Sequence similarity alignment algorithm in Bioinformatics: Techniques and Challenges

Yuren Liu[1], Yijun Yan[1], Jinchang Ren[1*] and Stephen Marshall[1]

[1] Dept. of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, UK

***Corresponding author: Jinchang.Ren@strath.ac.uk**

Abstract. Sequence similarity alignment is a basic information processing method in bioinformatics. It is very important for discovering the information of function, structure and evolution in biological sequences. The main idea is to use a specific mathematical model or algorithm to find the maximum matching base or residual number between two or more sequences. The results of alignment reflect to what extent the algorithm reflects the similarity relationship between sequences and their biological characteristics. Therefore, the simple and effective algorithm of sequence similarity alignment in bioinformatics has always been a concern of biologists. This paper reviews some widely used sequence alignment algorithms including double-sequence alignment and multi-sequence alignment, simultaneously, introduces a method to call genetic variants from next-generation gene sequence data.

Keywords: Bioinformatics; Longest common subsequence (LCS); Deoxyribonucleic acid (DNA); sequence alignment

1 Introduction

With the successful implementation of the Human Genome Project and the rapid development of information technology, the data volume of the three international nucleic acid sequence databases (Genebank, EMBL and DDBJ) has increased exponentially. Biologists, mathematicians and computer scientists are all facing the same and severe problem, i.e. how to use and express these data to analyse and explain the potential relationship between gene sequences, and to find out the beneficial information for human beings. In order to meet this challenge, bioinformatics emerged as the times require, and has increasingly become one of the core fields of natural science in the 21st century.

In the research of biology, a common method is to obtain useful information through comparative analysis. We analyze the similarities and differences of sequences at the level of nucleic acid and amino acid in order to infer their structure, function and evolutionary relationship. The most commonly used method of comparison is sequence alignment, which provides a very clear map of the relationship between residues of two or more sequences.

One of the purposes of sequence alignment is to enable people to judge whether there is enough similarity between two sequences, so as to determine whether there is homology between them. Therefore, sequence alignment is of great significance and practical value in bioinformatics. At present, many classical alignment algorithms have been proposed internationally, and many sequence alignment software have been developed. However, for the same set of sequences, different software uses different sequence alignment algorithms, and their operation speed and alignment results are quite different. Some software takes the comparison results into account and runs for a long time, while others do the opposite. Normally, they can't be achieved both. Therefore, the research of sequence alignment algorithm still needs to be deepened.

The main task of sequence alignment is to use algorithms to compare DNA sequences and discover similarities and differences between them. In bioinformatics, the similarity of DNA sequence or protein sequence is mainly manifested in three aspects: sequence, structure and function. Usually, sequence determines result and structure determines function. Therefore, sequence similarity research is mainly manifested in two aspects. On the one hand, sequence similarity analysis is used to discover the results and functions of sequences, and on the other hand, sequence similarity is used to analyze the evolutionary relationship between sequences.

The aim of this paper is to review a broad selection of sequence alignment algorithms used in bioinformatics with a particular focus on the LCS problem. It also discusses the principle of some classic sequence alignment algorithms. Furthermore, we also summarize the feature of double-sequence and multi-sequence alignment and introduce a method dealing with the compilation issue of many DNA fragmented reads.

The rest of this paper is organized as follows. Section 2 provides an overview of some basic concept of bioinformatics. Section 3 introduces the main frame of dynamic sequence alignment algorithm. Section 4 presents some classic sequence alignment algorithms. Section 5 discusses the main challenges and the future work on gene sequence alignment.

2 Brief of bioinformatics

The main task of bioinformatics is to analyze, process and study various biological information contained in DNA sequence data. Bioinformatics includes sequence comparison, protein structure comparison and prediction, gene recognition, molecular evolution and comparative genomics, sequence overlap group assembly, structure-based drug design and so on [1].

2.1 Nucleic acid and Protein

Nucleic acid is a one-dimensional polymer chain, which contains four monomers, each of which is called nucleotide. Nucleic acids carry genetic information, which is mainly expressed in the sequence of nucleotides. According to the different types of nucleotides, nucleic acids are divided DNA and ribonucleic acid (RNA). Nucleotides consist of phosphoric acid, deoxyribose or ribose and bases. The bases that make up nucleotides are divided into purine and pyrimidine. The former mainly refers to adenine (A) and guanine (G), both of which are contained in DNA and RNA. The latter mainly refers to cytosine (C), thymine (T) and uracil (U). Cytosine exists in DNA and RNA, thymine only exists in DNA, and uracil only exists in RNA. Among them, DNA is the main material basis for storing, replicating and transmitting genetic information, and RNA plays an important role in protein synthesis.

2.2 Variation

Variation refers to the alteration of some bases of DNA sequence in the course of biological evolution. Variations can be classified into three categories:

1. Substitution: Substitution of one base in a sequence by another in the course of biological evolution.
2. Insert or delete: Adding or deleting one or more bases in the course of biological evolution.
3. Rearrangement: Some segments of a DNA or protein sequence undergo a change in the sequence of links during synthesis.

Variation plays a very important role in the actual research process. Variation not only causes genetic variation and disease, but also species diversity.

3 Sequence Alignment

3.1 Overview of Sequence Alignment

In scientific research, comparison is one of the most common methods. In order to find the similarities and differences between objects or to discover the possible characteristics of objects, we usually use the method of comparison. In bioinformatics, comparisons are the alignment of multiple and similar sequences. Sequence alignment originated from the theory of evolution. If the two sequences are very similar, it can be inferred that the two sequences may have the same ancestors, which evolved from the compilation process of their ancestors through different gene substitution, addition, deletion and rearrangement. In addition, the structure and function of a given protein sequence can also be inferred by sequence alignment. Therefore, sequence alignment can be applied to secondary structure prediction, functional domain recognition of proteins and gene recognition.

Sequence alignment is to use a specific mathematical model or algorithm to find out the maximum matching base number between sequences, that is, insert a space '-' in two or more string sequences to achieve the maximum number of matched characters. For example, Fig.1 shows the sequence alignment of two sequences 'AGCTTCGACCA' and 'AGCTTCGCCA'. **Fig. 1**(a) contains 8 same bases and **Fig. 1**(b) contains 10 same bases.

```
A  G  C  T  T  C  G  A  C  C  A      A  G  C  T  T  C  G  A  C  C  A
|  |  |  |  |  |  |  |  |  |         |  |  |  |  |  |  |  |  |  |  |
A  G  C  T  T  C  G  C  C  A         A  G  C  T  T  C  G  -  C  C  A
           (a)                                    (b)
```

Fig. 1. Matching before the insertion of space '-' (a) and after the insertion of space '-'(b).

Compared with the method of not inserting spaces, it increases the number of matches. It can be seen from this that inserting vacancies is very necessary, and the process also reflects the process of biological evolution. The realization of sequence alignment generally depends on a mathematical model. Different mathematical models may reflect different characteristics of sequence structure, function and evolutionary relationship. It is difficult to judge whether a mathematical model is good or bad, or whether a mathematical model is right or wrong. It only reflects the biological characteristics of a sequence from a certain point of view.

3.2 Multi-sequence Alignment

To sum up, we can use a quintuple to describe the problem of multiple sequence alignment. See Eq 3.1:

$$\mathrm{MSA} = (\Sigma, \mathrm{S}, \mathrm{A}, \mathrm{O}, \mathrm{F}) \tag{1}$$

where Σ represents a set of symbols for multiple sequence alignments with a value of $\Sigma 1 \cup \{-\}$; $\Sigma 1$ represents finite set of symbols, while in protein sequence alignment $\Sigma 1 = \{\mathrm{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}\}$, and while in DNA sequence alignment, $\Sigma 1 = \{\mathrm{A, T, C, G}\}$, - means a space which will be inserted during the process of alignment.

S means the sequence set to be aligned. The unmodulated sequence of protein sequence alignment is composed of amino acids. DNA sequence alignment is that each sequence is composed of bases and the sequence length is different. $S = \{S_i | \mathrm{i} = 1,2, \dots, \mathrm{m}\}$,

$S_i = (C_{ij} | j = 1,2, \dots, l_i)$, where m equals to the number of sequences, C_{ij} is the j^{th} base in sequence S_i, l_i means the length of the i^{th} sequence.

A represents the result matrix, $A = (a_{ij})_{m \times n}, a_{ij} \in \Sigma$. In the result matrix, line I represents the i^{th} sequence, and each j lists of the matrix represents the result of the comparison of the j^{th} base. The base sequence in the sequence cannot be changed before and after alignment.

O is a set of comparison operations, $O = \{\text{insert\_space},\ \text{delete\_space}\}$, which is the operation of insertion and deletion of the gap '-'.

F is the algorithm of alignment in order to figure out the specific position of the insertion and deletion.

3.3 Vacancy Penalty

In the process of sequence alignment, in order to make the results of sequence alignment more in line with certain expectations, the insertion or deletion of sequences is compensated by introducing vacancies. However, we should not introduce vacancies indefinitely, otherwise the results will lack biological significance. In order to limit the insertion of spaces, the usual method is to deduct the total score by inserting spaces. The deduction score is a penalty score, which restricts the insertion of vacancies into the penalty score of vacancies. Therefore, when the result of sequence alignment scored, the total score of matching residues between two sequences and the sum of space penalty scores were obtained [2].

Suppose S_1 and S_2 are used to represent the sequence to be aligned, S_{10} and S_{20} are used to represent the result of alignment, and L is used to represent the length of alignment. Generally, there are three kinds of space penalty rules.

3.3.1 Vacancy Penalty

The simplest penalty rule is the vacancy penalty score. When a vacancy is inserted into a sequence, a fixed penalty score Wg is given. For the whole sequence, the total blank penalty score is equal to the inserted blank Rg multiplied by the penalty score Wg for each vacancy [3].

Vacancy penalty points do not add extra running time, so it is the simplest penalty rules. However, the mutation frequencies of bases in gene sequences are different according to different loci in the actual evolution process, but the penalty scores of each locus in the vacancy penalty score are the same, which is different from its actual biological significance.

3.3.2 Constant Vacancy Penalty

This penalty rule is not for every space, but for every vacancy. Here the connected space is called a vacancy. The penalty score is based on the vacancy inserted in the sequence, and the penalty score of every inserting vacancy is Wg [3]. The specific operation is as follows:

Assuming whether matched or not, the score value is expressed by σ, we have $\forall \mathrm{x}, \sigma(\mathrm{x}, -) = \sigma(-, x) = 0$;

The representation of alignment score becomes:

$$\sum_{i=1}^{L} \sigma(S_{10}[i], S_{20}[i]) + W_g \times gaps \qquad (2)$$

where *gaps* represents the number of vacancy.

Constant space penalty can avoid the defect of space length penalty, but when too many connected spaces are inserted, this penalty rule cannot be limited, which may lead to the splitting of sequence segments by connected spaces. This requires a penalty score rule which is closely related to the length of the space. Its penalty score does not only depend on the length of the space, but also does not ignore the length of the space.

3.3.3 Affine Vacancy Penalty

The rule divides the penalty of vacancy into two parts [3]: open vacancy penalty and extended vacancy penalty. If q is the length of the vacancy, W_g is the penalty score of the open vacancy, W_s is the penalty score of the extended vacancy and W is the total penalty score, then: $\mathrm{W} = W_g + \mathrm{q}\ \times W_s$, so we can have the formula calculating the alignment score:

$$\sum_{i=1}^{L} \sigma(S_{10}[i], S_{20}[i]) + W_g \times gaps + W_s \times spaces \tag{3}$$

where *gaps* is the number of space and spaces represents the number of vacancy.

In practical biology research, the probability of inserting and deleting multiple vacancies connected by a vacancy coin is small, so the affine vacancy penalty score has more biological significance.

3.4 Scoring Matrix

In sequence alignment, a matrix is usually used to record the score of each variation, which becomes the score matrix. The results of sequence alignment will be different if different scoring matrices are chosen. The simplest scoring matrix is a single matrix, also known as a sparse matrix. Using this matrix, we only need to detect whether the bases of corresponding sites between sequences are identical, and the scores of the same bases are 1, and the differences are 0. This matrix, which only considers the identity of bases, has great limitations.

In order to better reflect the biological characteristics, we need to design a more optimized scoring matrix. PAM (point accepted mutation) [4] is the first widely used optimal matrix. A PAM indicates that 1% of the amino acids have changed, that is, the evolutionary unit of variation. Generally, sequences with high similarity use lower PAM matrix, while sequences with low similarity use higher PAM matrix.

Besides PAM matrix, BLOSUM (blocks substitution matrix) [4] matrix is also widely used. BLOSUM matrices also use numbering to distinguish different BLOSUM matrices, where numbering is mainly used to distinguish the similarity of sequences. For example, BLOSUM62 matrix is generally used to align at least 62% of the same proportion of sequences. So the use of BLOSUM matrix is exactly the opposite to that of PAM matrix.

4 Classic Alignment Algorithms

At present, many sequence alignment algorithms are based on dynamic programming algorithm, considering different improvements in computing speed and storage space in Chengdu. Sequence alignment algorithms have several different classification methods. According to the number of alignment sequences, sequence alignment can be divided into double sequence alignment and multiple sequence alignment. According to the range of sequence alignment, sequence alignment can be divided into global sequence alignment and local sequence alignment.

4.1 Global Sequence Alignment and Local Sequence Alignment

Local sequence alignment considers the local similarity of sequences, which is a forehead method to find partial similarity regions of sequences. Local sequence alignment is mainly applied to protein sequence alignment, which is more sensitive and biologically significant than complete sequence alignment. Global sequence alignment is the whole sequence, which considers the similarity of sequences from the global scope. Global sequence alignment is mainly used to predict the homology between sequences and the structure and function of proteins.

4.2 Double-sequence Alignment

Double sequence alignment is to find the maximum similarity match between two DNA or protein sequences. The search process is based on some algorithm or model. Multiple sequence alignment and sequence database search are based on double sequence alignment. At present, the most classical double sequence alignment algorithms are lattice graph method and dynamic programming algorithm.

4.2.1 Lattice Graph Method

The simplest double sequence alignment algorithm is the lattice graph method. In this method, the sequence to be aligned is placed on a two-dimensional plane, a sequence is placed horizontally on the top of the plane, and a sequence is placed vertically on the left of the plane. A point is marked at the intersection of any two identical bases of the two sequences. Finally, linking the points parallel to the diagonal line constitutes the result of two sequence alignments [5].

The lattice graph method can visually display the insertion and deletion of sequences, and all matched base sequences between two sequences can be visually reflected by lattice graph. However, since the sequence length is counted in thousands, it is unrealistic to use all the lattice computing programs to calculate the real alignment sequence, so other alignment methods will be more used to achieve.

4.2.2 Dynamic Programming Algorithm

Dynamic programming algorithm was first proposed by Needleman and Wunsch and has been widely used and improved. It has gradually become one of the most important theoretical foundations in computational biology. The most classical dynamic programming algorithms are Needleman-Wunsch algorithm [6] and Smith-Waterman algorithm [7]. All global alignment algorithms are based on NW algorithm, while SW algorithm is improved based on NW algorithm, mainly applied to local sequence alignment. The following is a brief introduction to the dynamic programming algorithm.

Given sequence s_1 and s_2, the length of which are m and n correspondingly. $s_1[1 \dots i]$ and $s_2[1 \dots j] (1 \le i \le m, 1 \le j \le n)$ represent prefix subsequences separately of s_1 and s_2. And the alignment result of s_1 and s_2 contains that of $s_1[1 \dots i]$ and $s_2[1 \dots j]$, which is a recursive relationship.

From this relationship, it can be seen that the optimal solution to its subsequence is the premise of solving the global alignment of two sequences. Through this recursive relation, the optimal value of the whole sequence can be obtained. Then, the optimal alignment result of the sequence is obtained by backtracking the path of the obtained optimal value.

The basic step of dynamic programming algorithm is to use a binary matrix to store the similar scores of two sequences, and then retrieve the optimal alignment of the sequences according to the scores in the matrix. Assume that the sequence s_1 and s_2 are compared by using dynamic programming algorithm, their lengths are m and n, respectively. Firstly, we need to construct a two-dimensional matrix with the size of $(m+1) \times (n+1)$. The element $M[i,j] (0 \le i \le m, 0 \le j \le n)$ in the matrix represents the highest alignment score of its prefix subsequence $s_1[1 \ldots i]$ and $s_2[1 \ldots j]$. The ratio of prefix subsequence a to vacancy '-' is expressed in both row 0 and column 0 of the matrix. Therefore, the initial values of the elements in the matrix are:

$$M[0,0] = 0 \tag{4}$$

$$M[i,0] = \sum_{k=1}^{i} \sigma(s_1[i], -)\ (1 \le i \le m) \tag{5}$$

$$M[0,j] = \sum_{k=1}^{j} \sigma(-, s_2[j])\ (1 \le i \le n) \tag{6}$$

By analyzing the prefix subsequence $s_1[1 \ldots i]$ and $s_2[1 \ldots j]$, there may be three cases to get the optimal score $M[i,j]$:

1. The sum of the alignment score between $s_1[i]$ and $s_2[j]$ and the score between the subsequence $s_1[1 \ldots i-1]$ and $s_2[1 \ldots j-1]$ which is $M[i-1, j-1]$;
2. The sum of the alignment score between $s_1[i]$ and a space '-' and the score between the subsequence $s_1[1 \ldots i-1]$ and $s_2[1 \ldots j]$ which is $M[i-1, j]$;
3. The sum of the alignment score between $s_2[j]$ and a space '-' and the score between the subsequence $s_1[1 \ldots i]$ and $s_2[1 \ldots j-1]$ which is $M[i, j-1]$. And it's shown below (**Fig. 2**):

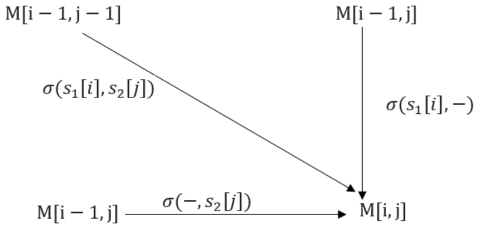

Fig. 2. Source of Matrix Elements

Thus, the recursive relation is obtained as:

$$M[i,j] = max\begin{cases} M[i-1, j-1] + \sigma(s_1[i], s_2[j]) \\ \quad M[i-1, j] + \sigma(s_1[i], -) \\ \quad M[i, j-1] + \sigma(-, s_2[j]) \end{cases} (1 \le i \le m, 1 \le j \le n) \tag{7}$$

Using the upper formula, the values of each element in the matrix are calculated in order from left to right and from top to bottom. When the position of the last column in the last row, element $\mathrm{M}[\mathrm{m}, \mathrm{n}]$, is calculated, the best alignment score of two sequences s1 and s2 is obtained.

After obtaining the optimal score, we use the backtracking method to construct the comparison record. That is, starting from the position of the optimal alignment score, i.e. the position of the last column in the last row of the matrix, we retrospect along the path to get the value until reaching column 0 of row 0. At this point, the sequence corresponding to the intersection points in the backtracking process is the alignment result.

When using dynamic programming algorithm for sequence alignment, it is necessary to calculate each element in a two-dimensional array of $(m + 1) \times (n + 1)$ size. Its time complexity is *O(mn)* and space complexity is *O(mn)*. The backtracking process is from the lower right of the array to the upper left of the array, passing through *(m+n)* elements, so its time complexity is *O(m+n)*, and there is no additional storage overhead. Therefore, the basic dynamic programming algorithm takes a lot of time and space to solve the problem of double sequence alignment, so people put forward various improved algorithms.

4.2.3 Multi-sequence Alignment

The generalization of double sequence alignment in multi-sequence alignment is to extend the alignment problem from two sequences to multiple sequences. Therefore, the method to solve the double sequence problem is also applicable to multiple sequence alignment, but the number of alignments has increased, which makes the problem more complex. Murata has successfully applied dynamic programming algorithm to the alignment of three sequences [8], but because the alignment time is long and the space required is large, it is almost impossible to extend it to more than three sequences alignment. Generally, the dynamic programming algorithm is seldom used in multi-sequence alignment. Basically, heuristic algorithm, random algorithm and partition algorithm are used. Among them, there are many kinds of heuristic algorithms, such as star alignment algorithm, progressive alignment algorithm, iterative thinning method and so on.

4.2.3.1 Star Alignment Algorithm

Star alignment algorithm is a fast heuristic method for solving multiple sequence alignment problems. It needs to find a central sequence, and the result of alignment is established by comparing the central sequence with other sequences. Star alignment algorithm follows a rule, that is, in the alignment process, the space must be added to the central sequence continuously so that the central sequence and alignment sequence can reach the maximum number of matches. Spaces added to the central sequence cannot be removed, and always remain in the central sequence, knowing that the central sequence and the ordered sequence are aligned.

4.2.3.2 Progressive Alignment Algorithm

Another simple and effective heuristic algorithm is the progressive alignment algorithm. The basic idea of progressive alignment algorithm is to use dynamic programming algorithm to iteratively align two sequences. That is to say, the two sequences are aligned first, and then the new sequences are added, until all the sequences are added. However, different addition order may lead to different comparison results. Therefore, the key of progressive alignment algorithm is how to determine the sequence of alignment. Generally, the alignment begins with the two most similar sequences, and then proceeds to the far alignment to complete all the sequences.

Progressive alignment algorithm mainly consists of three steps:

1. Computation of the distance matrix;
2. Construction of guidance tree;
3. Alignment of the sequences according to the constructed guide tree.

Nowadays, the most widely used progressive comparison procedure is ClustalW. It gives a set of schemes of dynamic selection of comparison parameters, which mainly solves the problem of parameter selection in the process of comparison. Usually, the scoring matrix and reflective blank penalty score are used to solve the selection problem of comparison parameters, and it is hoped that the effective parameters can be set to achieve the desired results.

5 Challenges and Future Directions

Classical similarity analysis of biological sequences is achieved by comparing biological sequences, and sequence alignment is achieved by comparing two or more nucleic acid sequences or protein sequences. By comparing the similarity between the position sequence and the known sequence, we can get their homology to predict the function of the unknown sequence.

At present, the main methods of sequence alignment are: double sequence alignment algorithm, multi-sequence alignment algorithm, knowledge discovery-based alignment algorithm and graphic representation-based alignment algorithm. The first two are based on the idea of dynamic programming, which is realized by mature algorithms and software, but the calculation is quite complicated. However, the method based on knowledge discovery is not yet mature. With the continuous improvement of graphical representation and related theory of biological sequence, there will be a great space for the development of alignment algorithm based on graphical representation.

Therefore, based on the above research, our future work is to improve the current sequence alignment algorithm and try to develop a bio-sequence analysis software based on graphical identification on small-indel variant calling [9], non-human variant calling [10] and high accurate SNP calling [11]. As an interdisciplinary subject of computer science and molecular biology, bioinformatics has become an indispensable and advantageous research tool in genome research [12]. And some advanced feature extraction algorithms [13] shall be adopted into a deep learning alignment method, hopefully. It is believed that in the future, improved alignment methods can be used to find genes that may be of research value, or to find out the other side of known genes that have not been discovered.

References

1. C. Zhang, "Current Situation and Prospect of Bioinformatics," *World Science and Technology Research and Development,* vol. 22, pp. 17-20, 2000.
2. D. J. Parry-Smith. T. K. Attwood, *Introduction to bioinformatics,* pp. 168-196, 1999.
3. Z. Xiao, “A Multiple Alignment Approach for DNA Sequence Based on the Maximum Weighed Path Algorithms," *Xi'an University of Electronic Science and Technology,* pp. 8-9, 2006.
4. W. R. Pearson, "Selecting the Right Similarity ‐ Scoring Matrix," *Current Protocols in Bioinformatics,* vol. 43, pp. 1-9, 2013.
5. A. J. Gibbs, G. A. McIntyre, “The Diagram a Method for Comparing Sequences its Use with Amino and Nucleotide Sequences," *Eur J Biochem,* vol. 16, pp. 1-11, 1970.
6. C. D. Wunsh, S. B. Needleman, "A General Method Applicable to the Search for Similarities in the Amino Acid Sequences of Two Proteins," *Journal of Molecular Biology,* vol. 48, pp. 443-453, 1970.
7. M. S. Waterman, T. F. Smith, "Identification of Common Molecular Subsequences," *J Mol Biol,* vol. 147, pp. 195-197, 1981.
8. J. S. Richardson, M. Murata, J. L. Sussman, "Simultaneous Comparison of Three Protein Sequences," *Proc Natl Acad Sci,* vol. 93, pp. 3073-3077, 1985.
9. P. C. Chang, R. Poplin, D. Alexander, "A Universal SNP and Small-indel Variant Caller Using Deep Neural Networks," *Nature Biotechnology,* vol. 36, pp. 983-987, 2018.
10. C. McLean, T. Yun, P. C. Chang, "Improved Non-human Variant Calling Using Specie-specific DeepVariant Models," *Google AI DeepVariant Blog,* 2018.
11. P. C. Chang, A. Kolesnikov, "Highly Accurate SNP and Indel Calling on PacBio CCS woth DeepVariant," *Google AI DeepVariant Blog,* 2019.
12. H. Li, G. Baid, P. C. Chang, "Using Nucleus and TenserFlow for DNA Sequencing Error Correction," *Google AI DeepVariant Blog,* 2019.
13. J. Ren, J. Zabalza, et al, "Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging," *Neurocomputing,* vol. 185, pp. 1-10, 2016.