Appl. Math. J. Chinese Univ.
2021, 36(1): 114-127

Analysis method and algorithm design of biological sequence problem based on generalized k-mer vector

LIU Wen-li[1,2] WU Qing-biao[1,*]

Abstract. K-mer can be used for the description of biological sequences and k-mer distribution is a tool for solving sequences analysis problems in bioinformatics. We can use k-mer vector as a representation method of the k-mer distribution of the biological sequence. Problems, such as similarity calculations or sequence assembly, can be described in the k-mer vector space. It helps us to identify new features of an old sequence-based problem in bioinformatics and develop new algorithms using the concepts and methods from linear space theory. In this study, we defined the k-mer vector space for the generalized biological sequences. The meaning of corresponding vector operations is explained in the biological context. We presented the vector/matrix form of several widely seen sequence-based problems, including read quantification, sequence assembly, and pattern detection problem. Its advantages and disadvantages are discussed. Also, we implement a tool for the sequence assembly problem based on the concepts of k-mer vector methods. It shows the practicability and convenience of this algorithm design strategy.

§1 Introduction

Sequence analysis is a fundamental problem in the bioinformatics area [1, 2, 3]. Developing algorithms for the sequence analysis problems is a hot topic due to the rapid updating speed of sequencing technology [4].

Except for the most common string representation of biological sequences, other sequence forms also exist. For example, the graphical representation of sequences with geometric features [5, 6]. As for the vector and matrix representation methods, researchers have given us some excellent results. In the work of Ding et al., the frequency of short k-words (length$\leq$7) was used to give a simple feature representation vector for DNA sequences [7]. In the work of Wren et al., they built a matrix by extracting s set of position-dependent features to represent a DNA

Received: 2020-02-05. Revised: 2020-04-03.
MR Subject Classification: 92B05.
Keywords: vector space, biological sequence, k-mer, algorithm design, analysis method.
Digital Object Identifier(DOI): https://doi.org/10.1007/s11766-021-4033-x.
Supported by the National Natural Science Foundation of China(11771393,11632015) and the Natural Science Foundation of Zhejiang Province, China (LZ14A010002).

[*] Corresponding author.

sequence [8]. Wen et al. used a Boolean logic value for the existence of each k-mer to build a matrix [9]. There are also works on the protein sequences as well [10]. However, these methods are incompatible with each other which makes the vector method of biological sequences limited when being extended to new problems. A more general definition is required.

Biological sequence data and sequence-based problems have some features in common. The large size of the data set has increased the problem complexity. Massive information redundancy exists in many cases while some are not necessary. The sequence length, read depth of coverage and the sequencing accuracy are some common variables seen within the analysis problems. And the calculation of sequence similarity is a key node in the network of the sequence-based problems. These bioinformatics features must be linked to the concepts in linear algebra and matrix theory when constructing the vector spaces. Then a myriad of methods and research findings from linear algebra and computational mathematics can be applied to the biological sequence-based problems in their matrix form.

K-mer is a frequently-used concept in solving sequence-based problems in bioinformatics. K-mers are short subsequences of a biology sequence. Using short subsequences can reduce the complexity of a problem by only considering part of the information. Researchers have given us a lot of analysis tools using k-mer as the key element in algorithm design, such as Kallisto [11] and Sailfish [12] while the k-mer length ranges from a few bps (base pair) to dozens of bps in different research work. Especially the sequencing reads quantification tool, kallisto. It has a highly efficient alignment-free algorithm by using the counts of k-mer, showing the ability of k-mer in solving a sequence-based problem. Also, there are some efficient methods for counting k-mers ([13] , KMC2 [14] and Kmer-SSR [15]) and discussions on the suitable k-mer length [16, 17]. These conditions make the k-mer a priority option for building solutions to sequence-based problems.

In this study, we defined the k-mer vector for the DNA sequence and sequence set. We constructed a vector space of the sequence vectors and defined the operations over it. Then we presented three basic sequence-based bioinformatics problems in their matrix form, including read quantification, sequence assembly problem, and sequence pattern detection. Moreover, we designed an algorithm and implemented a tool for the sequence assembly problem using the k-mer vector method. It is an instance of applying the concepts of sequence vector space and vector operations in solving a practical problem.

§2 The k-mer vector space for biological sequence

Biological sequences are sequences with limited alphabet size. In this work, we are mainly dealing with the DNA sequence ({A, G, C, T}) as a simplified model for discussion.

2.1 k-mer vector generated from a biological sequence

Let s be a DNA sequence of length l_s. So s is a string of length l_s over the alphabet $\Sigma = \{A, G, C, T\}$. Let the k-mer set of s with defined k-mer length k be $K_s = \{k_i | i \in \{1, 2, \cdots, l_s - k + 1\}\}$, where k_i is the subsequence of s starting from location i. We have

$|K_s| = l_s - k + 1$. Then a set of sequences can be denoted as $S = \{s_i | i \in \{1, 2, 3, \cdots, n_S\}\}$, where s_i stands for one sequence and n_S denotes the number of sequences in set S.

The total number of all possible k-mers with defined k-mer length k is 4^k. If we sort all the k-mers with the dictionary order, which is $(k^1_{mer}, k^2_{mer}, \cdots, k^{4^k}_{mer})$, a vector can be generated from sequence s by $\vec{v}_s = (v_1, v_2, \cdots, v_{4^k})$, where $v_i = 1$ if $k^i_{mer} \in K_s$, $v_i = 0$ if $k^i_{mer} \notin K_s$. We call $\vec{v}_s$ the k-mer vector of s. Thus a set S with n_S sequences can be denoted by a matrix:

$$\begin{bmatrix} \vec{v}_{s_1} \\ \vec{v}_{s_2} \\ \cdots \\ \vec{v}_{s_{n_S}} \end{bmatrix} = \begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,4^k} \\ v_{2,1} & v_{2,2} & \cdots & v_{2,4^k} \\ \cdots & \cdots & \ddots & \cdots \\ v_{n_S,1} & v_{n_S,2} & \cdots & v_{n_S,4^k} \end{bmatrix}, v_{i,j} = \begin{cases} 1, & k^j_{mer} \in K_{s_i} \\ 0, & k^j_{mer} \notin K_{s_i} \end{cases}. \tag{1}$$

In many sequence-based bioinformatics problems, there is a concept of read depth of coverage over each position along the sequence. When we introduce depth into the definition of k-mer vector, we have $\vec{v}_s = (v_1, v_2, \cdots, v_{4^k})$, where $v_i = d_i$, d_i is the depth on position i along sequence s and $d_i \in N^+$. Moreover, different sequencing technologies have different sequencing accuracy. The actual depth d'_i on position i should be calculated as $d_i * p$ where $*$ is the multiplication of real numbers and the sequencing accuracy is denoted by p. It shows that the depth value can be adjusted from a positive integer to a positive real number. Thus $\vec{v}_s = (v_1, v_2, \cdots, v_{4^k})$, where $v_i = d'_i$, d'_i is the depth of coverage on position i along sequence s and $d'_i \in R$. A negative value in the vector can be similarly given a practical meaning as well. It can stand for the unsupported degree of a position. Thus the definition of the matrix for sequence set S can be broadened to

$$\begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,4^k} \\ v_{2,1} & v_{2,2} & \cdots & v_{2,4^k} \\ \cdots & \cdots & \ddots & \cdots \\ v_{n_S,1} & v_{n_S,2} & \cdots & v_{n_S,4^k} \end{bmatrix}, v_{i,j} = \begin{cases} d_i, & k^j_{mer} \in K_{s_i}, d_i \in R \\ 0, & k^j_{mer} \notin K_{s_i} \end{cases}. \tag{2}$$

Using the above definition, any sequence can be transformed into a k-mer vector. A k-mer vector can represent one read in a sequence set S or the whole set S by $[1, 1, \cdots, 1]_{1*n_S} S_{n_S*4^k}$. We can treat a single sequence as a special case of sequence set which has size 1. So the k-mer vector is describing the k-mer set of a sequence set. Two identical sequences will have the same k-mer vector while the reverse does not hold. But in practice, the possibility that two different sequences have the same k-mer vector is small.

2.2 Generalized k-mer vector and the properties of k-mer vector space

A k-mer vector is defined as $\vec{v_s} = (v_1, v_2, \cdots, v_{4^k})$, where $v_i \in R$. It stands for a generalized biological sequence s. The concept of sequences has been extended here: not limited to continuous strings. So the sequence s is represented by its k-mer set which is described by the k-mer vector. From this definition of k-mer vector, we can generate a k-mer vector space.

Let V be the set of all possible k-mer vectors which is nonempty and R the real number set. We use the addition and multiplication between real numbers to define the operation on V. Define the addition operation between $\vec{v} = (v_1, v_2, \cdots, v_{4^k}) \in V$ and $\vec{v}' = (v_1', v_2', \cdots, v_{4^k}') \in V$ as $\vec{v} + \vec{v}' = (v_1 + v_1', v_2 + v_2', \cdots, v_{4^k} + v_{4^k}')$. It can be seen as the merging of two sequence sets which result in a new set. Define the multiple operation between $\vec{v} = (v_1, v_2, \cdots, v_{4^k}) \in V$ and $k \in R$ as $k\vec{v} = k(v_1, v_2, \cdots, v_{4^k}) = (kv_1, kv_2, \cdots, kv_{4^k})$. The multiplication is the scaling of the k-mer vector for a sequence set. We have

Theorem 1. *V is a linear space on R.*

Proof of Theorem 1. For the defined vector space V and field R with two operations, the following conditions are satisfied:

a.$\forall \vec{v}, \vec{v}' \in V, \vec{v}+\vec{v}' = (v_1+v_1', v_2+v_2', \cdots, v_{4^k}+v_{4^k}')$. Since $v_i+v_i' \in R, i \in 1, 2, \cdots, 4^k, \vec{v}+\vec{v}' \in V$;

b.$\forall \vec{v} \in V, k \in R, k\vec{v} = (k * v_1, k * v_2, \cdots, k * v_{4^k})$. Since $k * v_i \in R, i \in 1, 2, \cdots, 4^k, k\vec{v} \in V$.

So V is closed for the linear operations. □

Using the linear operations over the k-mer vector space V, we can merge the sequencing data from different samples into one, get the difference between two sequence sets, or do the normalization for multiple data. Now introduce the definition of the inner product on the k-mer space. The set of all possible k-mer vectors, V, is a linear space over R. $\forall \vec{v}, \vec{v}' \in V$, define the inner product of $\vec{v}$ and $\vec{v}'$ as $\langle \vec{v}, \vec{v}' \rangle = \sum_{i=1}^{4^k} v_i * v_i'$. Obviously we have

Theorem 2. *V is a Euclidean space with an inner product $\langle \vec{v}, \vec{v}' \rangle$.*

Denote $\langle \vec{v}, \vec{v}' \rangle$ as $\vec{v}\vec{v}'^T, \vec{v}, \vec{v}' \in V$. From the geometry, the inner product $\langle \vec{v}, \vec{v}' \rangle$ gives us the production of the projection of one vector on the direction of the other vector. It introduces the concept of included angle in the k-mer vector space.

The length of k-mer vector $\vec{v}$ is $||\vec{v}|| = \sqrt{\langle \vec{v}, \vec{v} \rangle} = \sqrt{\vec{v}\vec{v}^T} = \sqrt{v_1^2 + v_2^2 + \cdots + v_{4^k}^2}$, where k is the k-mer length. For a sequence with uniform distribution of depth, it reflects the sequence length. Since the total number of kmer is $\sum_{i=1}^{4^k} v_i$, the average depth of coverage is $\frac{\sum_{i=1}^{4^k} v_i}{k*L}$. It is the principle of genome size (L) estimation using the k-mer distribution. We also have the Cauchy-Schwarz inequation $|\langle \vec{v}, \vec{v}' \rangle| \leq ||\vec{v}||||\vec{v}'||$. The equality holds when $\vec{v}, \vec{v}'$ have a linear dependence relation. It means that these two sequences are composed of the same k-mers. They can be two identical sequences with different read depths.The distance between k-mer vectors is defined as $d(\vec{v}, \vec{v}') = ||\vec{v} - \vec{v}'||$.

The included angle of k-mer vector $\vec{v}$ and $\vec{v}'$ is $\varphi = arccos\frac{\langle \vec{v}, \vec{v}' \rangle}{||\vec{v}||||\vec{v}'||}$. Then $\frac{\langle \vec{v}, \vec{v}' \rangle}{||\vec{v}||||\vec{v}'||}$ gives us the cosine similarity between two k-mer vectors which stands for two sequences. It shows how close the directions of two sequences are in the k-mer vector space. For a sequence set S in (2), SS^T gives us a symmetric matrix with each element the inner product of two vectors. We calculate it by

$$SS^T = \begin{bmatrix} \vec{v}_{s_1} \\ \vec{v}_{s_2} \\ \cdots \\ \vec{v}_{s_{n_S}} \end{bmatrix} \begin{bmatrix} \vec{v}_{s_1} \\ \vec{v}_{s_2} \\ \cdots \\ \vec{v}_{s_{n_S}} \end{bmatrix}^T \tag{3}$$

$$= \begin{bmatrix} \sum_{i=1}^{4^k} v_{1,i} * v_{1,i} & \sum_{i=1}^{4^k} v_{1,i} * v_{2,i} & \cdots & \sum_{i=1}^{4^k} v_{1,i} * v_{n_S,i} \\ \sum_{i=1}^{4^k} v_{2,i} * v_{1,i} & \sum_{i=1}^{4^k} v_{2,i} * v_{2,i} & \cdots & \sum_{i=1}^{4^k} v_{2,i} * v_{n_S,i} \\ \cdots & \cdots & \ddots & \cdots \\ \sum_{i=1}^{4^k} v_{n_S,i} * v_{1,i} & \sum_{i=1}^{4^k} v_{n_S,i} * v_{2,i} & \cdots & \sum_{i=1}^{4^k} v_{n_S,i} * v_{n_S,i} \end{bmatrix}_{n_S * n_S} \tag{4}$$

$$\to \begin{bmatrix} \frac{\sum_{i=1}^{4^k} v_{1,i} * v_{1,i}}{\sqrt{\sum_{i=1}^{4^k} v_{1,i}^2 \sum_{i=1}^{4^k} v_{1,i}^2}} & \cdots & \frac{\sum_{i=1}^{4^k} v_{1,i} * v_{n_S,i}}{\sqrt{\sum_{i=1}^{4^k} v_{1,i}^2 \sum_{i=1}^{4^k} v_{n_S,i}^2}} \\ \frac{\sum_{i=1}^{4^k} v_{2,i} * v_{2,i}}{\sqrt{\sum_{i=1}^{4^k} v_{2,i}^2 \sum_{i=1}^{4^k} v_{2,i}^2}} & \cdots & \frac{\sum_{i=1}^{4^k} v_{2,i} * v_{n_S,i}}{\sqrt{\sum_{i=1}^{4^k} v_{2,i}^2 \sum_{i=1}^{4^k} v_{n_S,i}^2}} \\ \cdots & \ddots & \cdots \\ \frac{\sum_{i=1}^{4^k} v_{n_S,i} * v_{1,i}}{\sqrt{\sum_{i=1}^{4^k} v_{n_S,i}^2 \sum_{i=1}^{4^k} v_{1,i}^2}} & \cdots & \frac{\sum_{i=1}^{4^k} v_{n_S,i} * v_{n_S,i}}{\sqrt{\sum_{i=1}^{4^k} v_{n_S,i}^2 \sum_{i=1}^{4^k} v_{n_S,i}^2}} \end{bmatrix}_{n_S * n_S} . \tag{5}$$

Denote expression (4) as $T_S = SS^T$ and expression (5) as T_{sim}. T_{sim} is the cosine similarity matrix of the sequence set S. In some practical problems, for example the de novo transcriptome assembly of RNA-Seq reads where the reads have identical read length, using $T_S/length$ is equivalent to T_{sim} since all the vectors have the same length.

The columns of S show the distribution of each distinct k-mer in the sequence set. The row rank of S shows the maximum number of linearly independent sequences. And since the row rank equals the column rank, if the n_S sequences in S are linearly independent, k selected k-mers will be sufficient to distinguish them. It is a theoretical support for using part of the k-mers in RNA-Seq quantification in Sailfish [12].

§3 The matrix forms of sequence-based problems

In this section, we will describe three sequence-based problems in their k-mer vector/matrix forms. It offers us a new angle to look at these problems and helps us to build new analysis algorithms.

3.1 Quantification of short reads to reference sequences

In some problems, such as RNA-seq analysis, one of the main steps is aligning the short reads back to the reference sequence and get the quantification result.

Let the matrix of reference sequences be S_r, and the matrix of short reads be S. So the total aligned reads number of each sequence in S_r can be obtained the following from.

$$S{S_r}^T = \begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,4^k} \\ v_{2,1} & v_{2,2} & \cdots & v_{2,4^k} \\ \cdots & \cdots & \ddots & \cdots \\ v_{n_S,1} & v_{n_S,2} & \cdots & v_{n_S,4^k} \end{bmatrix} \begin{bmatrix} v'_{1,1} & v'_{2,1} & \cdots & v'_{n_{S_r},1} \\ v'_{1,2} & v'_{2,2} & \cdots & v'_{n_{S_r},2} \\ \cdots & \cdots & \ddots & \cdots \\ v'_{1,4^k} & v'_{2,4^k} & \cdots & v'_{n_{S_r},4^k} \end{bmatrix} \tag{6}$$

$$= \begin{bmatrix} \sum_{i=1}^{4^k} v_{1,i} * v'_{1,i} & \sum_{i=1}^{4^k} v_{1,i} * v'_{2,i} & \cdots & \sum_{i=1}^{4^k} v_{1,i} * v'_{n_{S_r},i} \\ \sum_{i=1}^{4^k} v_{2,i} * v'_{1,i} & \sum_{i=1}^{4^k} v_{2,i} * v'_{2,i} & \cdots & \sum_{i=1}^{4^k} v_{2,i} * v'_{n_{S_r},i} \\ \cdots & \cdots & \ddots & \cdots \\ \sum_{i=1}^{4^k} v_{n_S,i} * v'_{1,i} & \sum_{i=1}^{4^k} v_{n_S,i} * v'_{2,i} & \cdots & \sum_{i=1}^{4^k} v_{n_S,i} * v'_{n_{S_r},i} \end{bmatrix}_{n_S * n_{S_r}} . \tag{7}$$

When a reference sequence is highly expressed in the RNA-seq problem, the duplicated short reads will result in a high module value of the corresponding k-mer vector. The sum of each column in SS^T is correlated with the total number of the short reads belong to each reference sequence:

$$c_t = \sum_{j=1}^{n_S} \sum_{i=1}^{4^k} v_{j,i} * v_{t,i}, t \in 1, 2, \cdots, n_{S_r}. \tag{8}$$

The widely used RPKM value is defined as RPKM= total exon reads/ (mapped reads (Millions) * exon length(KB)). It counts the reads over each sequence while some other methods use the number of fragments over each sequence. Short subsequence with fixed length can also be used in quantification [9]. From c_t in (8), we can get an approximation of the RPKM value of s'_t in S by

$$\frac{c_t * 10^9}{\sum_{t=1}^{n_{S_r} c_t} * ||s'_t||}. \tag{9}$$

Further adjustment can be applied when some necessary assumptions were made in the analysis.

The follow-up differentially expressed gene analysis can also be presented in the matrix form. Suppose we have an experimental group and control group with three repetitions each. Then these six data set can be merged into a $6 * n_{S_r}$ matrix using (7)(8):

$$A_{6*n_{S_r}} = \left[\begin{array}{c|c|c|c|c|c} 1,\cdots,1 & 0,\cdots,0 & & & & \\ 0,\cdots,0 & 1,\cdots,1 & 0,\cdots,0 & & & \\ & 0,\cdots,0 & 1,\cdots,1 & 0,\cdots,0 & & \\ & & 0,\cdots,0 & 1,\cdots,1 & 0,\cdots,0 & \\ & & & 0,\cdots,0 & 1,\cdots,1 & 0,\cdots,0 \\ & & & & 0,\cdots,0 & 1,\cdots,1 \end{array} \right] \begin{bmatrix} S_{1,1} \\ S_{1,2} \\ S_{1,3} \\ S_{2,1} \\ S_{2,2} \\ S_{2,3} \end{bmatrix} {S_r}^T \tag{10}$$

$$= \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n_{S_r}} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n_{S_r}} \\ a_{3,1} & a_{3,2} & \cdots & a_{3,n_{S_r}} \\ a_{4,1} & a_{4,2} & \cdots & a_{4,n_{S_r}} \\ a_{5,1} & a_{5,2} & \cdots & a_{5,n_{S_r}} \\ a_{6,1} & a_{6,2} & \cdots & a_{6,n_{S_r}} \end{bmatrix}_{6*n_{S_r}} \tag{11}$$

For each reference gene g_j, $j \in \{1, 2, \cdots, n_{S_r}\}$, its fold change and t-test value between experimental group and control group can be calculated by:

$$\begin{cases} FC_j = mean(a_{1,j}, a_{2,j}, a_{3,j})/mean(a_{4,j}, a_{5,j}, a_{6,j}), \\ t_j \frac{mean(a_{1,j},a_{2,j},a_{3,j})-mean(a_{4,j},a_{5,j},a_{6,j})}{\sqrt{sd(a_{1,j},a_{2,j},a_{3,j})^2/3+sd(a_{4,j},a_{5,j},a_{6,j})^2/3}}, j \in 1, 2, \cdots, n_{S_r} \end{cases} . \tag{12}$$

3.2 Sequence assembly problem

A sequence assembly problem is a clustering problem of the input sequence set based on the similarities. Using the similarity matrix T_{sim} (5), we can quickly get the similarity value between any two sequences in the set. Moreover, by doing the line and column switching of the similarity matrix, we can generate a block diagonal matrix. Each block is a cluster of sequences.

Here we will discuss the *de novo* transcriptome assembly problem of RNA-Seq reads. To simplify the problem, assume all the vectors have the same length. So we use T_S as the similarity matrix for discussion. For a set of sequenced reads, let the matrix of k-mer vectors be S. Then, we have

$$T_S = SS^T = \begin{bmatrix} ss_{1,1} & ss_{1,2} & \cdots & ss_{1,n_S} \\ ss_{2,1} & ss_{2,2} & \cdots & ss_{2,n_S} \\ \cdots & \cdots & \ddots & \cdots \\ ss_{n_S,1} & ss_{n_S,2} & \cdots & ss_{n_S,n_S} \end{bmatrix}_{n_S * n_S} \tag{13}$$

$$\xrightarrow{\text{line/column switching}} \begin{bmatrix} T_S^{(1,1)} & 0 & \cdots & 0 \\ 0 & T_S^{(2,2)} & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots \\ 0 & 0 & \cdots & T_S^{(m,m)} \end{bmatrix}_{n_S * n_S} . \tag{14}$$

For $T_S^{(i,i)}, i \in \{1, 2, \cdots, m\}$, where m is the number of the diagonal blocks, it can generate an assembled unique sequence. The vectors clustered into a same block $T_S^{(i,i)}$ have a high probability to have the same origin. To get a more accurate clustering result, we can set a cutoff of the similarity value in the similarity matrix. For $T_S = SS^T$,

$$T_S \xrightarrow{\text{set cutoff}} \begin{bmatrix} ss'_{1,1} & ss'_{1,2} & \cdots & ss'_{1,n_S} \\ ss'_{2,1} & ss'_{2,2} & \cdots & ss'_{2,n_S} \\ \cdots & \cdots & \ddots & \cdots \\ ss'_{n_S,1} & ss'_{n_S,2} & \cdots & ss'_{n_S,n_S} \end{bmatrix}_{n_S * n_S} \tag{15}$$

$$\rightarrow \begin{bmatrix} T_S^{(1,1)} & 0 & \cdots & 0 \\ 0 & T_S^{(2,2)} & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots \\ 0 & 0 & \cdots & T_S^{(m,m)} \end{bmatrix}_{n_S * n_S} , ss'_{i,j} = \begin{cases} ss_{i,j} & ss_{i,j} \geq p_{sim} * l \\ 0 & ss_{i,j} < p_{sim} * l \end{cases} , \tag{16}$$

$i, j \in \{1, 2, \cdots, n_S\}$, l the read length and p_{sim} the predefined similarity cutoff.

In T_S, take the first line $\vec{t_s^1}$ for example, $\vec{t_s^1}$ stands for the similarity between $\vec{s_1}$ and $\vec{s_i}, i \in 1, 2, \cdots, n_S$. Thus $\vec{t_s^1}$ shows us the correlated vectors in S of $\vec{s_i}$ whose value is greater than 0 (or the cutoff $p_{sim} * l$). These vectors form a set generated from $\vec{s_1}$. Then $T_S * \vec{t_s^1}$ is

$$T_S * \vec{t_s^1} = \begin{bmatrix} t_s^1 \\ t_s^2 \\ \cdots \\ t_s^{n_S} \end{bmatrix} \begin{bmatrix} \vec{v}_{s_1} \\ \vec{v}_{s_2} \\ \cdots \\ \vec{v}_{s_{n_S}} \end{bmatrix}^T = \begin{bmatrix} \sum_{i=1}^{n_S} ss_{1,i} * ss_{i,1} \\ \sum_{i=1}^{n_S} ss_{2,i} * ss_{i,1} \\ \cdots \\ \sum_{i=1}^{n_S} ss_{n_S,i} * ss_{i,1} \end{bmatrix}_{n_S * 1} , \tag{17}$$

shows the correlation between any vector in S and the vector cluster generated from $\vec{s_1}$. $T_S * \vec{t_s^1}$ gives us a clustered group generated from $\vec{s_1}$ with high accuracy since it is not considering the correlation between two vectors but a vector and a cluster of vectors.

3.3 Pattern detection

The detection of a fixed pattern from a sequence set, such as the detection of motifs [18, 19], is a widely seen sequence analysis problem. It can help us find a conserved domain or make the function prediction of a sequence. Here we will discuss two simplified possible situations with pattern length smaller than k-mer length k.

Let S be a set of candidate sequences. Let s_p be a query pattern and l_p the length of s_p, $l_p < k$. Then we have a set of $4^{(k-l_p)}$ possible k-mers that contains s_p. Denote it as S_p. Use all these $4^{(k-l_p)}$ k-mers to build a sequence vector, $\vec{v^p}$.

$$\vec{v^p} = (v_1^p, v_2^p, \cdots, v_{4^k}^p), v_i^p = \begin{cases} 1 & km_j \in S_p \\ 0 & km_j \notin S_p \end{cases}, i \in \{1, 2, \cdots, 4^k\}. \tag{18}$$

So we have

$$S * \vec{v^p} = \begin{bmatrix} \vec{v}_{s_1} \\ \vec{v}_{s_2} \\ \cdots \\ \vec{v}_{s_{n_S}} \end{bmatrix} \begin{bmatrix} v_1^p \\ v_2^p \\ \cdots \\ v_{4^k}^p \end{bmatrix}^T = \begin{bmatrix} \sum_{i=1}^{4^k} ss_{1,i} * v_i^p \\ \sum_{i=1}^{4^k} ss_{2,i} * v_i^p \\ \cdots \\ \sum_{i=1}^{4^k} ss_{n_S,i} * v_i^p \end{bmatrix}_{n_S*1} . \tag{19}$$

Any positive value in $S * \vec{v}^p$ indicates a sequence in S with a possible pattern s_p.

Let s_{pp} be the query pattern and l_{pp} the length of s_{pp}, $l_{pp} < k$. Now suppose the first and second base of s_{pp} is not fixed. Let $P_1(x)$ ($P_2(x)$), $x \in \{A, G, C, T\}$, denote the probability of the first (second) base being A/G/C/T, respectively. Then we have $C_4^1 * C_4^1 = 16$ possible query patterns with occurrence possibility $P_1(x) * P_2(y), x, y \in \{A, G, C, T\}$. From each one of the 16 patterns, we can get a k-mer vector as we did in the above situation:

$$\vec{v}_i^{pp} = (v_{i,1}^{pp}, v_{i,2}^{pp}, \cdots, v_{i,4^k}^{pp}), v_i^{pp_i} = \begin{cases} P(x) * P(y) & km_j \in S_{pp} & j \in \{1, 2, \cdots, 4^k\}, \\ 0 & km_j \notin S_{pp} & x, y \in \{A, G, C, T\}. \end{cases} \tag{20}$$

$i \in \{1, 2, \cdots, 16\}$. So we have

$$S * \begin{bmatrix} \vec{v}_1^{pp} \\ \vec{v}_2^{pp} \\ \cdots \\ \vec{v}_{16}^{pp} \end{bmatrix}_{16*4^k}^T = \begin{bmatrix} \sum_{i=1}^{4^k} ss_{1,i} * v_{1,i}^p, \cdots, \sum_{i=1}^{4^k} ss_{1,i} * v_{16,i}^p \\ \sum_{i=1}^{4^k} ss_{2,i} * v_{1,i}^p, \cdots, \sum_{i=1}^{4^k} ss_{2,i} * v_{16,i}^p \\ \cdots \\ \sum_{i=1}^{4^k} ss_{n_S,i} * v_{1,i}^p, \cdots, \sum_{i=1}^{4^k} ss_{n_S,i} * v_{16,i}^p \end{bmatrix}_{n_S*16} . \tag{21}$$

Any positive value in expression (21) indicates a sequence in S with a possible form of pattern s_{pp}. It is an example of solving a sequence-based problem with indeterminate bases using the k-mer vector method. For more complicated situations, the Monte Carlo method could be a choice.

§4 Algorithm implementation and discussion

Many methods are capable of solving the assembly problem of biological sequences [20, 21, 22]. Most of their algorithm design strategy is based on the common string representation of sequences. Here, we will take the assembly problem in the matrix form discussed in section 3.2 as an example of the algorithm design in k-mer vector space. We will explain how the theory of k-mer vector and matrix is connected with real problems in sequence-based biological problem analysis.

4.1 Design and implementation of vector-based algorithm for assembly problem

We developed a program named iLoqu to implement the use of the similarity matrix in the k-mer space for sequence assembly. It is written in C++ and can be executed under either 32-bit or 64-bit Linux systems or windows system. To calculate the similarity matrix, we used the hashmap from C++. iLoqu contains three main steps: 1. extract the k-mers from the sequences to generate a hashmap; 2. use the hashmap to calculate the similarity matrix; 3. cluster the k-mer vectors into groups and do the assembly of each group.

iLoqu takes Fasta file as its input. The program starts with reading the sequences from the Fasta file and extracts all the k-mers to create a hashmap. The matrix of k-mer vectors generated from the sequence set is a sparse matrix. The data structure of the hashmap provides us with a feasible method to calculate the similarity matrix while reducing storage requirement. The k-mers from each sequence are added to the hashmap within a for-loop. Each k-mer is a key in the hash table, and other corresponding information of the sequence is stored in its value. Thus in the hashmap, the key is the column index of the matrix while the value contains the line information.

The second step is generating the similarity matrix using the hashmap built in the first step. While executing the algorithm, the generating of the similarity matrix is operated with step one in the same while loop of reading sequences. See algorithm-1. The similarity matrix is built from the upper-left corner to the right-bottom corner and it is a symmetric matrix. Mention that a symmetric matrix is a square matrix that is equal to its transpose: A is symmetric if and only if $A = A^T$. Here we set a cutoff value to remove the low similarity data. Also, two vectors coming up with an extremely high production value can be merged into one in advance before the following clustering step. Here, the inner product of two k-mer vectors, $\vec{v}, \vec{v'}$, was set to 0 if it is smaller than the preset similarity cutoff.

The last step is clustering the k-mer vectors basing on the similarity matrix. After processing the inner product value by filtering with a cutoff, the line coordinate and column coordinate of any positive value in the similarity matrix gives us two vectors that could be grouped into one. For the final assembly, since the number of sequences in each group is greatly reduced compared to the whole sequence set, many assembly methods can handle this situation. Here in the iLoqu, we also presented an assembling procedure using the k-mer. On the other hand,

Algorithm 1 Read the input sequences, build hashmap and calculate the similarity

```
while read-in sequence s do
   for $i \leftarrow 1$ to $n - k + 1$ do
      $kmer \leftarrow Getkmer(s, i, k)$
      $Array.Add(Hash\_seq(kmer))$
      $Hash\_seq(kmer) \leftarrow seq\_info$
   end for
   for $i \leftarrow 1$ to $Array.length$ do
      $Hash\_seq(id) + +$
   end for
   $similarity \leftarrow Hash\_seq(id)/sqrt(seq1.length * seq2.length)$
end while
```

if the assembly result is going to be used in the k-mer vector form in the downstream analysis, such as the quantification step described in section 3.1, there is no need for an actual assembly. We can generate a set of reference vectors from the block diagonal matrix result.

The assembly algorithm is described in algorithm 2. Starting from vectors with big length, we select one seed k-mer from the vector and look up the next k-mers on its two sides from the hash map.

Algorithm 2 Sequence assembly from the hashmap

```
for $i \leftarrow 1$ to $m$ do
   for $j \leftarrow 1$ to $n - k + 1$ do
      $kmer \leftarrow Getkmer(s, i, k)$
      $Hash\_assembly(kmer) \leftarrow seq\_info$
   end for
end for
for $key$ in Hash_assembly do
   if $key$ is not used as seed yet then
      $seed \leftarrow key$
      $nextkey \leftarrow substring(seed, 2, k - 1) + (A/C/G/T)$
      while $Hash\_assembly(nextkey)$ exists do
         $Seed \leftarrow seed + (A/C/G/T)$
         $nextkey \leftarrow substring(seed, length(seed) - k + 2, k - 1) + (A/C/G/T)$
      end while
   end if
end for
```

4.2 Numerical example

iLoqu is a tool designed for sequence assembly based on the concept of the k-mer vector. It is suitable for the overlap-based assembly problems of RNA-Seq. As mentioned above, the assembly process from the grouped vectors to string form is not necessary if it is going to be used in the downstream sequence-based analysis in vector/matrix form. So we will not compare the

Table 1. The test result of iLoqu with the simulating data and real data.

	data size	data coverage/ depth	assembled size	assembled coverage	running time
Arabidopsis thaliana					
simulated test data	400	98.3%, 10x	10	98.3%	3.2 s
real test data	18,438	79.2%, 50x	24	68.3%	8.5 min
Homo sapiens					
simulated test data	400	96.6%, 10x	10	96.6%	4.8 s
real test data	27,199	92.9%, 50x	16	77.1%	6.1 min

assembly speed with other tools but show the assembly accuracy. We made a test on simulating data and real biology data using iLoqu software. For the real RNA-Seq data, we used the reads that can be aligned to ten selected genes by Bowtie [23]. These ten genes are randomly selected from the cDNA file of Arabidopsis/Homo sapiens (file Araport11_genes.201606.cdna.fasta of *Arabidopsis thaliana* and the file GCF_000001405.38_GRCh38.p12_rna.fna of *Homo sapiens*) with a high expression level (at least 1000 reads covered) in the RNA-Seq sequencing dataset (SRR4024923.sra of *Arabidopsis thaliana* and ERR2870199.sra of *Homo sapiens*). These files can be downloaded from NCBI. These ten genes are also used for the generation of simulated data as a reference. We randomly generated a set of sequences normally distributed along with the reference genes with a length greater than 100 bp. And we set a 1% random SNP (single nucleotide polymorphism) variation on the simulated sequence dataset. The parameters used in iLoqu are minimum similarity percentage 1%, k-mer size 19, a minimum count of tag 30 and minimum assembly similarity 90%. Table 1 shows the assembly results of both test data. The assembly results were aligned back to the reference to show the accuracy using [24].

From Table 1, we can see that iLoqu successfully assembled the test data. In the test of *Arabidopsis thaliana* data, 98.3% and 68.3% of the reference is covered by the assembly result from the simulating data and real data, respectively. In the test of *Homo sapiens* data, the percentage is 96.6% and 77.1%. The depth and variations will affect the assembly quality. This result proves that the k-mer vector methods is a feasible solution for the assembly problem and it worth further studies.

4.3 Discussion

The usage of the generalized k-mer vector defined in this work is a method for describing a biological sequence or sequence set from its k-mer set. It is defined using the most basic common features extracted from different sequence analysis problems. Thus, it is a more general description method with better applicability and scalability when compared to existing research on matrix methods for sequence analysis problems. It has higher adaptability facing new problems. It also has better problem analysis capabilities when multiple sequence-based problems are combined as a whole. Many sequence-based problems in bioinformatics can be converted to their corresponding vector/matrix form using the k-mer vector under this definition. It can unmask part of the real nature of the sequence analysis problems that cannot be seen directly in other basic forms. It offers us a chance to rebuild a new analysis system.

There are complete research findings on linear space, matrix and matrix operation from linear algebra and computational mathematics. It gives us a whole new set of methods to deal with the sequence-based problems in bioinformatics studies. The challenge here is to use the language of biology to explain the mathematical concepts. In this work, we extend the value domain of the element in the k-mer vector from integer to real number. Furthermore, the vector operations are interpreted as operations of the k-mer sets of generalized sequences. It makes the matrix methods of biological sequences a practical tool for problem analysis and algorithm design.

The computation of sequence similarity is one of the common features of many sequence-based biological problems, such as the sequence alignment with reference, sequence assembly problem, and sequence evolution analysis. These features can be described uniformly using the k-mer vector form. It links different sequence-based problems or their upstream and downstream analysis as a whole, reducing the computational costs of the associated algorithm. The successful algorithm design examples using k-mer (for example, kallisto that avoid sequence alignment using k-mer [11]), indicates that k-mer is an effective tool for solving high complexity bioinformatics problems. While complimenting the algorithm design using k-mer vector/matrix form in the instance of sequence assembly problem, we find that the hashmap structure, k-mer and the sparse matrix match very well in the real problem. It is another evidence to support the feasibility of applying the sequence matrix methods theories in the bioinformatics.

The k-mer vector uses features extracted from a sequence to represent it. It has better control of the information redundancy which could be a burden for the design of computational analysis algorithms. On the other hand, it may cause a loss of information. More theoretical analysis and quantification tests need to be done to find the balance between different forms. The mapping between vectors in the space and the sequences is not a bijection. In practice, the effect of this shortcoming is limited and ignorable. It has been proved and used as a reasonable assumption by other researchers [6]. A remaining big challenge is the visualization of the sequences from its matrix form. The string representation of biological sequences has its advantage naturally while the vector form makes the sequence structure abstract visually.

In the future work, we will present a solution for sequence visualization from its vector form. Then match more concepts from matrix analysis in algebra with the sequence analysis. Finally, try to build equations from sequence analysis problems and solve them in matrix form. Moreover, we need a high-effective data structure and improved algorithms for solving real problems.

§5 Conclusions

We gave a generalized definition of the k-mer vector and its linear space for the biological sequences. It is a method of representing k-mer set in the vector form to describe a sequence-based problem. The basic concepts and operations in vector space are reasonably related to their corresponding biological meanings. We use three examples to explain the theory, including the use of basic operations, the relationship between upstream and downstream analysis steps, and cases with probabilities. This is a way to discover new features of sequence analysis problems in bioinformatics and generate new algorithm design strategies. Also, we proposed a solution to the sequence assembly problem as an application example of algorithm design using the vector form of biological sequence, which proves the feasibility of the vector (matrix) method in real sequence analysis.

References

[1] El Mustapha Bahassi, Peter J Stambrook. *Next-generation sequencing technologies: breaking the sound barrier of human genetics*, Mutagenesis, 2014, 29(5): 303-310.

[2] Rob Patro, Stephen M Mount, Carl Kingsford. *Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms*, Nature Biotechnology, 2014, 32(5): 462.

[3] Xin Bai, Kujin Tang, Jie Ren, Michael Waterman, Fengzhu Sun. *Optimal choice of word length when comparing two Markov sequences using a x2-statistic*, BMC Genomics, 2017, 18(6): 732.

[4] Nafiseh Jafarzadeh, Ali Iranmanesh. *C-curve: A novel 3d graphical representation of DNA sequence based on codons*, Mathematical Biosciences, 2013, 241(2): 217-224.

[5] B D Pickett, J B Miller, P G Ridge. *Kmer-SSR: A Fast and Exhaustive SSR Search Algorithm*, Bioinformatics, 2017, 219(24): 178.

[6] Heng Li. *Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM*, arXiv:1303.3997 [q-bio], 2013, arXiv: 1303.3997.

[7] Shuyan Ding, Qi Dai, Hongmei Liu, Tianming Wang. *A simple feature representation vector for phylogenetic analysis of DNA sequences*, Journal of Theoretical Biology, 2010, 265(4): 618-623.

[8] Mihai Pop, Steven L Salzberg. *Bioinformatics challenges of new sequencing technology*, Trends in Genetics, 2008, 24(3): 142-149.

[9] Subhram Das, Tamal Deb, Nilanjan Dey, Amira S Ashour, D K Bhattacharya, D N Tibarewala. *Optimal choice of k-mer in composition vector method for genome sequence comparison* , Genomics, 2018, 110(5): 263-273.

[10] Jonathan D Wren, David Johnson, Le Gruenwald. *Automating Genomic Data Mining via a Sequence-based Matrix Format and Associative Rule Set* , BMC Bioinformatics, 2005, 6(2): S2.

[11] Sebastian Deorowicz, Marek Kokot, Szymon Grabowski, Agnieszka Debudaj-Grabysz. *KMC 2: fast and resource-frugal k-mer counting*, Bioinformatics, 2015, 31(10): 1569-1576.

[12] Nicolas Bray, Harold Pimentel, Pll Melsted, Lior Pachter. *Near-optimal RNA-Seq quantification*, arXiv:1505.02710, 2015.

[13] Daniel R Zerbino, Ewan Birney. *Velvet: Algorithms for de novo short read assembly using de Bruijn graphs*, Genome Research, 2008, 18(5): 821-829.

[14] Aleksey V Zimin, Guillaume Marais, Daniela Puiu, Michael Roberts, Steven L Salzberg, James A Yorke. *The MaSuRCA genome assembler*, Bioinformatics, 2013, 29(21): 2669-2677.

[15] Ben Langmead, Cole Trapnell, Mihai Pop, Steven L Salzberg. *Ultrafast and memoryefficient alignment of short DNA sequences to the human genome*, Genome Biology, 2009, 10(3): R25.

[16] Giuseppe Lancia. *Mathematical Programming in Computational Biology: an Annotated Bibliography*, Algorithms, 2008, 1(2): 100-129.

[17] Marais G, Kingsford C. *A fast, lock-free approach for efficient parallel counting of occurrences of k-mers*, Bioinformatics (Oxford, England), 2011, 27(6): 764.

[18] Slatko Be, Gardner Af, Ausubel Fm. *Overview of Next-Generation Sequencing Technologies*, Current Protocols in Molecular Biology, 2018,122(1): e59-e59.

[19] Ping-an He, Dan Li, Yanping Zhang, Xin Wang, Yuhua Yao. *A 3d graphical representation of protein sequences based on the Gray code*, Journal of Theoretical Biology, 2012, 304: 8-87.

[20] Bin Fu, Yunhui Fu, Yuan Xue. *Sublinear Time Motif Discovery from Multiple Sequences*, Algorithms, 2013, 6(4): 636-677.

[21] Jia Wen, YuYan Zhang, Stephen S T Yau. *k-mer Sparse matrix model for genetic sequence and its applications in sequence comparison*, Journal of Theoretical Biology, 2014, 363: 145-150.

[22] Yao-Ting Huang, Chen-Fu Liao. *Integration of string and de Bruijn graphs for genome assembly*, Bioinformatics, 2016, 32(9): 1301-1307.

[23] Jinyu Yang, Anjun Ma, Adam D Hoppe, Cankun Wang, Yang Li, Chi Zhang, Yan Wang, Bingqiang Liu, Qin Ma. *Prediction of regulatory motifs from human Chip-sequencing data using a deep learning framework*, Nucleic Acids Research, 2019, 47(15): 7809-7824.

[24] Z H You, J Li, X Gao, Z He, L Zhu, Y K Lei, Z Ji. *Detecting protein-protein interactions with a novel matrix-based protein sequence representation and support vector machines*, BioMed research international, 2015, 2015: 867516-867516.

[1]Department of Mathematics, Zhejiang University, Hangzhou 310027, China.

[2]Zhejiang Provincial Key Laboratory of Horticultural Plant Integrative Biology, Zhejiang University, Zijingang Campus, Hangzhou 310012, China.

Email: qbwu@zju.edu.cn