-->DD-->GM-->CA)
18	(3,4,2,1)	(TE-->DD-->CA-->GM)
19	(4,1,2,3)	(DD-->GM-->CA-->TE)
20	(4,1,3,2)	(DD-->GM-->TE-->CA)
21	(4,2,1,3)	(DD-->CA-->GM-->TE)
22	(4,2,3,1)	(DD-->CA-->TE-->GM)
23	(4,3,1,2)	(DD-->TE-->GM-->CA)
24	(4,3,2,1)	(DD-->TE-->CA-->GM)

Each permutation represents a different scenario in the simulation model. For example: Permutation (1,2,3,4) corresponding to (GM → CA → TE → DD): the model will begin by examining the effect of Genetic Mutation (GM), followed by Chromosomal Abnormalities (CA), then Toxin Exposure (TE), and finally Daily Diet (DD).

After running the simulation for each permutation, the results are analyzed and compared to determine which permutation has the greatest impact on disease progression. By simulating each permutation and comparing the outcomes, the most influential variant in disease development and the sequence of their effects can be identified. This helps prioritize the contributing variants and provides a better understanding of the disease mechanism [11].

For permutation (1,2,3,4) → (GM → CA → TE → DD), the model executes as follows:

- The model starts by assessing the activity level of genes based on genetic mutations.
- Next, it examines the extent of chromosomal abnormalities.
- Then, it calculates the level of patient exposure to toxins.
- Finally, it considers the patient's age.

Thus, the influence of genetic mutations is evaluated first, followed by chromosomal abnormalities, toxin exposure, and finally the patient's age. Simulations are run for each permutation using different values of the variants (e.g., various types of genetic mutations, chromosomal abnormalities, toxin exposures, and age ranges). From the results, which permutation yields the highest severity of Retinitis Pigmentosa can be determined, thereby identifying the most important variants and the order of their impact.

Depending on a specific disease, the pathogenic variants vary greatly and may increase over time. As noted, the total number of permutations of n disease-causing factors is n!.

If a rare disease has fewer than 10 pathogenic variants, the simulation model is feasible. However, with 10 or more variants, enumerating all permutations in a short time becomes impractical.

Therefore, a simulation model is designed to simulate all contributing pathogenic variants when their number exceeds 10. This model uses Algorithm 1 for Permutation Block Enumeration combined with cloud computing to store data and execute blocks in parallel on a distributed system.

Model construction: Suppose let *n* pathogenic variants, labeled in order as (1,2,3,…,n).

Step 1: Divide the pathogenic variants into *n* blocks, each receiving different inputs *s* and g to run Algorithm 1 in parallel on each Slave.

- Block 1: s = (1,2,…,n); g = (1,n,n-1,…,2)
- Block 2: s = (2,1,3,4,…,n); g = (2,n,n-1,…,3,1)
- Block 3: s = (3,1,2,4,…,n); g = (3,n,n-1,…,4,2,1)
- …
- Block n: s = (n,1,2,…,n-1); g = (n,n-1,…,1)

For example, when n=4, the 4 blocks are divided as follows:

- Block 1: s = (1,2,3,4); g = (1,4,3,2)
- Block 2: s = (2,1,3,4); g = (2,4,3,1)
- Block 3: s = (3,1,2,4); g = (3,4,2,1)
- Block 4: s = (4,1,2,3); g = (4,3,2,1)

Table 1 is divided into 4 blocks, as illustrated in Table 4 below:

Table 4. Permutation(Per) Blocks of Four variants

No	Permutation	Variants	Blocks
1	(1,2,3,4)	(GM-->CA-->TE-->DD)	**Block 1**
2	(1,2,4,3)	(GM-->CA-->DD-->TE)	
3	(1,3,2,4)	(GM-->TE-->CA-->DD)	
4	(1,3,4,2)	(GM-->TE-->DD-->CA)	
5	(1,4,2,3)	(GM-->DD-->CA-->TE)	
6	(1,4,3,2)	(GM-->DD-->TE-->CA)	
7	(2,1,3,4)	(CA-->GM-->TE-->DD)	**Block 2**
8	(2,1,4,3)	(CA-->GM-->DD-->TE)	
9	(2,3,1,4)	(CA-->TE-->GM-->DD)	

10	(2,3,4,1)	(CA-->TE-->DD-->GM)	
11	(2,4,1,3)	(CA-->DD-->GM-->TE)	
12	(2,4,3,1)	(CA-->DD-->TE-->GM)	
13	(3,1,2,4)	(TE-->GM-->CA-->DD)	**Block 3**
14	(3,1,4,2)	(TE-->GM-->DD-->CA)	
15	(3,2,1,4)	(TE-->CA-->GM-->DD)	
16	(3,2,4,1)	(TE-->CA-->DD-->GM)	
17	(3,4,1,2)	(TE-->DD-->GM-->CA)	
18	(3,4,2,1)	(TE-->DD-->CA-->GM)	
19	(4,1,2,3)	(DD-->GM-->CA-->TE)	**Block 4**
20	(4,1,3,2)	(DD-->GM-->TE-->CA)	
21	(4,2,1,3)	(DD-->CA-->GM-->TE)	
22	(4,2,3,1)	(DD-->CA-->TE-->GM)	
23	(4,3,1,2)	(DD-->TE-->GM-->CA)	
24	(4,3,2,1)	(DD-->TE-->CA-->GM)	

Step 2: After the Blocks are generated from step 1 on different processors, Each block is then mapped to the clients. On the Clients, simulation tools are used to simulate the blocks on different Slaves, and the results are subsequently transferred back to the Server**.**

Step 3: A table is created to analyze and compare the results to see which permutation has the greatest impact on the development of the disease.

Figure 2 below will show the simulation of all the pathogenic variants according to the 3 above steps.

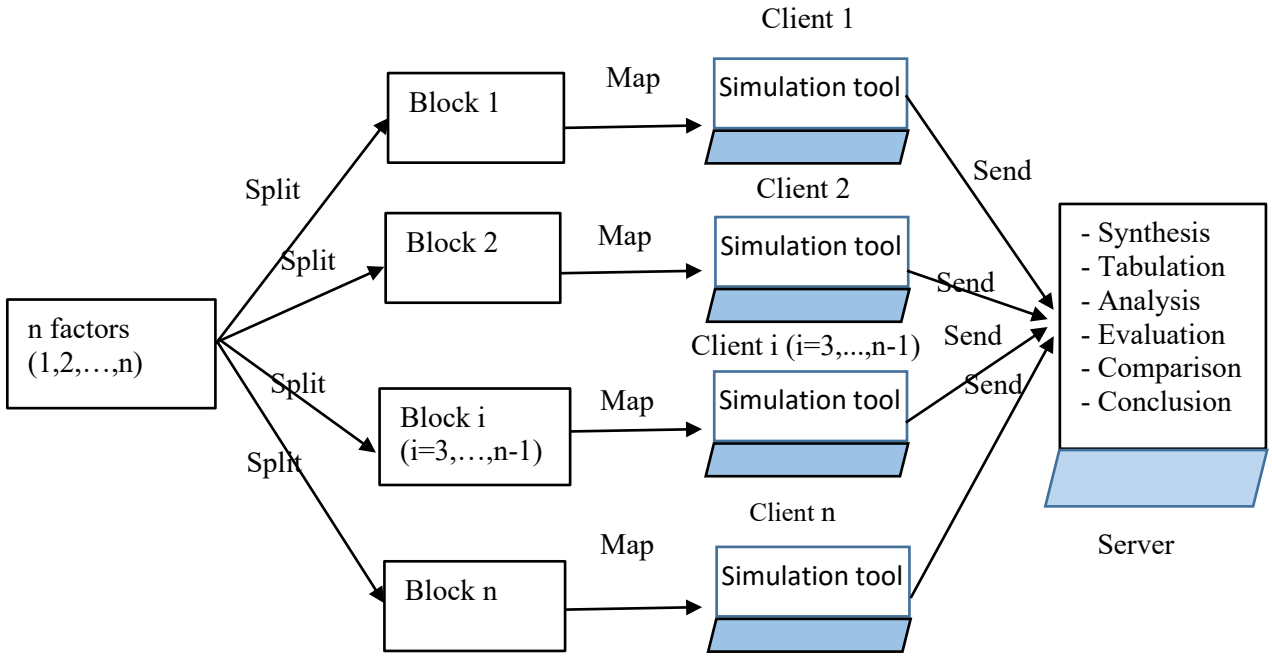

Fig 2: The simulation model of 3 steps

5. ANALYSIS AND COMPARISON

In work [12], the authors evaluated population-based disease simulation models, indicating that such models are not really reliable. In contrast, the study proposes algorithms based on combinatorial problems, in particular the enumeration of all permutations, thus ensuring absolute accuracy in the proposed model. In study [13] published in 2023, the authors applied mathematical algorithms (ML) in AI to study rare diseases, which also proves that applying algorithms in mathematics together with information technology is a new direction in the expectations and treatment of dangerous diseases. In study [14], the authors applied Molecular Anchoring Technique (Molecular Docking) and Molecular Dynamics Simulation (MD Simulation) to treat potential risk variants for Alzheimer's disease (AD). In study [15], the authors used the GeneBreaker tool to simulate the variants from which to predict and treat the toxic diseases. However, the variables continuously increased leading to slow and unreliable modeling.

In study [16], the authors simulated the genetic variants to identify previously unknown diseases. This confirmed that simulating the genetic variants is very important in anticipating the toxic diseases. In study [17], the authors raised the diverse network of human diseases and the tool that can analyze the complexity of the interaction between phenotype (phenotype) and genotype (genotype). They showed that single-gene disorders are grouped with complex diseases that increase the risk factors for eye diseases. Thus, the pathogenic variants are increasing and simulation is very essential.

6. CONCLUSION

In this paper, the theory of combinations has been applied to list all permutations to propose a rare disease prediction model. Moreover, there are three main pathogenic variant groups of rare diseases: phenotype, genetics and environment. Therefore, pathogenic variant groups of rare diseases need to be simulated to assist rare disease prevention, diagnosis and treatments. An algorithm was developed to partition the pool of pathogenic variants and distribute them across a client–server architecture for simulation. This model will ensure that all possibilities quickly to offer appropriate and timely prediction and treatments of rare diseases. In the upcoming paper, the tools will be selected to simulate this model from choosing various pathogenic variants for experimentation. It is expected that this model can solve the problem with big data on pathogenic variants of rare diseases.

7. DATA AVAILABILITY

No new datasets were generated in the course of this study. All datasets utilized in this research are available from the authors upon request.

8. ACKNOWLEDGMENTS

The authors gratefully acknowledge the experts, colleagues, and teachers who contributed to the development of this work.

9. CONFLICT OF INTEREST

The authors have no conflict of interest. All co-authors have seen and agree with the contents of the manuscript, and there is no financial interest to report.

10. REFERENCES

[1] United States. Congress. Senate. Committee on Health, Education, Labor, and Pensions, 2010. Treating rare and neglected pediatric diseases: Promoting the development of new treatments and cures: Hearing before the Committee on Health, Education, Labor, and Pensions, United States Senate, One Hundred Eleventh Congress, second session, U.S. Government Printing Office.

[2] https://soytequangninh.gov.vn/menu-second/tin-tuc-su-kien/viet-nam-hien-co-khoang-100-can-benh-hiem-voi-6-trieu-nguoi-.html

[3] Nguyen Dinh Lau, 2018. Improved computing performance for listing combinatorial algorithms using multi-processing mpi and thread library, International Journal of Computer Science & Information Technology (IJCSIT).

[4] Nguyen Dinh Lau, 2012. A New Approach to Listing Combinatorial Algorithm of Cnr, Australian Journal of Science and Technology, Melbourne Scientific Publishers.

[5] Nguyen Dinh Lau, Tran Thuy Trang and Le Thanh Tuan 2023. The MapReduce based approach to improve the shortest path computation, Journal of Mathematical and Computational Science, ISBN:1927-5307, vol 13.

[6] Lee CE, Singleton KS, Wallin M, Faundez V. Rare, 2020. Genetic Diseases: Nature's Experiments on Human Development, iScience, 22-23(5):101123.

[7] Angural A, Spolia A, Mahajan A, Verma V, Sharma A, Kumar P, Sharma S. Review, 2020. Understanding Rare Genetic Diseases in Low Resource Regions Like Jammu and Kashmir – India. Frontiers in Genetics, 11.

[8] Melissa Haendel, Nicole Vasilevsky, Deepak Unni, Cristian Bologa, Nomi Harris, Heidi Rehm, Ada Hamosh, Gareth Baynam, Tudor Groza, Julie McMurry, Hugh Dawkins, Ana Rath, Courtney Thaxon, Giovanni Bocci, Marcin P. Joachimiak, Sebastian Köhler, Peter N. Robinson, Chris Mungall, Tudor I. Oprea, 2020. *How many rare diseases are there?*, Nat Rev Drug Discov. 19(2): 77–78. doi:10.1038/d41573-019-00180-y.

[9] http://dananghospital.org.vn/tin-tuc/viet-nam-hien-co-khoang-100-can-benh-hiem-voi-6-trieu-nguoi-dang-mac.htm

[10] https://www.vinmec.com/vie/bai-viet/benh-viem-vong-mac-sac-nguyen-nhan-trieu-chung-va-cach-dieu-tri-vi

[11] https://www.ncbi.nlm.nih.gov/books/NBK208609/

[12] Jacek A Kopec, Philippe Finès, Douglas G Manuel, David L Buckeridge William M Flanagan, Jillian Oderkirk, Michal Abrahamowicz, Samuel Harper, Behnam Sharif, Anya Okhmatovskaia, Eric C Sayre, M Mushfiqur Rahman, Michael C Wolfson, 2010. Validation of population-based disease simulation models: a review of concepts and methods, BMC Public Health.

[13] Anna Visibelli, Bianca Roncaglia, Ottavia Spiga and Annalisa Santucci, 2023. The Impact of Artificial Intelligence in the Odyssey of Rare Diseases, Biomedicines.

[14] Weixue Xiong, Jiahui Cai, Ruijia Li, Canhong Wen, Haizhu Tan and on behalf of the Alzheimer's Disease Neuroimaging Initiative (ADNI) Database, 2022. Rare Variant Analysis and Molecular Dynamics Simulation in Alzheimer's Disease Identifies Exonic Variants in FLG, Genes.

[15] Phillip A. Richmond, Tamar V. Av-Shalom, Oriol Fornes, Bhavi Modi, Alison M. Elliott, Wyeth W. Wasserman, GeneBreaker, 2021. Variant simulation to improve the diagnosis of Mendelian rare genetic diseases, Human Mutation, pp 346–358.

[16] Emily Alsentzer, Samuel G. Finlayson, Michelle M. Li, Undiagnosed Diseases Network, Shilpa N. Kobren & Isaac S. Kohane, 2023. Simulation of undiagnosed patients with novel genetic conditions, Nature communications.

[17] Arda Halu, Manlio De Domenico, Alex Arenas and Amitabh Sharma, 2019 The multiplex network of human diseases, NPJ