ELSEVIER Theoretical Computer Science 289 (2002) 783–800

Theoretical Computer Science

www.elsevier.com/locate/tcs

Algorithms for computing lengths of chains in integral partition lattices

Honghui Wan*, John C. Wootton

Computational Biology Branch, National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Building 38A, 8th Floor, 8600 Rockville Pike, Bethesda, MD 20894, USA

Received April 2000; received in revised form October 2001; accepted October 2001
Communicated by A. Apostolico

Abstract

Let $P_{l,n}$ denote the partition lattice of l with n parts, ordered by Hardy–Littlewood–Polya majorization. For any two comparable elements $\mathbf{x}$ and $\mathbf{y}$ of $P_{l,n}$, we denote by $M(\mathbf{x},\mathbf{y})$, $m(\mathbf{x},\mathbf{y})$, $f(\mathbf{x},\mathbf{y})$, and $F(\mathbf{x},\mathbf{y})$, respectively, the sizes of four typical chains between $\mathbf{x}$ and $\mathbf{y}$: the longest chain, the shortest chain, the lexicographic chain, and the counter-lexicographic chain. The covers $\mathbf{u}=(u_1,\dots,u_n)\succ\mathbf{v}=(v_1,\dots,v_n)$ in $P_{l,n}$ are of two types: N-shift (nearby shift) where $v_i=u_i-1$, $v_{i+1}=u_{i+1}+1$ for some i; and D-shift (distant shift) where $u_i-1=v_i=v_{i+1}=\cdots=v_j=u_j+1$ for some i and j. An N-shift (a D-shift) is *pure* if it is not a D-shift (an N-shift). We develop linear algorithms for calculating $M(\mathbf{x},\mathbf{y})$, $m(\mathbf{x},\mathbf{y})$, $f(\mathbf{x},\mathbf{y})$, and $F(\mathbf{x},\mathbf{y})$, using the leftmost pure N-shift first search, the rightmost pure D-shift first search, the leftmost N-shift first search, and the rightmost D-shift first search, respectively. Those algorithms have significant applications in complexity analysis of biological sequences. © 2002 Elsevier Science B.V. All rights reserved.

MSC: 68Q20; 06B99; 68R15; 92B99; 92-08

Keywords: Algorithm; Length; Chain; Partition lattice; Majorization; Nearby shift; Distant shift

1. Introduction

Many attempts have been made to analyze the compositional complexity of biological sequences. In particular, we develop fully the axiomatic foundation of compositional

* Corresponding author. Fax: +1-301-435-2433.
E-mail address: hwan@nih.gov (H. Wan).

complexity functions of biological sequences based on only three underlying postulates of monotonicity, nonnegativity, and normalizability [19]. The term "*compositional complexity*" is used here to denote the complexity only based on residue composition, regardless of the patterns or periodicity of sequence repetitiveness. Why have compositional complexity functions of nucleic acid and protein sequences proved to be remarkably informative for inferring, describing and understanding biological properties? The nonspecific generality of complexity concepts may seem to be inconsistent with the goals of much current research that seeks precise molecular details of biological structures, dynamics, interactions and evolution. However, these important details cannot be inferred for a large proportion of genomic and deduced protein sequences for which relevant experimental data or homologous precedents are lacking. Unbiased methods of description and inference are necessary to explore sequence data for new discoveries, and two scientific principles make compositional complexity measures a particularly important part of this inference.

First, DNA and protein sequences do not, in general, resemble random strings of letters, but rather consist of a heterogeneous mixture of local regions with distinct genetic functions and evolutionary origins. These regions show many different compositional characteristics and types of sequence patterns, as if written in a mosaic of different languages. Genomic sequences show, for example, coding sequences, untranslated regions, introns, exons, intergenic regions, promoters, terminators, regulatory signals, RNA genes, direct or inverted repeats of widely different sizes in tandem or interspersed arrangements, microsatellites, CpG islands, centromeres, telomeres, and origins of replication. A large fraction of protein sequences consist of multiple domains or modules, including globular folds (many of which may be classified by sequence homology), helical nonglobular rods and fibres, mobile linkers, low-complexity interaction domains, and membrane interaction segments.

It is well established that the statistical structure of sequences cannot be generally well described by simple random or Markov models [7,27]. From the viewpoint of information theory, real sequences are not generated by a simple stochastic process from a stationary source, and they are not ergodic in their scaling properties. Nevertheless, methods and algorithms based on local combinatorial complexity of composition have been successful for analyzing, classifying and segmenting these functionally distinct regions of DNA and protein sequences [23,24,25,26,27,28].

Second, local sequence complexity of proteins may be thought of as a physical molecular property, related to the ability of any polypeptide chain to adopt a unique globular fold. This property is embodied in the concept that proteins are edited statistical polymers [8] that have evolved many different sequences of random nature [5], each of stereospecific structure. High sequence complexity is also consistent with the random energy model of protein folding, which corresponds to the spin glass theory of polymer physics [9,4]. Complexity measurements on natural protein sequences are generally consistent with this theoretical perspective [24,25] with a few exceptions, globular domains are generally of high compositional complexity, contrasting with nonglobular or conformationally mobile lower-complexity regions which show regular or irregular sequence repetition.

In most previous work on biological sequences, the ‘local’ complexity measures have been based on well-known functions in information theory or statistical theory, such as informational entropy, information content, data compression schemes, or fractal dimension. Those ‘local’ measures resemble entropy functions and are inherently dependent on an underlying probability distribution. Local measures cannot rigorously compare complexity across sequences of substantially different size, because real sequences show very irregular heterogeneity and do not have the necessary ergodicity in scaling and asymptotic properties. Recently, we have started to develop a new class of scale-independent, distribution-independent complexity functions by means of four types of chains on integer partition lattices (see Section 2.5). Their scaling properties do not depend on the assumption of ergodicity [20,19], as required for global comparisons of genomic or protein sequences of any size, which overcomes a crucial limitation of earlier methods. A member G_1 of the new class, derived from the longest chain in the integer partition lattice, has some important applications in molecular evolution. Actually, the distributions of G_1 were calculated in [20] for the entire sets of translated proteins encoded by extensively sequenced genomes. The results establish the existence of a clear evolutionary principle, common to bacteria, archaea and eukaryotes, that the proteins encoded by more extreme AT-rich and GC-rich genomes have generally lower compositional complexity than those of more typical organisms. In addition, the points of intersection of G_1 and G_2, another complexity function defined by using the shortest chain in the integer partition lattice, provide a natural basis for segmentation of a sequence and particularly a new possibility to find out those functional distinct regions of sequences [22]. Those points of intersection correspond to the points of contact between globular and non-globular domains in protein sequences.

To implement and apply these new types of complexity measures for analyzing and classifying functionally distinct regions of both nucleotide and amino acid sequences, it is crucial to develop efficient algorithms for computing sizes of chains in integral partition lattices. A goal of this paper is to devise novel algorithms to calculate four typical chains between a comparable pair in the partition lattice, defined in Section 2.4: the longest chain, the shortest chain, the lexicographic chain, and the counter-lexicographic chain.

2. State vectors and integral partition lattices

This section is devoted to a survey of concepts and results about state vectors and integral partition lattices, which will be needed later.

2.1. State vectors

Let $\mathbf{A} = \{a_1, a_2, \ldots, a_n\}$ denote an alphabet in which a_i is called the letter of type i ($1 \leqslant i \leqslant n$). In particular, the 20-letter, amino acid alphabet of proteins or the purine/pyrimidine alphabet for DNA can be used. Let $\mathbf{A}^l$ denote the set of all sequences of length l over $\mathbf{A}$ for a positive integer l. For a sequence $s \in \mathbf{A}^l$, s_i is the ith symbol (or residue) of s. In addition, we denote by u_i the number of occurrences of a_i in s $(1 \leqslant i \leqslant n)$. Clearly, $u_1 + u_2 + \cdots + u_n = l$ and $0 \leqslant u_i \leqslant l$. The vector $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ is called the *composition vector* of s, and the configuration $a_1^{u_1} a_2^{u_2} \cdots a_n^{u_n}$ is called a *composition* of s. For convenience, we may omit those terms in configurations whose corresponding letters do not appear in s. The vector $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ is called the (*complexity*) *state vector* of s where $v_1 \geqslant v_2 \geqslant \cdots \geqslant v_n$ is the decreasing rearrangement of $u_1, u_2, \ldots, u_n$. The set of sequences with the state vector $\mathbf{v}$ is called $\mathbf{v}$-*equivalence class or* $\mathbf{v}$-*composition class*, denoted by $\mathscr{S}(\mathbf{v})$. For example, the nucleotide sequence TTGTGTTT has $u_1 = u_2 = 0$, $u_3 = 2$, $u_4 = 6$; $v_1 = 6$, $v_2 = 2$, $v_3 = v_4 = 0$, $n = 4$ and $l = 8$. The alphabet in this case is $\mathbf{A} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$, the composition vector is $(0, 0, 2, 6)$. The composition is $\mathtt{G}^2\mathtt{T}^6$ and the state vector is $\mathbf{v} = (6, 2, 0, 0)$. $|\mathscr{S}(\mathbf{v})| = 336$. The $(6, 2, 0, 0)$-equivalence class $\mathscr{S}(6, 2, 0, 0)$ consists of 336 sequences and the set $\mathbf{A}^8$ of 8-nucleotide sequences contains 65536 sequences.

The state vector provides an excellent "data structure" for intrinsically representing sequences. This type of representation possesses several advantages: it is informative and of intuitive biological significance, and it is computationally simple and efficient. A partition of a positive integer l is a representation

$$l = p_1 + p_2 + \cdots + p_n \quad (p_1 \geqslant p_2 \geqslant \cdots \geqslant p_n \geqslant 0). \tag{1}$$

We also use the vector representation for the partition: $(p_1, p_2, \ldots, p_n)$. The numbers $p_1, \ldots, p_n$ are the *parts* of the partition, in which p_1 is called the *largest part*. Hence (1) is a partition of l into n parts. All partitions and vectors throughout this paper are integral, whose parts or components are arranged in decreasing order (we may allow trailing zero components in vectors and partitions), unless otherwise specified.

From the viewpoint of combinatorial set theory, the set $\mathbf{A}^l$ can exactly be partitioned into the union of $\mathbf{v}$-equivalence classes:

$$\mathbf{A}^l = \bigcup_{v_1 + v_2 + \cdots + v_n = l} \mathscr{S}(v_1, v_2, \ldots, v_n),$$

where the union ranges over all partitions of l with n parts (every partition is ranked in decreasing order). Thus, it is natural to consider the set of partitions of a positive integer associated with $\mathbf{A}^l$.

For any partition $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ of l with n parts, we denote the kth partial sum by $\sigma_k(\mathbf{z})$, i.e.

$$\sigma_k(\mathbf{z}) = z_1 + z_2 + \cdots + z_k.$$

Also, we write $\sigma(\mathbf{z}) = (\sigma_1(\mathbf{z}), \sigma_2(\mathbf{z}), \ldots, \sigma_n(\mathbf{z}))$ and call it to be the *partial-sum vector* of $\mathbf{z}$. For example, the partial-sum vector of $(4, 2, 1, 1)$ is $(4, 6, 7, 8)$.

2.2. Majorization and integral partition lattices

Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ be two partitions of l with n parts. We say $\mathbf{y}$ *majorizes* $\mathbf{x}$, write it as $\mathbf{x} \prec \mathbf{y}$ or $\mathbf{y} \succ \mathbf{x}$, if the partial sums of $\mathbf{y}$ are at least as big as the corresponding partial sums of **x**:

$$\sigma_k(\mathbf{x}) \leqslant \sigma_k(\mathbf{y}) \quad (k = 1, 2, \ldots, n-1) \quad \text{and} \quad \sigma_n(\mathbf{x}) = \sigma_n(\mathbf{y}).$$

That is, **y** majorizes **x** if and only if the partial-sum vector of **y** is componentwisely greater than or equal to that of **x**: $\sigma(\mathbf{y}) \geqslant \sigma(\mathbf{x})$. For example, $(4,2,1,1) \prec (4,2,2,0)$. However, there is no the majorization ordering relationship between $(4,2,1,1)$ and $(3,3,2,0)$.

The notation and terminology of majorization of real numbers are due to Hardy, Littlewood and Polya [6]. This mathematical concept has been shown to have many applications to problems in the fields of statistics, economics, ecology, sociology, political science, system science, operations research, and information theory, as well as in many branches of mathematics, such as combinatorics, algebra, geometry, and matrix theory. For two fixed positive integers l and n, we define $(P_{l,n}, \prec)$ to be the partially ordered set (poset) consisting of all partitions of l with n parts, ordered by majorization [10,11]. An element **x** in $P_{l,n}$ is the *top* (*bottom*) element of $P_{l,n}$ if $\mathbf{x} \succ \mathbf{z}$ $(\mathbf{x} \prec \mathbf{z})$ for every **z** in $P_{l,n}$. Divide l by n and let q be the quotient and r be the remainder $(0 \leqslant r < n)$. Then $P_{l,n}$ is a lattice with top element $\mathbf{x}_{\text{top}}^{(l)} = (l, 0, \ldots, 0)$ and bottom element

$$\mathbf{x}_{\text{bot}}^{(l)} = (\underbrace{q+1, \ldots, q+1}_{r}, q, \ldots, q),$$

that is, for any two elements **x** and **y** in $P_{l,n}$, there exist *the supremum* (*the least upper bound*) and *the infimum* (*the greatest lower bound*) of **x** and **y**. Further explanations of these terms, together with the proofs of stated results here are found in [10,11,13]. $P_{l,n}$ is called an *n-part partition lattice* of l.

The set of state vectors of all sequences in $\mathbf{A}^l$ on alphabet **A**, ordered by majorization, is identical with the lattice $(P_{l,n}, \prec)$. The size of the partition lattice $P_{l,n}$, denoted by $p_n(l)$, i.e., the accurate number of state vectors in $P_{l,n}$, is computable from well-established principles of combinatorial number theory [2,26], although there is no elementary explicit formulation for it. Actually, the investigation of the deeper properties of $p_n(l)$ was one of the jewels of 20th century analysis, involving researches of Hardy and Ramanujan and further work by Rademacher. The whole story is described in Andrews [2]. To estimate the value of $p_n(l)$, the following asymptotic formula is useful [11,14]:

$$p_n(l) \sim \frac{(l+n)^{n-1}}{n!(n-1)!}(l \to \infty).$$

Clearly, the number $p_n(l)$ becomes very big for a large l. For example, there are a (rounded) total of 1.1×10^{52} amino acid sequences of length 40 on the 20-letter protein alphabet, which correspond to 35251 complexity state vectors, i.e., $p_{20}(40) = 35251$ [26,27].

2.3. Order diagram

Two distinct elements $\mathbf{x}$ and $\mathbf{y}$ of $(P_{l,n}, \prec)$ are *comparable* if either $\mathbf{x} \prec \mathbf{y}$ or $\mathbf{y} \prec \mathbf{x}$, and *incomparable* otherwise; $\mathbf{y}$ is said to *cover* $\mathbf{x}$ if $\mathbf{x} \prec \mathbf{y}$ and there is no element $\mathbf{z}$ in $P_{l,n}$ such that $\mathbf{x} \prec \mathbf{z} \prec \mathbf{y}$. If $\mathbf{y}$ covers $\mathbf{x}$ in $(P_{l,n}, \prec)$, then $\mathbf{x}$ is called a *lower cover* of $\mathbf{y}$, $\mathbf{y}$ is called an *upper cover* of $\mathbf{x}$, and $\{\mathbf{x}, \mathbf{y}\}$ is called a *covering pair*. For $(P_{l,n}, \prec)$, it is clear that the entire relation is determined by the covering relation. The following result describes the covering relation in the partition lattice.

Proposition 2.1 (Wan [11,12]). *Let* $\mathbf{x} = (x_1, x_2, \dots, x_n) \prec \mathbf{y} = (y_1, y_2, \dots, y_n)$ *in* $P_{l,n}$. *Then* $\mathbf{y}$ *covers* $\mathbf{x}$ *if and only if there exist indices* $i < j$ *such that* $\mathbf{x} = \mathbf{y} - \mathbf{e}_i + \mathbf{e}_j$, *and either* $j = i + 1$ *or* $x_i = x_{i+1} = \cdots = x_j$, *where* $\mathbf{e}_k$ *denotes the vector with one in the* kth *position and zeros elsewhere*:

$$\mathbf{e}_k = (\underbrace{0, \dots, 0, 1}_{k}, \underbrace{0, \dots, 0}_{n-k}).$$

The covering relation of a poset can be displayed via an graphical rendering with an implied upward orientation, which is called the *order diagram*. A point is drawn for each element of the poset, and line segments are drawn between these points according to the following two rules: (i) If one element $\mathbf{x}$ is less that another element $\mathbf{y}$ in the poset, then the point corresponding to $\mathbf{x}$ appears lower in the drawing than the point corresponding to $\mathbf{y}$. (ii) The line segment between the points corresponding to any two elements $\mathbf{x}$ and $\mathbf{y}$ of the poset is included in the drawing if and only if $\mathbf{x}$ and $\mathbf{y}$ are a covering pair.

In graph-theoretic terms, a finite poset P is a lattice if and only if for any two elements $\mathbf{x}$ and $\mathbf{y}$ in P, there exist a unique *least common ancestor* and a unique *greatest common descendant* of $\mathbf{x}$ and $\mathbf{y}$ in its order diagram. This is an essential combinatorial characterization of finite lattices [11,16], and, indeed is the graph-theoretic definition of a finite lattice. As a lattice, $P_{l,n}$ has special restricted properties, compared with acyclic directed graphs in general, such as being triangle-free and having conjugate relationships between pairs of elements.

As an example, Fig. 1, left column illustrates the order diagram of the 4-part integral partition lattice $P_{8,4}$ of 8 which shows the fifteen possible state vectors of 8-nucleotide sequences on the 4-letter DNA alphabet $\mathbf{A} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$. The top element is $\mathbf{x}^{(8)}_{\text{top}} = (8, 0, 0, 0)$ and the bottom element is $\mathbf{x}^{(8)}_{\text{bot}} = (2, 2, 2, 2)$. One of these state vectors, $(6, 2, 0, 0)$, has 12 different compositions shown in middle column, with different nucleotides assigned to the four numbers in this vector, and each of these compositions has 28 possible sequences, as indicated on the right. The $(6, 2, 0, 0)$-equivalence class $\mathscr{S}(6, 2, 0, 0)$ consists of 336 sequences. The possibility of 28 sequences per composition makes $(6, 2, 0, 0)$ more complex than, for example, $(7, 1, 0, 0)$ which has only 8 possible sequences per composition. There are, in total, 165 different compositions generated by 15 state vectors in $P_{8,4}$. Elementary considerations yield that all compositions with the same state vector have the identical number of possible sequences, and therefore have the same compositional complexity value.

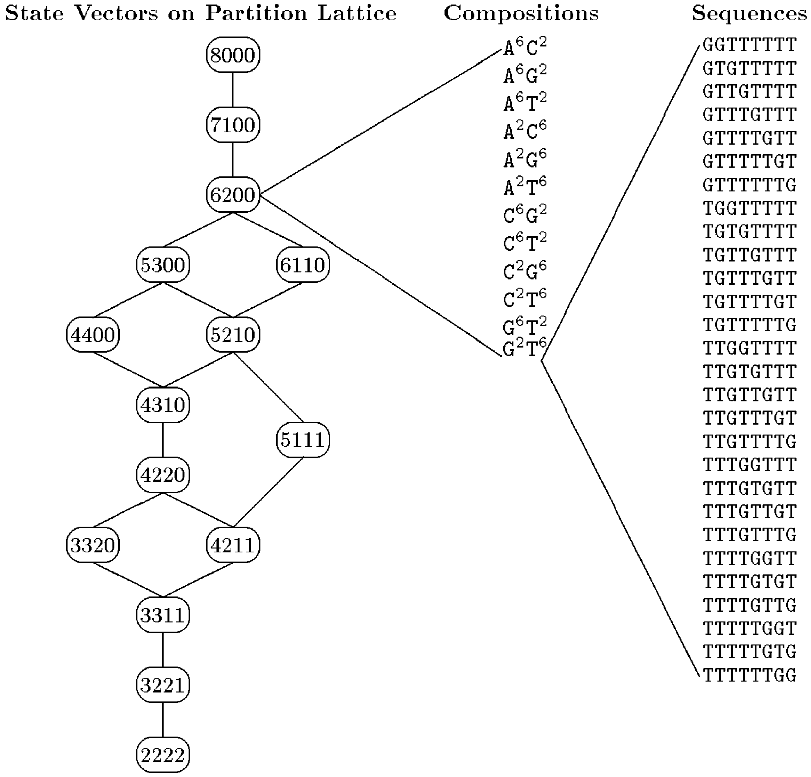

Fig. 1. The order diagram of $P_{8,4}$, compositions and sequences.

2.4. Covering chains and shifts

A subset C of $P_{l,n}$ is called a *chain* if any two elements in C are comparable, and is called an *antichain* if no two elements in C are comparable. The *size* of a chain C in $P_{l,n}$ is the number of elements in C. A chain $\mathbf{x}^{(1)} \prec \mathbf{x}^{(2)} \prec \cdots \prec \mathbf{x}^{(r)}$ in $P_{l,n}$ is called a *covering chain* or (a *saturated chain*) between $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(r)}$ if $\mathbf{x}^{(i+1)}$ covers $\mathbf{x}^{(i)}$ for all i with $1 \leqslant i \leqslant r-1$. For $\mathbf{x} \prec \mathbf{y}$ in $P_{l,n}$, we define the maximum size $M(\mathbf{x},\mathbf{y})$ and the minimum size $m(\mathbf{x},\mathbf{y})$ from $\mathbf{y}$ to $\mathbf{y}$ to be the sizes of the longest covering chain and of the shortest covering chain from $\mathbf{x}$ to $\mathbf{y}$, respectively. For example, $(3,3,1,1) \prec (4,2,1,1) \prec (5,2,1,0)$ is a chain in $P_{8,4}$, but not a covering chain. $(3,3,1,1) \prec (4,2,1,1) \prec (4,2,2,0) \prec (4,3,1,0) \prec (5,2,1,0)$ is both a covering chain and the longest chain. $\{(5,1,1,1),(3,3,2,0)\}$ is an antichain. The maximum size from $(3,3,1,1)$ to $(5,2,1,0)$ is 5 and the minimum size is 4.

A special type of subset of $P_{l,n}$ is the *closed interval* $[\mathbf{x},\mathbf{y}] = \{\mathbf{z} \in P_{l,n} | \mathbf{x} \prec \mathbf{z} \prec \mathbf{y}\}$, defined whenever $\mathbf{x} \prec \mathbf{y}$. We denote by $L_{\mathbf{x}}(\mathbf{y})$ the set of lower covers (children) of $\mathbf{y}$ in the interval $[\mathbf{x},\mathbf{y}]$. For example, $[(4,2,1,1),(5,2,1,0)] = \{(4,2,1,1),(4,2,2,0),(4,3,1,0),$

$(5,1,1,1),(5,2,1,0)\}$ in $P_{8,4}$.

$$L_{(4,2,1,1)}(5,2,1,0)=\{(4,3,1,0),(5,1,1,1)\}.$$

For $\mathbf{x}$ and $\mathbf{y}$ in $P_{l,n}$, by Proposition 2.1 $\mathbf{x}$ is obtained by a "unit transformation" of $\mathbf{y}$—shifting 1 from the ith component y_i to the jth component y_j of $\mathbf{y}$. Of course, the resulting vector $\mathbf{x}$ should be decreasing. In fact, there are two types of covers [10,11]. If $j=i+1$, i.e., $\mathbf{x}=\mathbf{y}-\mathbf{e}_i+\mathbf{e}_{i+1}$, then we call $\mathbf{y}\succ\mathbf{x}$ is a *nearby shift* (*N-shift*, in short). If $x_i=x_j$, i.e., $\mathbf{x}=\mathbf{y}-\mathbf{e}_i+\mathbf{e}_j$ and $x_i=\cdots=x_j$, then we call $\mathbf{y}\succ\mathbf{x}$ is a *distant shift* (*D-shift*, in short). Note that a cover $\mathbf{y}\succ\mathbf{x}$ can be both an N-shift and a D-shift (if $j=i+1$ and $x_i=x_j$), and in this case it is called an *ND*-shift. An N-shift (a D-shift) is *pure* if it is not a D-shift (an N-shift) [11,15]. In the partition lattice $P_{8,4}$, for example, $(5,2,1,0)\succ(4,3,1,0)$ is a pure N-shift, $(5,2,1,0)\succ(5,1,1,1)$ is a pure D-shift, and $(4,2,1,1)\succ(3,3,1,1)$ is an *ND*-shift.

Now we define a linear order-*lexicographic order* (*dictionary order*) on $P_{l,n}$. For two elements $\mathbf{x}=(x_1,x_2,\dots,x_n)$ and $\mathbf{y}=(y_1,y_2,\dots,y_n)$ in $P_{l,n}$, define $\mathbf{x}<\mathbf{y}$, if there is some k with $1\leqslant k\leqslant n$, such that $x_j=y_j$ for $1\leqslant j<k$, but $x_k<y_k$. We now define two typical chains between two comparable elements $\mathbf{x}\prec\mathbf{y}$ as follows.

Definition 2.1. For a covering pair $\mathbf{x}\prec\mathbf{y}$ in $P_{l,n}$ where $\mathbf{x}$ is not a lower cover (child) of $\mathbf{y}$, we write $\mathbf{y}=\mathbf{y}^{(1)}$ and first select a $\mathbf{y}^{(2)}\in L_{\mathbf{x}}(\mathbf{y}^{(1)})$. Then, we continue to choose a $\mathbf{y}^{(i+1)}\in L_{\mathbf{x}}(\mathbf{y}^{(i)})$ (i.e., take a child of $\mathbf{y}^{(i)}$ in the interval $[\mathbf{x},\mathbf{y}]$) until $\mathbf{y}^{(r)}=\mathbf{x}$. Thus, we obtain a covering chain between $\mathbf{y}$ and $\mathbf{x}$:

$$\mathbf{y}=\mathbf{y}^{(1)}\succ\mathbf{y}^{(2)}\succ\cdots\succ\mathbf{y}^{(r)}=\mathbf{x}. \tag{2}$$

If each $\mathbf{y}^{(i+1)}$ is the maximal element (largest child) in $L_{\mathbf{x}}(\mathbf{y}^{(i)})$ under the lexicographic ordering $(i=1,2,\dots,r-1)$, then the chain (2) is called the *lexicographic chain* (*L-chain*, in short) between $\mathbf{y}$ and $\mathbf{x}$. If each $\mathbf{y}^{(i+1)}$ is the minimal element (smallest child) in $L_{\mathbf{x}}(\mathbf{y}^{(i)})$ under the lexicographic ordering $(i=1,2,\dots,r-1)$, then the chain (2) is called the *counter-lexicographic chain* (*CL-chain*, in short) between $\mathbf{y}$ and $\mathbf{x}$.

The sizes of the L-chain and of the CL-chain between $\mathbf{x}$ and $\mathbf{y}$ are called the *lexicographic size* and the *counter-lexicographic size* between $\mathbf{x}$ and $\mathbf{y}$, respectively.

2.5. Measures based on sizes of chains

For a sequence s in the $\mathbf{v}$-equivalence class $\mathscr{S}(\mathbf{v})$, we have presented four new complexity functions in [19] based on the sizes of the following four typical chains between a comparable pair in the partition lattice $P_{l,n}$: (i) the longest chain; (ii) the shortest chain; (iii) the lexicographic chain; (iv) the counter-lexicographic chain.

Generally, complexity functions are defined as the proportion of the size of a chain in $P_{l,n}$ between the state vector $\mathbf{v}$ of s and the top element $\mathbf{x}^{(l)}_{\text{top}}$ to that of a chain between the top element $\mathbf{x}^{(l)}_{\text{top}}$ and the bottom element $\mathbf{x}^{(l)}_{\text{bot}}$ which passes through $\mathbf{v}$.

Definition 2.2. The *global compositional complexity of type I, II, III, and* IV of s are, respectively, defined as

$$G_1(s) = \frac{M(\mathbf{v}, \mathbf{x}^{(l)}_{\text{top}})}{M(\mathbf{x}^{(l)}_{\text{bot}}, \mathbf{v}) + M(\mathbf{v}, \mathbf{x}^{(l)}_{\text{top}})},$$

$$G_2(s) = \frac{m(\mathbf{v}, \mathbf{x}^{(l)}_{\text{top}})}{m(\mathbf{x}^{(l)}_{\text{bot}}, \mathbf{v}) + m(\mathbf{v}, \mathbf{x}^{(l)}_{\text{top}})},$$

$$G_3(s) = \frac{F(\mathbf{v}, \mathbf{x}^{(l)}_{\text{top}})}{F(\mathbf{x}^{(l)}_{\text{bot}}, \mathbf{v}) + F(\mathbf{v}, \mathbf{x}^{(l)}_{\text{top}})},$$

$$G_4(s) = \frac{f(\mathbf{v}, \mathbf{x}^{(l)}_{\text{top}})}{f(\mathbf{x}^{(l)}_{\text{bot}}, \mathbf{v}) + f(\mathbf{v}, \mathbf{x}^{(l)}_{\text{top}})},$$

where $M(\mathbf{x},\mathbf{y}), m(\mathbf{x},\mathbf{y}), F(\mathbf{x},\mathbf{y})$, and $f(\mathbf{x},\mathbf{y})$ denote the maximum-size, minimum-size, counter-lexicographic size, and lexicographic size between $\mathbf{x}$ and $\mathbf{y}$, respectively.

It is not difficult to see that when $n=4$, i.e., for nucleotide sequences, there is a small difference among the values of $G_1(\mathbf{v}), G_2(\mathbf{v}), G_3(\mathbf{v})$, and $G_4(\mathbf{v})$ for $\mathbf{v} \in P_{l,4}$. $G_1(\mathbf{v})$ is almost identical with $G_3(\mathbf{v})$, and $G_2(\mathbf{v})$ is almost identical with $G_4(\mathbf{v})$ in $P_{l,4}$. For example, $G_1(\mathbf{v}) = G_3(\mathbf{v}) \approx G_2(\mathbf{v}) = G_4(\mathbf{v})$ for all $\mathbf{v} \in P_{8,4}$. However, for protein sequences in this case $n=20$, there is a big difference among the values of $G_1(\mathbf{v}), G_2(\mathbf{v}), G_3(\mathbf{v})$, and $G_4(\mathbf{v})$. Making a comparison among these four types of complexity values, we can gain different insights into compositional bias of protein sequences [22].

3. Algorithm to compute the maximum length between a comparable pair

We present in this section an efficient algorithm to find the maximum length of a comparable pair, which is the key to calculating global compositional complexity functions defined in Section 2.5.

3.1. ND-paths

For $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ in $P_{l,n}$, the *conjugate* of an element $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ in $P_{l,n}$ is the element $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$, where x_k^* denotes the number of x_i that are greater than or equal to the integer k [12,15]. An *N-path* is a series of N-shifts, and a *D-path* is a series of D-shifts [12,14]. In the partition lattice $P_{8,4}$, for example, $(5,2,1,0) \prec (5,3,0,0) \prec (6,2,0,0)$ is an N-path, while $(5,1,1,1) \prec (5,2,1,0)$ is a D-path. $(4,2,2,0)$ is the conjugate of $(3,3,1,1)$ and $(4,2,1,1)$ is the conjugate of itself.

A chain $\mathbf{y} = \mathbf{y}^{(0)} \succ \mathbf{y}^{(1)} \succ \cdots \succ \mathbf{y}^{(r-1)} \succ \mathbf{y}^{(r)} = \mathbf{x}$ is an *ND-path* if there exists an integer k with $0 \leqslant k \leqslant r$ such that $\mathbf{y}^{(0)} \succ \mathbf{y}^{(1)} \succ \cdots \succ \mathbf{y}^{(k)}$ is an N-path and $\mathbf{y}^{(k)} \succ \cdots \succ \mathbf{y}^{(r)}$

is a D-path. For example, $(5,1,1,1) \prec (5,2,1,0) \prec (5,3,0,0) \prec (6,2,0,0)$ is an ND-path in $P_{8,4}$.

Theorem 3.1. *If* $C = \{\mathbf{y} = \mathbf{y}^{(0)} \succ \mathbf{y}^{(1)} \succ \cdots \succ \mathbf{y}^{(r-1)} \succ \mathbf{y}^{(r)} = \mathbf{x}\}$ *is a covering chain in* $P_{l,n}$, *then there exists an ND-path of length* $\geqslant r$ *from* $\mathbf{y}$ *to* $\mathbf{x}$.

Proof. We use induction on r. First, we consider the chain $\mathbf{y}^{(0)} \succ \mathbf{y}^{(1)} \succ \mathbf{y}^{(2)}$. Assume that $\mathbf{y}^{(0)} \succ \mathbf{y}^{(1)}$ is a pure D-shift, and $\mathbf{y}^{(1)} \succ \mathbf{y}^{(2)}$ is a pure N-shift. We write $\mathbf{y}^{(i)} = (y_1^{(i)}, y_2^{(i)}, \ldots, y_n^{(i)})$, $i = 0, 1, \ldots, r$. First, we consider the chain $\mathbf{y}^{(0)} \succ \mathbf{y}^{(1)} \succ \mathbf{y}^{(2)}$. Thus, we have $\mathbf{y}^{(1)} = \mathbf{y}^{(0)} - \mathbf{e}_k + \mathbf{e}_t$, where $y_k^{(1)} = \cdots = y_t^{(1)}$, $t - k > 1$, and $\mathbf{y}^{(2)} = \mathbf{y}^{(1)} - \mathbf{e}_m + \mathbf{e}_{m+1}$, where $y_m^{(1)} - y_{m+1}^{(1)} > 2$.

It is impossible that $k \leqslant m < t$, since $\mathbf{y}^{(2)}$ is decreasing. We must have $m < k$ or $m \geqslant t$. If $m < k - 1$ or $m > t$, we construct a new element u in $P_{l,n}$: $\mathbf{z} = \mathbf{y}^{(0)} - \mathbf{e}_m + \mathbf{e}_{m+1}$. Then $C' = \{\mathbf{y}^{(0)} \succ \mathbf{z}\}$ is an N-path and $\mathbf{z} \succ \mathbf{y}^{(2)}$ is a D-shift. If $m = k - 1$, then

$$y_{k-1}^{(1)} - y_k^{(1)} = y_{k-1}^{(0)} - (y_k^{(0)} - 1) > 2$$

since $\mathbf{y}^{(1)} \succ \mathbf{y}^{(2)}$ is a pure N-shift. We produce two new elements $\mathbf{u}$ and $\mathbf{z}$ in $P_{l,n}$: $\mathbf{u} = \mathbf{y}^{(0)} - \mathbf{e}_{k-1} + \mathbf{e}_k$ and $\mathbf{z} = \mathbf{y}^{(0)} - \mathbf{e}_{k-1} + \mathbf{e}_{k+1}$. In addition, $\mathbf{y}^{(2)} = \mathbf{y}^{(0)} - \mathbf{e}_{k-1} + \mathbf{e}_t$. $C' = \{\mathbf{y}^{(0)} \succ \mathbf{u} \succ \mathbf{z}\}$ is an N-path, and $\mathbf{z} \succ \mathbf{y}^{(2)}$ is a D-shift. If $m = t$, then

$$y_t^{(1)} - y_{t+1}^{(1)} = (y_t^{(0)} + 1) - y_{t+1}^{(0)} > 2$$

since $\mathbf{y}^{(1)} \succ \mathbf{y}^{(2)}$ is a pure N-shift. We make two new elements $\mathbf{v}$ and $\mathbf{z}$ in $P_{l,n}$: $\mathbf{v} = \mathbf{y}^{(0)} - \mathbf{e}_t + \mathbf{e}_{t+1}$ and $\mathbf{z} = \mathbf{y}^{(0)} - \mathbf{e}_{t-1} + \mathbf{e}_{t+1}$. Moreover, $\mathbf{y}^{(2)} = \mathbf{y}^{(0)} - \mathbf{e}_k + \mathbf{e}_{t+1}$. $C' = \{\mathbf{y}^{(0)} \succ \mathbf{v} \succ \mathbf{z}\}$ is an N-path, and $\mathbf{z} \succ \mathbf{y}^{(2)}$ is a D-shift. By induction, we can construct an ND-path C'' of length $\geqslant r - 1$ between $\mathbf{y}$ and $\mathbf{z}$ based on the chain

$$\mathbf{z} \succ \mathbf{y}^{(2)} \succ \cdots \succ \mathbf{y}^{(r-1)} \succ \mathbf{y}^{(r)} = \mathbf{x}$$

of length $r - 1$. Merging C' and C'', we finally obtain an ND-path of length $\geqslant r$ from $\mathbf{x}$ to $\mathbf{y}$ based on the chain C. This completes the proof of the theorem. □

Throughout this paper, we use the following notation for a set of positive integers and a subvector defined by those integers: $[p,q] := \{p, p+1, \ldots, q\}$ and $\mathbf{v}[p,q] := (v_p, v_{p+1}, \ldots, v_q)$ for a vector $\mathbf{v} = (v_1, \ldots, v_n)$, which is called a *segment* of $\mathbf{v}$. For $\mathbf{x} = (x_1, x_2, \ldots, x_n) \prec \mathbf{y} = (y_1, y_2, \ldots, y_n)$ in $P_{l,n}$, let

$$\{t \mid \sigma_t(\mathbf{x}) = \sigma_t(\mathbf{y})\} = \{t_0 = 0 < t_1 < t_2 < \cdots < t_m\}.$$

Then $[t_{j-1} + 1, t_j]$ is called a *majorizing interval* of $\mathbf{x}$ and $\mathbf{y}$, $(j = 1, 2, \ldots, m)$ [10]. A segment $\mathbf{y}[p,q]$ of $\mathbf{y}$ is said to be *feasible* [15] if $y_i - y_{i+1} \leqslant 1, i = p, p+1, \ldots, q-1$. For example, there are two majorizing intervals of (5, 4, 3, 3, 0, 0) and (4, 4, 4, 1, 1, 1): [1, 3] and [4, 6]. $(5,4,3,3,0,0)[1,4] = (5,4,3,3)$ is a feasible segment of (5, 4, 3, 3, 0, 0).

3.2. N-realizablity and D-realizablity

For $\mathbf{x} \prec \mathbf{z} \prec \mathbf{y}$ in $P_{l,n}$, $\mathbf{z}$ is *N-realizable* from $\mathbf{y}$ if there exists an N-path from $\mathbf{y}$ to $\mathbf{z}$, and is said to be *downmost N-realizable* from $\mathbf{y}$ if in addition, for any N-realizable element $\mathbf{w}$ from $\mathbf{y}$, $\mathbf{w} \succ \mathbf{z}$. $\mathbf{x}$ is *D-realizable* from $\mathbf{z}$ if there exists a D-path from $\mathbf{z}$ to $\mathbf{x}$, and is called to be *upmost D-realizable* from $\mathbf{z}$ if in addition, for any N-realizable element $\mathbf{w}$ from $\mathbf{y}$, $\mathbf{w} \prec \mathbf{z}$.

Theorem 3.2. *If $\mathbf{x} = (x_1, x_2, \ldots, x_n) \prec \mathbf{y} = (y_1, y_2, \ldots, y_n)$ in $P_{l,n}$, then there exist the downmost N-realizable element $\mathbf{z}$ from $\mathbf{y}$ and the upmost D-realizable element $\mathbf{w}$ to $\mathbf{x}$ in $P_{l,n}$.*

Proof. To prove the theorem, we give an algorithm for constructing the downmost N-realizable element $\mathbf{z}$ from $\mathbf{y}$ toward $\mathbf{x}$ in $P_{l,n}$. Roughly, we start from finding the leftmost N-shift between $\mathbf{y}$ and $\mathbf{x}$ and construct a new element $\mathbf{y}^{(1)}$ in $P_{l,n}$ based on this N-shift, then find the leftmost N-shift between $\mathbf{y}^{(1)}$ and $\mathbf{x}$ and use this N-shift to make a new element $\mathbf{y}^{(2)}$ in $P_{l,n}$, continue toward $\mathbf{x}$ by the leftmost N-shifts until no further N-shifts are available. □

Algorithm 3.1. Given two elements $\mathbf{x} = (x_1, x_2, \ldots, x_n) \prec \mathbf{y} = (y_1, y_2, \ldots, y_n)$ in $P_{l,n}$, construct the downmost N-realizable element $\mathbf{z}$ from $\mathbf{y}$ in $P_{l,n}$.

Step 1: Find the first positive difference $y_k - x_k$ for k in $[1, n]$ and the smallest index t for which $y_t - y_{t+1} \geqslant 2$ with $t \geqslant k$.

Step 2: Check if $[k, t]$ is not a majorizing interval of $\mathbf{x}$ and $\mathbf{y}$. If not, we construct a new element in $P_{l,n}$ as follows: $\mathbf{y}^{(1)} = \mathbf{y} - \mathbf{e}_t + \mathbf{e}_{t+1}$. If so, go to Step 1 to look for new k using $[t+1, n]$ instead of $[1, n]$, and then find new t with $t \geqslant k$.

Step 3: Check if $k = n - 1$. If so, stop and output $\mathbf{z} = \mathbf{y}^{(1)}$. Otherwise, go to Step 1 substituting $\mathbf{z}$ for $\mathbf{y}$.

Proof of Theorem 3.2 (continued). We first prove that $\mathbf{z} \prec \mathbf{y}^{(1)}$. By definition, there exists an N-path $\mathbf{y} \succ \mathbf{z}^{(1)} \succ \cdots \succ \mathbf{z}^{(p)} \succ \mathbf{z}$ from $\mathbf{y}$ to $\mathbf{z}$. Since $\mathbf{x}[1, k-1] = \mathbf{y}[1, k-1]$, we must have that $\mathbf{x}[1, k-1] = \mathbf{z}[1, k-1] = \mathbf{y}[1, k-1]$. Note that $\mathbf{y}[k, t]$ is a feasible segment of $\mathbf{y}$. There does not exist any N-shift among the components of $\mathbf{y}[k, t]$. Thus, $\mathbf{x}[1, t] = \mathbf{z}[1, t] = \mathbf{y}[1, t]$. Since $y_t - y_{t+1} \geqslant 2$, $\mathbf{y}^{(1)}$ is decreasing. We distinguish two cases.

Case 1: $[k, t]$ is not a majorizing interval of $\mathbf{x}$ and $\mathbf{y}$. It is easy to verify in this case that $\mathbf{x} \prec \mathbf{y}^{(1)} \prec \mathbf{y}$. Because t is the smallest index of components in $\mathbf{y}$ from which we can do an N-shift, we should have that $\mathbf{z}^{(1)} \prec \mathbf{y}^{(1)}$. Thus, $\mathbf{z} \prec \mathbf{y}^{(1)}$.

Case 2: $[k, t]$ is a majorizing interval of $\mathbf{x}$ and $\mathbf{y}$. We can continue to search new k in $[t+1, n]$ and then find new t with $t \geqslant k$. It can be dealt with as did in Case 1. Clearly, we can construct $\mathbf{y}^{(1)}$ in at most $n - 1$ stages.

It is not difficult to see that $\mathbf{y}^{(1)}$ is completely determined by both $\mathbf{x}$ and $\mathbf{y}$. If $\mathbf{z} \neq \mathbf{y}^{(1)}$, we can use an N-shift to construct in a similar fashion a vector $\mathbf{y}^{(2)}$ from $\mathbf{y}^{(1)}$ such that $\mathbf{z} \prec \mathbf{y}^{(2)}$. Like $\mathbf{y}^{(1)}$, $\mathbf{y}^{(2)}$ is uniquely decided by $\mathbf{x}$ and $\mathbf{y}$. If $\mathbf{z} \neq \mathbf{y}^{(2)}$, repeating such a construction step, we obtain, in a finite number of steps, say s ($s < n$), an N-path:

$$\mathbf{y} \succ \mathbf{y}^{(1)} \succ \mathbf{y}^{(2)} \succ \cdots \succ \mathbf{y}^{(s)} = \mathbf{z},$$

where all of $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(s)}$ are totally determined by $\mathbf{x}$ and $\mathbf{y}$. This shows the uniqueness of the downmost N-realizable element from $\mathbf{y}$ toward $\mathbf{x}$, which can be constructed by using Algorithm 3.1.

In a similar way, we can prove the following:

Theorem 3.3. *If $\mathbf{z}$ is N-realizable from $\mathbf{y}$ and $\mathbf{z} \prec \mathbf{w}$ is an N-shift where $\mathbf{w} \prec \mathbf{y}$, then $\mathbf{w}$ is also N-realizable from $\mathbf{y}$. Dually, if $\mathbf{z}$ is D-realizable from $\mathbf{x}$ and $\mathbf{z} \succ \mathbf{w}$ is an D-shift where $\mathbf{w} \succ \mathbf{x}$, then $\mathbf{w}$ is also D-realizable to $\mathbf{x}$.*

We denote by $\underline{\mathbf{y}}_{\mathbf{x}}$ the downmost N-realizable element in the interval $\mathbf{x} \prec \mathbf{y}$ of $P_{l,n}$. $\underline{\mathbf{y}}_{\mathbf{x}}$ is called the "*turning point*". The following theorem will play crucial roles in the proof of our main result.

Theorem 3.4. *Let $\mathbf{y} \succ \mathbf{w} \succ \mathbf{x}$ in $P_{l,n}$ where $\mathbf{w}$ is D-realizable to $\mathbf{x}$ and is N-realizable from $\mathbf{y}$. If $\mathbf{w}$ is not downmost N-realizable from $\mathbf{y}$, then there exists a D-realizable element $\mathbf{v}$ to $\mathbf{x}$ with $\mathbf{w} \succ \mathbf{v} \succ \mathbf{x}$ such that $\mathbf{w} \succ \mathbf{v}$ is an ND-shift.*

Proof. It follows from D-realizability of $\mathbf{w}$ to $\mathbf{x}$ that there exists a D-path from $\mathbf{w}$ to $\mathbf{x}$: $\mathbf{w} = \mathbf{w}^{(0)} \succ \mathbf{w}^{(1)} \succ \cdots \succ \mathbf{w}^{(t)} = \mathbf{x}$. Since $\mathbf{w}$ is not downmost N-realizable from $\mathbf{y}$, there is an element $\mathbf{v}$ with $\mathbf{w} \succ \mathbf{v} \succ \mathbf{x}$ such that $\mathbf{w} \succ \mathbf{v}$ is an N-shift: $\mathbf{v} = \mathbf{w} - \mathbf{e}_k + \mathbf{e}_{k+1}$. Thus,

$$\sigma_k(\mathbf{x}) \leqslant \sigma_k(\mathbf{v}) = \sigma_k(\mathbf{w}) - 1. \tag{3}$$

Note that each $\mathbf{w}^{(i)} \succ \mathbf{w}^{(i+1)}$ is a D-shift. If $\mathbf{w} \succ \mathbf{v}$ is not an ND-shift, then there should be no D-shifts between the segments $\mathbf{w}[1,k]$ and $\mathbf{w}[k+1,n]$. Consequently, we must have that $\sigma_k(\mathbf{w}) = \sigma_k(\mathbf{w}^{(t-1)}) = \cdots = \sigma_k(\mathbf{w}^{(1)}) = \sigma_k(\mathbf{x})$, which contradicts inequality (3). This means that $\mathbf{w} \succ \mathbf{v}$ is an ND-shift. By Theorem 3.3 and D-realizability of $\mathbf{w}$ to $\mathbf{x}$, v is D-realizable from $\mathbf{x}$. This completes the proof. □

3.3. The maximum length function

Now we state and prove our main result in this section, which gives a characterization of the maximum length function $M(\mathbf{x}, \mathbf{y})$ in $P_{l,n}$. We start with two definitions. By Proposition 3.1, a cover $\mathbf{y} \succ \mathbf{x}$ in $P_{l,n}$ is an N-shift if and only if $\sigma(\mathbf{x}) - \sigma(\mathbf{y})$ is a unit vector, which is equivalent to the following:

$$\sum_{i=1}^{n} (\sigma_i(\mathbf{x}) - \sigma_i(\mathbf{y})) = \sum_{i=1}^{n} (i-1)(x_i - y_i) = 1.$$

For convenience, we define the *N-function* $f_N(\mathbf{x})$ of an element $\mathbf{x}$ in $P_{l,n}$ as

$$f_N(\mathbf{x}) = \sum_{i=1}^{n} (i-1)x_i.$$

Obviously, a cover $\mathbf{y} \succ \mathbf{x}$ in $P_{l,n}$ is an N-shift if and only if $f_N(\mathbf{x}) - f_N(\mathbf{y}) = 1$. Thus, all N-paths between $\mathbf{y}$ and $\mathbf{x}$ (if one or more exist) in $P_{l,n}$ have the same length: $M(\mathbf{x}, \mathbf{y}) = f_N(\mathbf{x}) - f_N(\mathbf{y})$.

Similarly, we define the *D-function* $f_D(\mathbf{x})$ as

$$f_D(\mathbf{x}) = \sum_{k=1}^{n} (k-1)x_k^*,$$

where $\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_n^*)$ is the conjugate of $\mathbf{x}$. A cover $\mathbf{y} \succ \mathbf{x}$ in $P_{l,n}$ is a D-shift if and only if $f_D(\mathbf{y}) - f_D(\mathbf{x}) = 1$. All D-paths between $\mathbf{y}$ and $\mathbf{x}$ (if one or more exist) in $P_{l,n}$ have the same length: $M(\mathbf{x},\mathbf{y}) = f_D(\mathbf{y}) - f_D(\mathbf{x})$.

Theorem 3.5. *If* $\mathbf{x} \prec \mathbf{y}$ *in* $P_{l,n}$, *then all ND-paths from* $\mathbf{y}$ *to* $\mathbf{x}$ *have the maximum length*:

$$M(\mathbf{x},\mathbf{y}) = f_N(\underline{\mathbf{y}}_{\mathbf{x}}) - f_N(\mathbf{y}) + f_D(\underline{\mathbf{y}}_{\mathbf{x}}) - f_D(\mathbf{x}) \tag{4}$$

Proof. By Theorem 3.1, there exists an *ND*-path which is a maximum-length chain between $\mathbf{x}$ and $\mathbf{y}$. Let $C = \{\mathbf{y} = \mathbf{y}^{(0)} \succ \mathbf{y}^{(1)} \succ \cdots \succ \mathbf{y}^{(r-1)} \succ \mathbf{y}^{(r)} = \mathbf{x}\}$ be such an *ND*-path, where $\mathbf{y}^{(0)} \succ \mathbf{y}^{(1)} \succ \cdots \succ \mathbf{y}^{(t)}$ is an N-path and $\mathbf{y}^{(t)} \succ \cdots \succ \mathbf{y}^{(r)}$ is a D-path. By the definition of $\underline{\mathbf{y}}_{\mathbf{x}}$, we must have $\underline{\mathbf{y}}_{\mathbf{x}} \prec \mathbf{y}^{(t)}$. By Theorem 3.4, if $\mathbf{y}^{(t)} \neq \underline{\mathbf{y}}_{\mathbf{x}}$, then there exists a D-realizable element $\mathbf{w}^{(1)}$ to $\mathbf{x}$ such that $\mathbf{y}^{(t)} \succ \mathbf{w}^{(1)} \succ \mathbf{x}$, and $\mathbf{y}^{(t)} \succ \mathbf{w}^{(1)}$ is an *ND*-shift. By induction on $M(\underline{\mathbf{y}}_{\mathbf{x}}, \mathbf{y}^{(t)})$, we con construct a chain: $\mathbf{y}^{(t)} \succ \mathbf{w}^{(1)} \succ \cdots \succ \mathbf{w}^{(p-1)} \succ \mathbf{w}^{(p)} = \underline{\mathbf{y}}_{\mathbf{x}}$, which is both a D-path and an N-path. Hence,

$$p = f_D(\mathbf{y}^{(t)}) - f_D(\underline{\mathbf{y}}_{\mathbf{x}}) = f_N(\underline{\mathbf{y}}_{\mathbf{x}}) - f_N(\mathbf{y}^{(t)}).$$

Noting that any two D-paths from $\mathbf{y}^{(t)}$ to $\mathbf{x}$ have the same length, we obtain

$$\begin{aligned} f_D(\mathbf{y}^{(t)}) - f_D(\mathbf{x}) &= p + (f_D(\underline{\mathbf{y}}_{\mathbf{x}}) - f_D(\mathbf{x})) \\ &= f_N(\underline{\mathbf{y}}_{\mathbf{x}}) - f_N(\mathbf{y}^{(t)}) + f_D(\underline{\mathbf{y}}_{\mathbf{x}}) - f_D(\mathbf{x}). \end{aligned}$$

Thus, the length of C is

$$\begin{aligned} M(\mathbf{x},\mathbf{y}) &= f_N(\mathbf{y}^{(t)}) - f_N(\mathbf{y}) + f_D(\mathbf{y}^{(t)}) - f_D(\mathbf{x}) \\ &= f_N(\underline{\mathbf{y}}_{\mathbf{x}}) - f_N(\mathbf{y}) + f_D(\underline{\mathbf{y}}_{\mathbf{x}}) - f_D(\mathbf{x}). \end{aligned}$$

This completes the proof. □

3.4. Algorithm design

Based on Theorem 3.5, we have the following incremental algorithm for constructing a maximum-length chain between $\mathbf{x} \prec \mathbf{y}$ in $P_{l,n}$.

Algorithm 3.2. Given two elements $\mathbf{x} = (x_1, x_2, \dots, x_n) \prec \mathbf{y} = (y_1, y_2, \dots, y_n)$ in $P_{l,n}$, construct an *ND*-path between $\mathbf{y}$ and $\mathbf{x}$, and calculate the maximum length $M(\mathbf{x},\mathbf{y})$.

Step 1: Begin from $\mathbf{y}$ and continue toward $\mathbf{x}$ by leftmost N-shifts, according to Algorithm 3.1, until no further N-shifts are available. This appears whenever the "turning point" $\underline{\mathbf{y}}_{\mathbf{x}}$ is reached. We obtain an N-path:

$$\mathbf{y} = \mathbf{y}^{(0)} \succ \mathbf{y}^{(1)} \succ \cdots \succ \mathbf{y}^{(t)} = \underline{\mathbf{y}}_{\mathbf{x}}.$$

Step 2: Continue from $\underline{\mathbf{y}}_{\mathbf{x}}$ to $\mathbf{x}$ by leftmost D-shifts. We get a D-path:

$$\underline{\mathbf{y}}_{\mathbf{x}} = \mathbf{y}^{(t)} \succ \mathbf{y}^{(t+1)} \succ \cdots \succ \mathbf{y}^{(r)} = \mathbf{x},$$

and thus we obtain an ND-path from $\mathbf{y}$ to $\mathbf{x}$:

$$\mathbf{y} = \mathbf{y}^{(0)} \succ \mathbf{y}^{(1)} \succ \cdots \succ \underline{\mathbf{y}}_{\mathbf{x}} \cdots \succ \mathbf{y}^{(r-1)} \succ \mathbf{y}^{(r)} = \mathbf{x}.$$

Step 3: Compute the value of $M(\mathbf{x},\mathbf{y})$ by the formula (4).

For example, to construct an ND-path between $(4,3,0,0)$ and $(2,2,2,1)$ and to calculate the maximum length $M((2,2,2,1),\ (4,3,0,0))$ in the partition lattice $P_{8,4}$, we firstly produce an N-path starting from $(4,3,0,0)$:

$$(4,3,0,0) \succ (4,2,1,0) \succ (3,3,1,0) \succ (3,2,2,0) \succ (3,2,1,1),$$

where $(3,2,1,1)$ is the downmost N-realizable element (turning point) from $(4,3,0,0)$. Then, we make a D-path starting from $(3,2,1,1)$:

$$(3,2,1,1) \succ (2,2,2,1).$$

Finally, we obtain an ND-path from $(4,3,0,0)$ to $(2,2,2,1)$:

$$(4,3,0,0) \succ (4,2,1,0) \succ (3,3,1,0) \succ (3,2,2,0) \succ (3,2,1,1) \succ (2,2,2,1),$$

and calculate the maximum length:

$$M((2,2,2,1),(4,3,0,0)) = 5.$$

We now turn to computational complexity analysis of Algorithms 3.1 and 3.2. Note that for a given alphabet $\mathbf{A}$, n is fixed (usually, $n=4$ or 20) in biological applications. Step 1 and 2 in Algorithm 3.2 for finding the leftmost N-shift require O(1) time and space. To analyze Step 3 and the loop in Algorithm 3.2, we consider the worst case which occurs when $\mathbf{y}=X^{(l)}_{\max}$ and $\mathbf{x}=X^{(l)}_{\min}$. Let X denote the turning point (downmost N-realizable element) in the interval $X^{(l)}_{\min} \prec X^{(l)}_{\max}$. Step 3 and the loop in Algorithm 3.2 for constructing X requires $f_N(X)$ stages. We will find a simple expression for $f_N(X)$. To this end, we write

$$l = pn + \frac{q(q+1)}{2} + t,$$

where p, q, and t are nonnegative integers, and $0 \leqslant t \leqslant q < n$. Then

$$X = (p,p,\ldots,p) + (q,q-1,\ldots,t,t,t-1,\ldots,2,1,0,\ldots,0).$$

For example, the turning point in $P_{7,4}$ is $(3,2,2,1)$ and we have $p=1$, $q=2$, and $t=1$ in this case.

By an elementary calculation, we have

$$f_N(X) = \tfrac{1}{2}pn(n-1) - \tfrac{1}{2}(q-t)(q-t+1) - \tfrac{1}{3}q(q-1)(q-2) \,<\, \tfrac{1}{2}nl.$$

Therefore, we obtain the following:

Theorem 3.6. *Algorithm* 3.2 *requires* O(l) *time and space to calculate the maximum length between a comparable pair in* $P_{l,n}$.

Similarly, we have the following theorem:

Theorem 3.7. *Algorithm* 3.2 *requires* O(l) *time and space to construct a maximum-length chain between a comparable pair in* $P_{l,n}$.

Recall that the determination of the state vector of a sequence S in $\mathbf{A}^l$ requires O(l) time and space. Thus, Algorithm 3.2 is fully efficient in the sense that the computation of the maximum length is of the same order of computational complexity O(l) as the determination of the state vector.

4. Sizes of covering chains

As mentioned in Section 2.5, there are four typical chains between a comparable pair in the partition lattice $P_{l,n}$: the longest chain, the shortest chain, the lexicographic chain, and the counter-lexicographic chain. We have developed an algorithm for calculating the size of the longest chain in Section 3. In this section, we will present efficient algorithms for computing the sizes of other three types of chains.

4.1. Maximum-size chain and minimum-size chain

For $\mathbf{x} \prec \mathbf{y} \in P_{l,n}$, using the leftmost pure N-shift first search, Algorithm 3.2 can be compactly rewritten as the following two-step algorithm to find the maximum size from $\mathbf{y}$ to $\mathbf{x}$.

Algorithm 4.1. Given two elements $\mathbf{x} \prec \mathbf{y} = \mathbf{y}^{(1)}$ in $P_{l,n}$, calculate the maximum size $M(\mathbf{x},\mathbf{y})$.

Step 1: Check if there exists an N-shift in $L_{\mathbf{x}}(\mathbf{y}^{(1)})$. If so, find the leftmost N-shift in $L_{\mathbf{x}}(\mathbf{y}^{(1)})$ and construct a new element $\mathbf{y}^{(2)}$ in $L_{\mathbf{x}}(\mathbf{y}^{(1)})$ based on this N-shift. Otherwise, find the leftmost D-shift in $L_{\mathbf{x}}(\mathbf{y}^{(1)})$ and construct a new element $\mathbf{y}^{(2)}$ in $L_{\mathbf{x}}(\mathbf{y}^{(1)})$ based on this D-shift.

Step 2: Check if $\mathbf{y}^{(2)} = \mathbf{x}$. If so, stop and output $M(\mathbf{x},\mathbf{y}) = 2$. Otherwise, go to Step 1 and continue toward $\mathbf{x}$ by Step 1 until $\mathbf{y}^{(r)} = \mathbf{x}$. Then, stop and output $M(\mathbf{x},\mathbf{y}) = r$.

Similarly, using the rightmost pure D-shift first search, we can devise an algorithm to find the minimum size of a comparable pair in $P_{l,n}$.

Algorithm 4.2. Given two elements $\mathbf{x} \prec \mathbf{y} = \mathbf{y}^{(1)}$ in $P_{l,n}$, calculate the minimum size $m(\mathbf{x},\mathbf{y})$.

Step 1: Check if there exists a pure D-shift in $L_{\mathbf{x}}(\mathbf{y}^{(1)})$. If so, find the rightmost pure D-shift in $L_{\mathbf{x}}(\mathbf{y})$ and construct a new element $\mathbf{y}^{(2)}$ in $L_{\mathbf{x}}(\mathbf{y}^{(1)})$ based on this D-shift. Otherwise, find the rightmost N-shift in $L_{\mathbf{x}}(\mathbf{y})$ and construct a new element $\mathbf{y}^{(2)}$ in $L_{\mathbf{x}}(\mathbf{y}^{(1)})$ based on this N-shift.

Step 2: Check if $\mathbf{y}^{(2)} = \mathbf{x}$. If so, stop and output $m(\mathbf{x}, \mathbf{y}) = 2$. Otherwise, go to Step 1 and continue toward $\mathbf{x}$ by Step 1 until $\mathbf{y}^{(r)} = \mathbf{x}$. Then, stop and output $m(\mathbf{x}, \mathbf{y}) = r$.

We have showed in the previous section that there is a "turning point" in any longest chain between $\mathbf{x}$ and $\mathbf{y}$ generated by Algorithm 3.2 or Algorithm 4.1. However, there does not exist in general an intermediate point, similar to the "turning point", in the shortest chain between $\mathbf{x}$ and $\mathbf{y}$ generated by Algorithm 4.2.

4.2. L-chain and CL-chain

It is easy to find the relationship among the L-chain, the CL-chain, and shifts. In fact, the rightmost shift in $\mathbf{y}$ toward $\mathbf{x}$ produces the the largest child of $\mathbf{y}$, while the leftmost shift in $\mathbf{y}$ toward $\mathbf{x}$ always makes the the smallest child of $\mathbf{y}$. Thus, using the rightmost shift-first search and the leftmost shift-first search, we have two algorithms for computing the lexicographic size and the counter-lexicographic size between a comparable pair in the partition lattice, respectively.

Algorithm 4.3. Given two elements $\mathbf{x} \prec \mathbf{y} = \mathbf{y}^{(1)}$ in $P_{l,n}$, calculate the lexicographic size $F(\mathbf{x}, \mathbf{y})$.

Step 1: Find the rightmost shift (N-shift or D-shift) in $L_{\mathbf{x}}(\mathbf{y}^{(1)})$ and construct a new element $\mathbf{y}^{(2)}$ in $L_{\mathbf{x}}(\mathbf{y}^{(1)})$ based on such shift.

Step 2: Check if $\mathbf{y}^{(2)} = \mathbf{x}$. If so, stop and output $f(\mathbf{x}, \mathbf{y}) = 2$. Otherwise, go to Step 1 and continue toward $\mathbf{x}$ by Step 1 until $\mathbf{y}^{(r)} = \mathbf{x}$. Then, stop and output $f(\mathbf{x}, \mathbf{y}) = r$.

Algorithm 4.4. Given two elements $\mathbf{x} \prec \mathbf{y} = \mathbf{y}^{(1)}$ in $P_{l,n}$, calculate the counter-lexicographic size $f(\mathbf{x}, \mathbf{y})$.

Step 1: Find the leftmost shift (N-shift or D-shift) in $L_{\mathbf{x}}(\mathbf{y}^{(1)})$ and construct a new element $\mathbf{y}^{(2)}$ in $L_{\mathbf{x}}(\mathbf{y}^{(1)})$ based on such shift.

Step 2: Check if $\mathbf{y}^{(2)} = \mathbf{x}$. If so, stop and output $F(\mathbf{x}, \mathbf{y}) = 2$. Otherwise, go to Step 1 and continue toward $\mathbf{x}$ by Step 1 until $\mathbf{y}^{(r)} = \mathbf{x}$. Then, stop and output $F(\mathbf{x}, \mathbf{y}) = r$.

4.3. Computational complexity analysis

For a comparable pair $\mathbf{x} \prec \mathbf{y}$ in $P_{l,n}$, we have

$$m(\mathbf{x}, \mathbf{y}) \leqslant f(\mathbf{x}, \mathbf{y}) \leqslant F(\mathbf{x}, \mathbf{y}) \leqslant M(\mathbf{x}, \mathbf{y}).$$

Obviously, there exists a unique L-chain and a unique CL-chain between $\mathbf{x}$ and $\mathbf{y}$. There may be many maximum-size chains and many minimum-size chains between $\mathbf{x}$ and $\mathbf{y}$, particularly for larger n and l.

We now turn to computational complexity analysis of Algorithms 4.1–4.4. In a similar way as in the proof of linearity of Algorithm 3.2 in the previous section, we can prove the following theorem:

Theorem 4.1. *Algorithms* 4.1–4.4 *run in* $\mathrm{O}(l)$ *time and space for a fixed n.*

Algorithms 4.1–4.4 are very efficient, and the best possible algorithms in the sense that $\mathrm{O}(l)$ is a lower bound for any algorithm which solves these problems. Note that the order digram of $P_{l,n}$ is a circle-free and triangle-free directed graph whose vertex number is $p_n(l)$. Little information is know about finding the longest path between two vertices in a general directed graph. Traditionally the shortest path problem has received most attention. In fact, many algorithms have been proposed that compute the shortest path information probabilistically fast, over a large class of directed graphs. The best computational complexity bound so far is $O(v^2 \log v)$ expected time, due to Moffat and Takaoka, where v is the number of vertices of the given directed graph. If we directly apply Moffat and Takaoka's graph algorithm to calculating the minimum-size of a comparable pair in $P_{l,n}$, it runs in time $O(p_n^2(l)\log(p_n(l)))$. According to the asymptotic estimate formula (3), Moffat and Takaoka's shortest path algorithm in this case requires $O((l+n)^{2(n-1)}\log(l+n))$ time, whose computational complexity is much higher than that of our Algorithm 4.2.

Acknowledgements

This work was supported by the grant A9710660 from National Research Council.

References

[1] S.F. Altschul, M.S. Boguski, W. Gish, J.C. Wootton, Issues in searching molecular sequence databases, Natur. Genet. 6 (1994) 119–129.

[2] G. Andrews, The Theory of Partitions, Encyclopedia of Mathematics and its Applications, Vol. 2, Addison-Wesley, Reading, MA, 1976.

[3] G.I. Bell, Evolution of simple sequence repeats, Comput. Chem. 20 (1996) 41–48.

[4] J.B. Bryngelson, P.G. Wolynes, Spin glasses and the statistical mechanics of protein folding, Proc. Natl. Acad. Sci. USA 84 (1987) 7524–7528.

[5] A.V. Finkelstein, Implications of the random characteristics of protein sequences for their three-dimensional structure, Curr. Opin. Struct. Biol. 4 (1994) 422–428.

[6] G.H. Hardy, J.E. Littlewood, G. Polya, Inequalities, Cambridge University Press, London, New York, 1952.

[7] S. Karlin, V. Brendel, Chance and statistical significance in protein and DNA sequence analysis, Science 257 (1992) 39–49.

[8] O.B. Ptitsyn, Protein as an 'edited' statistical copolymer?, in: R. Srinivasan, R.M. Sarma (Eds.), Conformation in Biology, Academic Press, New York, 1983, pp. 49–58.

[9] E.I. Shakhnovitch, A.M. Gutin, Formation of unique structure in polypeptide chains: theoretical investigation with the aid of a replica approach, Biophys. Chem. 34 (1989) 187–199.

[10] H. Wan, Structure and cardinality of the class $A(R,S)$ of $(0,1)$-matrices, J. Math. Res. Exposition 4 (1984) 87–93.

[11] H. Wan, $(0,1)$-matrices class with prescribed row and column sums and integral partition lattices, Master's Thesis, Huazhong, Central China, University of Science and Technology, 1984.

[12] H. Wan, Cardinal function of the class $A(R,S)$ of $(0,1)$-matrices and its nonzero-point set, J. Math. Res. Exposition 5 (1985) 117–120.

[13] H. Wan, Combinatorial and Computing Theory of Nonnegative Integral Matrices, Dalian University of Technology Press, Dalian, 1986.

[14] H. Wan, On the structure and enumeration of $(0,1)$-matrices, Acta Math. Sinica 30 (1987) 289–302.

[15] H. Wan, On nearly self-conjugate partition of a finite set, Discrete Math. 175 (1997) 239–247.

[16] H. Wan, Q. Li, On the number of tournaments with prescribed score vector, Discrete Math. 61 (1986) 213–219.

[17] H. Wan, E. Song, Quasi-periods in biological sequences, Theoret. Comput. Sci., submitted for publication.

[18] H. Wan, J.C. Wootton, Sequence complexities and symmetries deduced from partition lattices and self-difference matrices, manuscript, 1997.

[19] H. Wan, J.C. Wootton, Axiomatic foundations of complexity functions of biological sequences, Ann. Combin. 3 (1999) 105–127.

[20] H. Wan, J.C. Wootton, A global compositional complexity measure for biological sequences: AT-rich and GC-rich genomes encode less complex proteins, Comput. Chem. 24 (2000) 67–88.

[21] H. Wan, J.C. Wootton, Graph-theoretic approaches to biological sequences, manuscript, 1997.

[22] H. Wan, J.C. Wootton, The points of contact between globular and non-globular domains in protein sequences, in preparation.

[23] H. Wan, H. Liu, J.C. Wootton, Compositional complexity functions of biological sequences in integral partition lattices, SIAM J. Appl. Math., submitted for publication.

[24] J.C. Wootton, Non-globular domains in protein sequences: automated segmentation using complexity measures, Comput. Chem. 18 (1994) 269–285.

[25] J.C. Wootton, Sequences with ‘unusual’ amino acid compositions, Curr. Opin. Struct. Biol. 4 (1994) 413–421.

[26] J.C. Wootton, Simple sequences of protein and DNA, in: M.J. Bishop, C.J. Rawlings (Eds.), DNA and Protein Sequence Analysis, Oxford University Press, Oxford, 1996, pp. 169–183.

[27] J.C. Wootton, S. Federhen, Statistics of local complexity in amino acid sequences and sequence databases, Comput. Chem. 17 (1993) 149–163.

[28] J.C. Wootton, S. Federhen, Taxonomy of simple amino acid sequences, in: H.A. Lim, C.R. Cantor (Eds.), Bioinformatics and Genome Research, World Scientific Publishing, Singapore, 1995, pp. 159–172.

[29] J.C. Wootton, S. Federhen, Analysis of compositionally biased regions in sequence databases, Methods Enzymol. 266 (1996) 554–571.