Encoding biological sequences in the complex space

J. Avila Cartes

March 20, 2024

Abstract

In this work, we present a reversible encoding of DNA based on complex numbers inspired by the Chaos Game Representation of DNA, named *ComplexCGR*, which was designed using the Fundamental Theorem of Algebra. This representation allows us to propose a new $k-$mer-based visualization, and to encode any sequence using two integers. In addition, this encoding can be generalized to any alphabet. We show how complementary DNA sequences are related through the conjugate of their encoding in the complex space and how to skip the recursive procedure of the encoding by a vectorial representation.

Keywords— Encoding, Chaos Game Representation, Complex Numbers, k-mers

1 Introduction

The Chaos Game Representation of DNA (CGR) was proposed by Jeffrey in 1990 [3]. And was inspired by Chaos Game, an iterative process that creates points inside a polygon embedded in Euclidean space, these points are created by choosing randomly one of the vertices of the polygon and moving in the direction of the vertex a portion of the distance between the current position and the chosen vertex. The result of applying the Chaos Game usually leads to fractal images, like the Sierpinski triangle (see Figure 2).

Jeffrey noticed that he could use this game to encode DNA sequences in a square, by labeling each vertex by a nucleotide base in the alphabet $\{A, C, G, T\}$ and choosing points based on how nucleotides appear in a genome sequence. His experiments showed that genome sequences show fractal patterns under this representation.

Later on, there was shown that the same visual representation can be obtained by using the k-mer distribution of a sequence, instead of plotting each point at the time [2], known as the Frequency Matrix of CGR (FCGR). This can be explained by analyzing how k-mers are encoded in the square, and the conclusion is quite simple: in each sub-quadrant of the square, the four k-mers encoded there share the same $(k-1)$-long prefix, and sequences longer than k sharing a specific k-mer as suffixes are also inside that subquadrant.

CGR has led to many applications and studies, like new encoding aiming to move from floating point representations to fully integer ones [7], the connection with data

structures [6], or extensions for proteins to work with deep learning [4], and classification of SARS-CoV-2 sequences [1]. We refer the reader to [5] for a more extensive review of applications of CGR in bioinformatics.

Inspired by the CGR encoding, in this work we propose an encoding for a general alphabet in the complex space, named ComplexCGR, which relies on the Fundamental Theorem of Algebra, proved by Carl Friedrich Gauss in 1799. In particular, we show how complex numbers and their properties, in particular the conjugate of the complex encoding, is able to capture the complementary behavior of the double strand of DNA.

We also show how to skip the recursive encoding by defining the *reverse encoding space*, where the encoding of sequences in the original space, and *reverse sequences* in the *reverse encoding space* share the same numeric representation. This helps us to understand the connection between DNA sequences and the encoding of their complementary sequences in the complex space.

Finally, we show how this complex representation leads to a very intuitive visual representation of k-mer distribution embedded in a circle, which can complement the visual analysis made with the square representation obtained from the FCGR, like the identification of homopolymer regions, or GC content.

The source code to use the encoding and generate visual representations of k-mer distributions with both, CGR and ComplexCGR, can be found in our repository, and installed via pip `https://github.com/AlgoLab/complexCGR`.

2 Preliminaries

Let s be a string (or sequence) of N letters over the alphabet Σ of constant size. The length of s is given by N, the i-th element of s is denoted by $s_i, i \in \{1, \ldots, N\}$.

Given the DNA alphabet $\{A, C, G, T\}$, and a sequence s over this alphabet, the reverse sequence of s is denoted by s^r, and satisfies $s_i^r = s_{N-i+1}, \forall i$. The complementary sequence of s is denoted by s^c and satisfies $s_i^c = A$ iff $s_i = T$, $s_i^c = C$ iff $s_i = G$, $s_i^c = G$ iff $s_i = C$, and $s_i^c = T$ iff $s_i = A$. The reverse and complement $\hat{s}$ of a sequence s is obtained by reversing s and then computing the complementary sequence of s^r, or equivalently, by first computing s^c and then reversing it, see Figure (1).

Example 1. *Given the sequence $s = ACCGTTT$, its reverse sequence $s^r = TTTGCCA$, the complementary sequence $s^c = TGGCAAA$, and the reverse and complement $AAACGGT$, notice that the reverse and complement can be obtained by first computing the complement, and then reversing it, as illustrated in Figure 1*

Given a complex number z, its conjugate is denoted by $\bar{z}$, if $z = x + iy$, then $\bar{z} = x - iy$. The polar form of the complex number z is $|z|cis(\theta)$, where $cis(\theta) = \cos(\theta) + i\sin(\theta)$, , θ is called the argument of z, and corresponds to the angle from the real positive axis $z = 1 \in \mathbb{C}$ to the point z, and $|z| = \sqrt{x^2 + y^2}$ is the module of z.

3 Encoding biological sequences

3.1 Chaos Game Representation of DNA (CGR)

Let us consider the alphabet $\Sigma = \{A, C, G, T\}$. And the function $g : \Sigma \mapsto \mathbb{Z}^2$, which maps each nucleotide b to a vertex of the square $[-1, 1]^2$ embedded in the Euclidean plane.

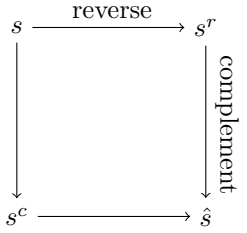

Figure 1: Illustration on how to obtain the reverse and complement of a sequence. First computing the reverse and then its complementary sequence, or the other way around.

$$g(b) = \begin{cases} (1,1) & ,b = A \\ (-1,1) & ,b = C \\ (-1,-1) & ,b = G \\ (1,-1) & ,b = T \end{cases} \tag{1}$$

In what follows, we will define $g(b) = (g_x(b), g_y(b))$, to refer to each component independently. Notice that the assignment of the corners (or quadrants) is in lexicographic order, this can be modified arbitrarily, but we will stick to this order, which leads to interesting results in both the CGR encoding and in the complex space.

Definition 1 (CGR Encoding)**.** *Given a sequence $s \in \Sigma^N$, its CGR encoding is given by the two dimensional vector (x_N, y_N), which is obtained iteratively by the following recursion,*

$$\begin{cases} (x_0, y_0) & = (0,0) \\ (x_i, y_i) & = \left(\frac{x_{i-1}+g_x(s_i)}{2}, \frac{y_{i-1}+g_y(s_i)}{2}\right), i = 1, \ldots, N \end{cases} \tag{2}$$

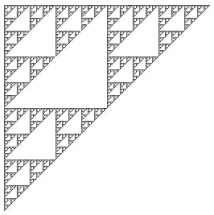

Figure 2: FCGR of a random sequence in the absence of nucleotide T.

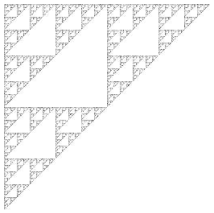

Figure 3: FCGR of a random sequence in the absence of nucleotide T and lots of unknown ones.

3.2 Encoding biological sequences in the complex space $\mathbb{C}$

Our representation is more general than the CGR encoding since it applies to an arbitrary alphabet, and differs conceptually w.r.t other extensions of CGR that are attempts to directly use the square where the CGR encoding was defined.

The ComplexCGR is an encoding that allows representing a string, of any alphabet, as a complex number z, which we will see can be represented by two integers.

From the Fundamental Theorem of Algebra, we know that the equation $z^n = 1$ has exactly n different roots in $\mathbb{C}$, given by

$$w_k = |z|^{1/n} cis\Big(\frac{\theta + 2k\pi}{n}\Big), k \in \{0, \ldots, n-1\} \tag{3}$$

where $cis(\alpha) = \cos(\alpha) + i\sin(\alpha)$, $\theta \in [0, 2\pi)$ is the argument of z, *i.e.* the angle of the line between the origin and z w.r.t. the positive real axis, and $|z|$ its module. The ComplexCGR is built under the idea that all N-long sequences over an alphabet Σ can be uniquely assigned to one of the roots of the equation

$$z^{|\Sigma|^N} = 1$$

The bijective mapping between the N-long sequences and the roots is made by sorting the N-long sequences in colexicographic order, which keeps the property of the CGR, where sequences sharing the same suffix are close in the encoding space. See Figure 4 to see the location of 2-mers under the ComplexCGR.

Given an N-long sequence over an alphabet Σ, our goal is to find the root w_k of $z^{|\Sigma|^N} = 1$ that should be assigned to s.

$$w_k = cis\Big(\frac{2k\pi}{|\Sigma|^N}\Big) \tag{4}$$

Let us consider the mapping $g : \Sigma \mapsto \mathbb{N} \cup \{0\}$, which maps each character $b \in \Sigma$ to an integer, in lexicographic order, starting from 0 until $|\Sigma| - 1$. For the particular case of the DNA alphabet, we have,

$$g(b) = \begin{cases} 0 & , b = A \\ 1 & , b = C \\ 2 & , b = G \\ 3 & , b = T \end{cases} \tag{5}$$

To encode a sequence following the approach of CGR, we need to define an iterative procedure to encode a sequence character by character. Since we started by defining the order we want to encode each N-long sequence w.r.t. the complex roots, we need to define how we should jump from a l-esim root to the corresponding $(l+1)$-esim root, for all $1 \leq l \leq N-1$, when encoding the next character s_{l+1} in the sequence. The intuition behind the ComplexCGR encoding is that every time a new character is encoded, the region between two consecutive roots is subdivided in 4 equidistant points in the circumference, one for each character in the alphabet $\{A, C, G, T\}$, and this easily generalizes to other Σ alphabets.

The iterative procedure of the ComplexCGR encoding for an arbitrary alphabet Σ is defined as follows:

Definition 2 (ComplexCGR)**.** *Given a sequence $s \in \Sigma^N$, its ComplexCGR encoding is given by the complex number $w_k = cis(2k\pi/|\Sigma|^N)$, or equivalently, by the pair (k, N), where $k = k_N$ is obtained iteratively by the following recursion*

$$\begin{cases} k_0 & = 0 \\ k_i & = g(s_i)|\Sigma|^{i-1} + k_{i-1}, i = 1, \ldots, N \end{cases} \tag{6}$$

In what follows we will focus on the DNA alphabet, and describe some properties that link complex numbers and DNA sequences.

Example 2. *Given $s = ACGT$, we can compute the ComplexCGR following 6: $k_0 = 0$, then $k_1 = g(A) + k_0 = 0$, $k_2 = g(C) \cdot 4 + k_1 = 4$, $k_3 = g(G) \cdot 4^2 + k_2 = 36$, and finally $k_4 = g(T) \cdot 4^3 + k_3 = 228$.*

Then, the ComplexCGR encoding of s is given by the pair $(228, 4)$, or equivalently the complex number $z = cis(456\pi/256)$

3.2.1 Encoding the Complementary Sequence in $\mathbb{C}$

Lemma 1. *Given $s \in \{A, C, G, T\}^N$ and, s^c its complementary sequence. If (k, N) is the ComplexCGR encoding for s, then $(4^N - k - 1, N)$ is the ComplexCGR encoding for s^c*

Proof. (By induction) For $N = 1$, we have $w_0 = (1, 0) \to A$,$w_3 = (0, -1) \to T$ and $w_1 = (0, 1) \to C$, $w_2 = (-1, 0) \to G$, then the result is true for this case.

Now, given $s \in \{A, C, G, T\}^N$ with encoding (k_N, N), and $\bar{s}$ with encoding $(\bar{k}_N, N)$, we need to prove that $\bar{k}_N = 4^N - k_N - 1$.

$$
\begin{aligned}
\bar{k}_N &= g(\bar{s}_N)4^{N-1} + \bar{k}_{N-1}, \text{ by Definition 6} \\
&= g(\bar{s}_N)4^{N-1} + 4^{N-1} - k_{N-1} - 1, \text{ by induction hypothesis} \\
&= g(\bar{s}_N)4^{N-1} + 4^{N-1} - k_N + g(s_N)4^{N-1} - 1, \text{ by Definition 6} \\
&= 4^{N-1}\Big(g(s_N) + g(\bar{s}_N) + 1\Big) - k_N - 1 \\
&= 4^N - k_N - 1, \text{by Definition of } g \text{ (5) complementary nucleotides sums 3}
\end{aligned}
$$

□

Example 3. *Given $s = TGCA$, we can compute its ComplexCGR encoding using Lemma 1, since $s^c = ACGT$, and from Example 2 we have $ComplexCGR(s^c) = (228, 4)$, then $ComplexCGR(s) = (4^4 - 228 - 1, 4) = (27, 4)$*

A consequence of Lemma 1 is that under a rotation of $\Delta/2$ of the roots, where Δ is the angle between two consecutive roots, the complex number resulting from the ComplexCGR encoding of the complementary sequence of s can be found by computing the conjugate of the complex number of s, as stated by the following Collorary.

Corollary 1. *Given a sequence $s \in \{A, C, G, T\}^N$ and its complementary sequence s^c. If $z \in \mathbb{C}$ is the root for s, and $w \in \mathbb{C}$ is the root for s^c, then $w \cdot cis(\Delta/2) = \overline{z \cdot cis(\Delta/2)}$, where $\Delta = 2\pi/4^N$ is the angle between two consecutive 4^N-esim roots of $z^{4^N} = 1$.*

Proof. Given $s \in \{A, C, G, T\}^N$, with $z \in \mathbb{C}$ the root associated to s under the ComplexCGR encoding, then there exists $k \in \{0, ..., 4^{N-1}\}$ such that

$$z = cis\Big(2k\pi/4^N\Big)$$

By Lemma 1 the complementary sequence s^c can be computed by

$$w = cis\Big(2(4^N - k - 1)\pi/4^N\Big)$$

Then,

$$
\begin{aligned}
w \cdot cis(\Delta/2) &= cis\Big(\frac{2(4^N - k - 1)\pi}{4^N}\Big) \cdot cis\Big(\frac{\Delta}{2}\Big) \\
&= cis\Big(2\pi - \frac{2k\pi}{4^N} - \frac{2\pi}{4^N}\Big) \cdot cis\Big(\frac{\pi}{4^N}\Big) \\
&= cis\Big(2\pi - \frac{2k\pi}{4^N} - \frac{2\pi}{4^N} + \frac{\pi}{4^N}\Big) \\
&= cis\Big(-\frac{2k\pi}{4^N} - \frac{\pi}{4^N}\Big) \\
&= \overline{cis\Big(\frac{2k\pi}{4^N} + \frac{\pi}{4^N}\Big)} \\
&= \overline{cis\Big(\frac{2k\pi}{4^N}\Big) \cdot cis\Big(\frac{\pi}{4^N}\Big)} \\
&= \overline{z \cdot cis(\Delta/2)}
\end{aligned}
\tag{7}
$$

□

Notice that Corollary 1 is very intuitive once you notice that $4^N - k - 1$ is the k-esim root clockwise direction, then the rotation $\Delta/2$ helps to align those roots with some z and $\overline{z}$.

Corollary 1 can be helpful to rotate the kmer positions when visualizing the distribution the circumference, as we will see in Section 3.2.5

3.2.2 Finding the encoding of the reverse and complement in $\mathbb{C}$

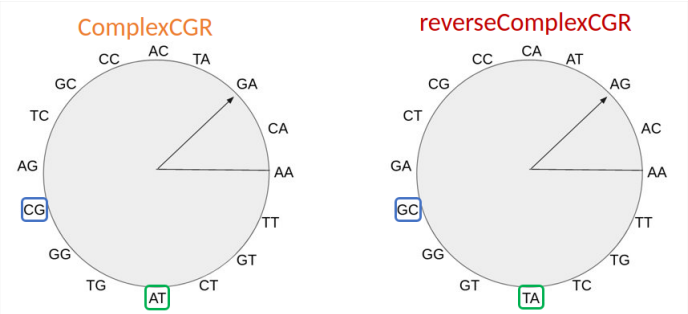

Figure 4: (left) ComplexCGR and (right) reverseComplexCGR spaces for 2-mers in the DNA alphabet. Under the ComplexCGR encoding sequences are ordered in colexicographic order, while under the reverse ComplexCGR encoding, sequence are ordered in reverse order w.r.t. the ComplexCGR space, *i.e.* in lexicographic order.

Given a sequence $s \in \Sigma^N$ and its reverse sequence s^r, *i.e.* $s_i = s^r_{N-i+1}, \forall i$, we define the reverseComplexCGR as the encoding that satisfies the following Indentity

Identity 1 (Fundamental Identity of ComplexCGR)**.**

$$
ComplexCGR(s) = reverseComplexCGR(s^r) \tag{8}
$$

The $reverseComplexCGR$ encoding is obtained similarly to the $ComplexCGR$, where each sequence of a fixed length is assigned to one of the roots, but this time assigning each sequence in Σ^N in lexicographic order to the roots of $z^{4^N} = 1$. The $reverseComplexCGR$ encodes close in the circumference sequences sharing the same prefix. Let us recall that under the $ComplexCGR$ sequences sharing the same suffix are closer in the complex space.

Definition 3 (reverseComplexCGR)**.** *Given a sequence $s \in \Sigma^N$, its $reverseComplexCGR$ encoding is given by the pair (k, N), where k is obtained as follows*

$$k = \sum_{i=1}^{N} 4^{N-i} g(s_i) \tag{9}$$

In what follows, given the sequence s, we will use $ComplexCGR(s)$ to denote the value k of the encoding (k, N), same for $reverseComplexCGR(s)$, and the same applies to s^r, s^c and $\hat{s}$.

$$\begin{aligned} reverseComplexCGR(s^c) =& \sum_{i=1}^{N} 4^{N-i} g(s_i^c) \\ =& \sum_{i=1}^{N} 4^{N-i} (3 - g(s_i)) \\ =& 3\sum_{i=1}^{N} 4^{N-i} - \sum_{i=1}^{N} 4^{N-i} g(s_i) \\ =& 3\sum_{i=1}^{N} 4^{N-i} - reverseComplexCGR(s) \end{aligned} \tag{10}$$

It follows that

$$reverseComplexCGR(s) + reverseComplexCGR(s^c) = 3\sum_{i=1}^{N} 4^{N-i} \tag{11}$$

now in terms of $ComplexCGR$ by using property 8, we obtain the following Identity.

Identity 2 (Fundamental Identity of ComplexCGR)**.**

$$ComplexCGR(s^r) + ComplexCGR(\hat{s}) = 3\sum_{i=1}^{N} 4^{N-i} \tag{12}$$

or equivalently,

$$reverseComplexCGR(s) + ComplexCGR(\hat{s}) = 3\sum_{i=1}^{N} 4^{N-i} \tag{13}$$

The following Identity is a consequence of Identity 1 and Lemma 1,

Identity 3.

$$4^N - 1 = 3\sum_{i=1}^{N} 4^{N-i}$$

Proof. By using Identity 2 with the reverse sequence of each argument, we have (s^c is the reverse of $\hat{s}$)

$$ComplexCGR(s) + ComplexCGR(s^c) = 3\sum_{i=1}^{N} 4^{N-i}$$

On the other hand, if $ComplexCGR(s) = k$, by Lemma 1, $ComplexCGR(s^c) = 4^N - k - 1$. Then, it follows that

$$4^N - 1 = 3\sum_{i=1}^{N} 4^{N-i}$$

□

Now it is time to write an alternative definition for $ComplexCGR$.

Proposition 1.

$$ComplexCGR(s) = \sum_{i=1}^{N} 4^{i-1} g(s_i)$$

Proof. Given an N-long sequence s and s^r its reverse sequence, $s^r_i = s_{N-i+1}$ holds for all positions i in s. By Identity 1, it follows

$$\begin{aligned} ComplexCGR(s) =& reverseComplexCGR(s^r) \\ =& \sum_{i=1}^{N} 4^{N-i} g(s^r_i) \\ =& \sum_{i=1}^{N} 4^{N-i} g(s_{N-i+1}) \\ =& \sum_{i=1}^{N} 4^{i-1} g(s_i) \end{aligned} \tag{14}$$

□

Notice that Proposition 1 can be also obtained directly from Definition 2. Using the recursion 6 and induction over N,

$$\begin{aligned} ComplexCGR(s) =& k_N \\ =& 4^{N-1} g(s_N) + k_{N-1} \text{ , by Definition 2} \\ =& 4^{N-1} g(s_N) + \sum_{i=1}^{N-1} 4^{i-1} g(s_i), \text{ , by induction hypothesis} \\ =& \sum_{i=1}^{N} 4^{i-1} g(s_i) \end{aligned} \tag{15}$$

With Proposition 1 we have skipped the iterative procedure to encode the sequence with $ComplexCGR$. A consequence of this is that now the $ComplexCGR$ encoding has a vectorial representation:

$$ComplexCGR(s) = \begin{pmatrix} 4^0 & 4^1 & \dots & 4^{N-1} \end{pmatrix} \cdot \begin{pmatrix} g(s_1) \\ g(s_2) \\ \dots \\ g(s_N) \end{pmatrix} \tag{16}$$

3.2.3 A more general ComplexCGR

Definition 4 ($(\mathcal{P}, \mathcal{F}) - ComplexCGR$: A general encoding)**.** *Given a sequence $s \in \{A, C, G, T\}$, $\mathcal{P}$ a permutation matrix of size $N \times N$, and $\mathcal{F}$ a bijective mapping between the alphabet and $\mathbb{Z}$, that is applied to each character of the sequence s independently. Let us define the $(\mathcal{P}, \mathcal{F}) - ComplexCGR$ as follows,*

$$(\mathcal{P}, \mathcal{F}) - ComplexCGR(s) = \vec{v} \cdot \mathcal{P} \cdot \mathcal{F}(s) \tag{17}$$

where $\vec{v} = [1, 4, 4^2, \dots, 4^{N-1}]^t$, $\mathcal{P}$ is a permutation matrix of size (N, N), and $\mathcal{F} : \{A, C, G, T\} \mapsto [0, 3]$ is a bijective mapping between the alphabet and integers from 0 to 3.

Let us define $P = (p_{ij})$ the permutation matrix s.t. $p_{ij} = 1 \Leftrightarrow i + j = N + 1$, I the identity matrix. g is the function defined in 5, and $f = 3 - g$ which maps each nucleotide its complementary nucleotide value trough g.

1. $(I, g) - ComplexCGR(s) = ComplexCGR(s)$
2. $(P, g) - ComplexCGR(s) = ComplexCGR(s^r)$
3. $(I, f) - ComplexCGR(s) = ComplexCGR(s^c)$
4. $(P, f) - ComplexCGR(s) = ComplexCGR(\hat{s})$

Alternatively, to obtain the encoding of the reverse and complement sequence of s, $ComplexCGR(\hat{s})$, we can compute the encoding of the reverse sequence $ComplexCGR(s^r)$ and then use Lemma 1 to compute the encoding of the complement.

In summary, given s, s^r, s^c, and $\hat{s}$, the sequence, reverse, complementary, and reverse and complement sequences, respectively. With $ComplexCGR$ encodings (k, N),(k^r, N), (k^c, N), $(\hat{k}, N)$, respectively. Using Lemma 1 we can obtain k^c from k, and $\hat{k}$ from k^r. Using property 8 we can obtain k^r from k, and $\hat{k}$ from k^c. And finally, we can go directly from k to $\hat{k}$ using Identity 13.

draw a graph with the above relationships. Also the relationships in the sequence space between s, s^r, s^c, and $\hat{s}$

3.2.4 Decoding the ComplexCGR

The ComplexCGR encoding, as well as the CGR encoding, is reversible, *i.e.* , given a valid encoding (k, N), we can reconstruct the sequence s it is encoded there. To do this, the procedure is very simple, we start by decoding the last character in the sequence by checking in which quadrant is encoded the current number, by using the argument $\theta = 2k\pi/4^N$.

Given a sequence $s \in \{A, C, G, T\}^N$ with ComplexCGR encoding (k_N, N), and argument $\theta = 2k\pi/4^N$, we can identify which is the last character in s by simply

checking the argument θ,

$$\begin{cases} \theta \in [0, \pi/2) & \Rightarrow s_n = A \\ \theta \in [\pi/2, \pi) & \Rightarrow s_n = C \\ \theta \in [\pi, 3\pi/4) & \Rightarrow s_n = G \\ \theta \in [3\pi/4, 2\pi) & \Rightarrow s_n = T \end{cases}$$

once the last character is identified, we can obtain the encoding of the prefix of s by using 6, as follows $k_{i-1} = k_i - g(s_i)4^{i-1}$, and repeat the process until the N characters has been identified.

Include visualization of 2−mers and 3−mers before and after rotation

3.2.5 Visualization of k-mer distributions with ComplexFCGR

- Plot a species with homopolymer regions, and show what is the figure and how to identify this region
- Plot a species with high GC content, and show how to identify this. Also give the numbers of kmers with GC
- show how to aggregate the results of the distribution. Eg: count all A's from the plot.
- For each species, plot k=6,7,8,9 for both CGR and ComplexCGR
- consider to plot samples (reads)

Given a sequence, we can count how many times each k-mer is present in it. Then, we obtained the distribution of k-mers by dividing each frequency by the total number of k-mers. Finally, a circular plot is created by creating a bar from the center of the unit circumference in the direction of the encoding of each k-mer, with a length equal to its representativity which is a value in $[0, 1]$. We named this visual representation $ComplexFCGR$, following the naming of the $FCGR$ obtained from the CGR encoding.

See Figure 5 for a comparison between $ComplexFCGR$ and $FCGR$ for different species and values of k. GC content can be identified as a diagonal in the FCGR, and by looking at the middle of the second quadrant in the ComplexFCGR, *i.e.* $3\pi/4$ w.r.t. positive real axis (see Figure 4 for reference). Homopolymer regions like AAAAAA can be seen as a black dot in the right upper corner in the FCGR, and as a long bar in the positive real axis in the plot, or a 0 degree angle.

4 The meaning of...

Notice that for any $s_1, s_2 \in \mathcal{S}_N = \{A, C, G, T\}^N$, and $ComplexCGR$ encodings (k_1, N) and (k_2, N), with $z_1 = cis(\theta_1)$, $z_2 = cis(\theta_2)$, their complex representations under the $ComplexCGR$ encoding, *i.e.* $\theta_1 = 2k_1\pi/4^N$, $\theta_2 = 2k_2\pi/4^N$ the output of the following operation is a valid $ComplexCGR$ encoding.

4.1 Inverse multiplicative of a root

Given s with encoding (k, N), and complex number z. Let z^{-1} the inverse multiplicative of z, then $(4^N - 1, N)$ is the encoding associated with z^{-1}

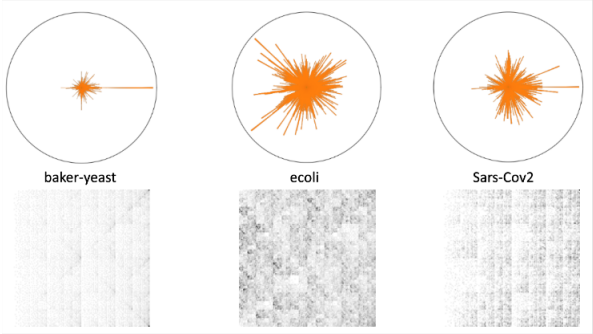

Figure 5: ComplexFCGR vs FCGR for different species.

4.2 Multiplication between two roots

$z_1 \cdot z_2 = cis(\theta_1 + \theta_2) = z$, and s_z the sequence with complex number z then

$$ComplexCGR(s_z) = (k_1 + k_2) mod 4^N$$

4.3 Division between two roots

$z_1/z_2 = cis(\theta_1 - \theta_2) = z$, and s_z the sequence with complex number z then

$$ComplexCGR(s_z) = (k_1 - k_2) mod 4^N$$

5 Conclusions

In this work, we presented a new reversible encoding for biological sequences over an arbitrary alphabet. This led to a visualization of k-mer distributions that could help to identify patterns in the sequences. We showed how DNA and complementary sequences can be linked directly in the complex space. By using the $reverseComplexCGR$, we were able to skip the recursive encoding needed to find the complex representation of a sequence, this resulted in an equation involving $ComplexCGR$ and $reverseComplexCGR$.

We expect this work can serve as the basis for more applications and explorations of sequences in the complex space, which also opens the possibility to explore polynomial representations of a set of $k-$mers, or sequences.

The resulting visual representation could be used as input for Deep Learning models that work with images.

5.1 Funding

This project has received funding from the European Union's Horizon 2020 Innovative Training Networks program under the Marie Skłodowska-Curie grant agreement No. 956229.

References

[1] Jorge Avila Cartes, Santosh Anand, Simone Ciccolella, Paola Bonizzoni, and Gianluca Della Vedova. Accurate and fast clade assignment via deep learning and frequency chaos game representation. *GigaScience*, 12, 2022.

[2] P J Deschavanne, A Giron, J Vilain, G Fagot, and B Fertil. Genomic signature: characterization and classification of species assessed by chaos game representation of sequences. *Molecular Biology and Evolution*, 16(10):1391–1399, 10 1999.

[3] H.Joel Jeffrey. Chaos game representation of gene structure. *Nucleic Acids Research*, 18(8):2163–2170, 04 1990.

[4] Hannah F Löchel, Dominic Eger, Theodor Sperlea, and Dominik Heider. Deep learning on chaos game representation for proteins. *Bioinformatics*, 36(1):272–279, 2020.

[5] Hannah Franziska Löchel and Dominik Heider. Chaos game representation and its applications in bioinformatics. *Computational and Structural Biotechnology Journal*, 19:6263–6271, 2021.

[6] Susana Vinga, Alexandra M Carvalho, Alexandre P Francisco, Luís MS Russo, and Jonas S Almeida. Pattern matching through chaos game representation: bridging numerical and discrete data structures for biological sequence analysis. *Algorithms for Molecular Biology*, 7:1–12, 2012.

[7] Changchuan Yin. Encoding dna sequences by integer chaos game representation. *Journal of Computational Biology*, 26, 12 2017.

Appendix A String operations with Complex-CGR

Definition 5 (concatenation of two strings)**.** *Given two strings s and t, the concatenation of them is the string $w = \text{concat}(s,t)$ s.t. $w[:|s|] = s$ and $w[|s|+1:] = t$.*

Proposition 2 (concatenation of two strings in the complex space)**.** *Given z_s and z_t the encoding of the strings s and t, respectively. Then, the encoding of the string $w = \text{concat}(s,t)$ is the complex number $z = z_s \cdot z_t^{1/4^n}$*