Note: This Series is quite long, but you can do various bits at whatever time you like. Questions 1 and 2 involve coding. The objective is to familiarise yourself with the main outputs of the cgDNA+ Matlab package which contructs a ground state $\mu(S)$ and a symmetric stiffness matrix $K(S)$ for any input sequence S (given a parameter set $\mathcal{P}$). In turn the ground state $\mu(S)$ and the stiffness matrix $K(S)$ give the cgDNA+ model Gaussian pdf approximation to the equilibrium distribution of the dsDNA fragment of that sequence. The underlying modelling assumptions and algorithms that are implemented in the cgDNA+ package are the subject of the Week 7 lectures. Questions 3-5 are primarily more theoretical.

1 On the main outputs of the cgDNA+ package

1.1 Stiffness matrices

For any input sequence, cgDNA+ first constructs a stiffness matrix K, i.e, a symmetric positive definite matrix ($K^T = K$, $K > 0$).

1. Using the Matlab function `spy`, visualize the sparsity pattern of the predicted K for the example sequence $S = CGCGAATTCGCG$. What is the sparsity pattern of the inverse of the stiffness?

2. Perform a single point mutation on the given sequence, i.e, change a single base in the sequence. Reconstruct the stiffness matrix $\tilde{K}$ for the new sequence and use `spy` to visualize the difference $(K - \tilde{K})$. What can you say?

3. Fully comparing two symmetric positive definite matrices is not a simple mathematical problem. A first strategy is to compare their eigenvalues. Use the `eig` function to compare the eigenvalues of K and $\tilde{K}$, then plot them in the same graph. What can you say?

4. Construct randomly a 200 base-pair long sequence and compute the eigenvalues of its stiffness matrix. Compare it with the eigenvalues of K and $\tilde{K}$. What can you say?

1.2 Visualization of the ground state

There are many ways to visualize the ground state predicted by cgDNA+. The first way is to visualize the helical parameters (see Figure 1 Series 5) and rotational (i.e. Cayley vectors) and translational coordinates of phosphates (see Figure 2 Series 5), of the predicted ground state. Run the matlab script `main.m` given in cgDNA+ package to plot the helical parameters and phosphates coordinates. What can you say about the plots? Then using the function `cgDNAp3dplot`, visualize the 3D shape of groundstate of the given sequence. What can you observe?

Matlab molecular viewer

Note that EPFL provide you Bioinformatics Toolbox of Matab, and you can use the built-in molecular viewer, `molviewer` of Matlab, to visualize the ground-state predicted by cgDNA+. In fact by running the script `main.m` of the cgDNA+ package it automatically generates and saves a `.pdb` file that contains the Cartesian coordinates of all the atoms forming all the bases and phosphates of the predicted fragment. As we predict a coarse grained configuration, the atom positions in each

bases and phosphates are retrieved by embedding idealized atoms. For this reason we have a set of idealized atoms position for each base and phosphates.
Remark: You can also construct and visualize the ground state for any sequence from the web server cgDNAweb+ (`http://cgdnaweb.epfl.ch`), which also allows a download of the pdbfile for the ground state of any sequence.

1.3 Understanding the function `frames`

Begin first by reading the description of the function in `frames.m`. Then

1. Understand from the function `frames`, the reconstruction rule for the absolute position and orientation for each base and each phosphate on both strands, by writing down explicitly the main recursion rules.

2. Modify the output of the function `frames` by adding two more fields: one for the absolute position of the base pair and another for its orientation. Your modified version of `frames` must produced the following output using the default sequence:

```
basepair =

12x1 struct array with fields:

    R
    r
    Rw
    rw
    Rc
    rc
    Rpw
    rpw
    Rpc
    rpc
```

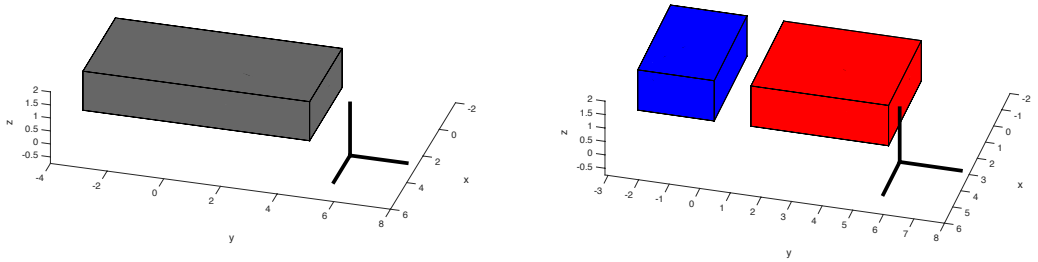
2 A MATLAB cgDNA viewer

The aim of this exercise is to write a MATLAB script to visualize the bases (in this particular exercise we ignore visualization of phosphates) as rigid bodies. Moreover your viewer should also plot, in a different figure, the rigid body associated to the base pairs. First download the MATLAB functions `getRigidBodyConfig` and `displayRigidBodies` from `https://lcvmwww.epfl.ch/teaching/modelling_dna/protected_files/codes_exercises/Suppl_MATLAB_func.zip`, that will help you plot a parallelepiped in MATLAB. Here you have an example of how the functions work (run script `visualize_run.m`):

```
% Visualize a rigid body corresponding to a base pair
RB=getRigidBodyConfig([0;0;0],eye(3)) ;
displayRigidBodies(RB)

% Visualize two rigid bodies corresponding to two bases
RB=getRigidBodyConfig([0;0.5;0],eye(3),'A') ;
RBc=getRigidBodyConfig([0;-0.5;0],eye(3),'T','Compl') ;
displayRigidBodies(RB,RBc)
```

And you have following output of the example:



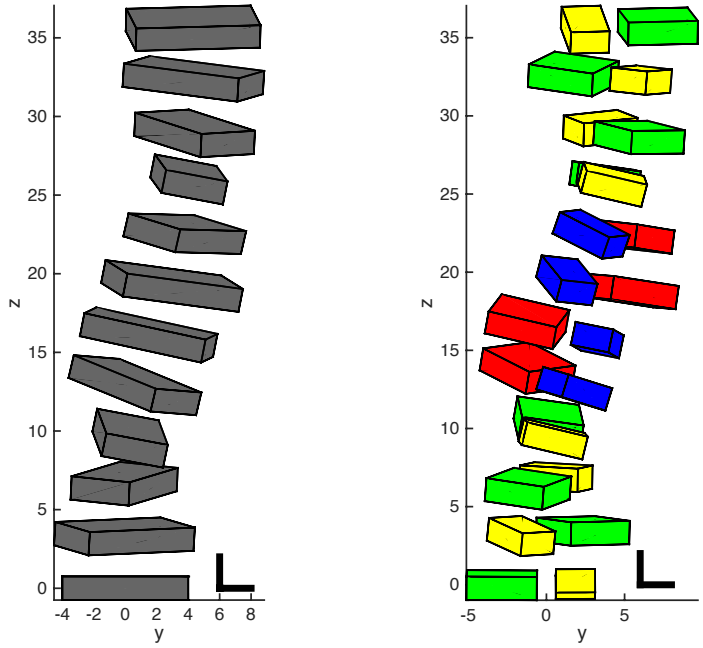
Before running the example you should complete a line in the function `getRigidBodyConfig`, then use to check your result.
Your viewer must be called `cgDNAviewer` and must take two input arguments: the `basepair` structure array and the `sequence` variable, i.e, you will add to your main function the following lines:

```
%% Plot the rigid bodies corresponding to the bases and the base pairs
cgDNAviewer( basepair, sequence );
```

Code the following steps of the `cgDNAviewer`:

1. Use the structure array `basepair` to plot the rigid bodies of all the bases for both strands.
2. Use the new added fields of the structure array `basepair` to plot the rigid bodies corresponding to the base pairs.

Once your `cgDNAviewer` works you will be able to visualize the 3D reconstruction of any sequence:

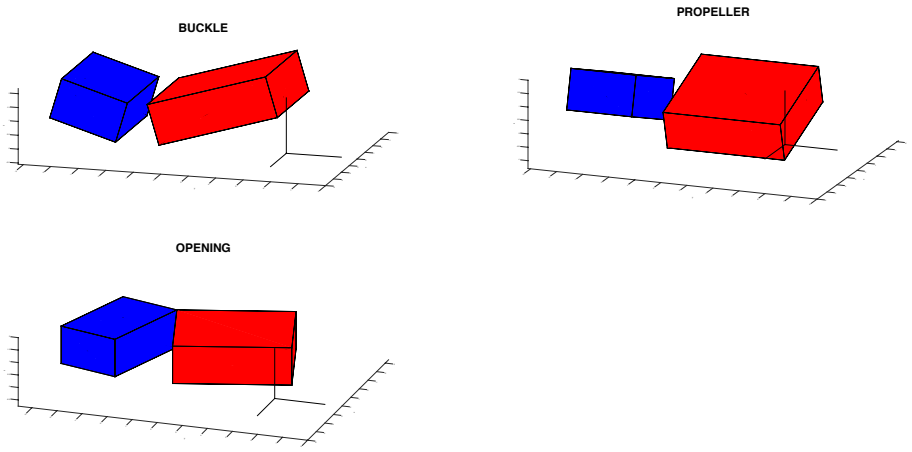


Now you can use your viewer to check that the components (intras and inters) of the cgDNA+ coordinates are the standard helical parameters (see figure 1 Serie 5). From solution of Qu 1.3 of this serie,

recall that the cgDNA+ groundstate $z = (y_1, y_1^{pC}, x_1, y_2^{pW}, y_2, y_2^{pC}, x_2, \ldots, x_{n-1}, y_n^{pW}, y_n) \in \mathbb{R}^{24n-18}$ is divided into *intra variables* $y_i := (\eta_i, w_i) \in \mathbb{R}^6$, *inter variables* $x_i = (u_i, v_i) \in \mathbb{R}^6$ and phosphates coordinates ($y_i^{pC} \in \mathbb{R}^6, y_i^{pW} \in \mathbb{R}^6$), where η_i, u_j are Cayley vectors encoding the rotational part and w_i, v_j are the translational part for intras and inters. Each component of the intras and inters variables has a name respecting the standard nomenclature for the DNA helical parameters (see Figure 1 in Serie 5). For example the components for each η_i are respectively Buckle-Propeller-Opening.

You should write a function called `cgDNAviewer_test ( nondimshape, sequence )` that can reproduce the following example:

```
cgDNAviewer_test( [3 0 0 0 0 0], 'A' ) ; % BUCKLE
cgDNAviewer_test( [0 3 0 0 0 0], 'A' ) ; % PROPELLER
cgDNAviewer_test( [0 0 3 0 0 0], 'A' ) ; % OPENING
```



Check also the translational part of the intra variable. Perform then all the tests to check the base pair helical parameters. Of course all six arguments can be non zero.

3 Scaled Cayley transform

In exercise 4 of serie 5 the function $CayTra$ is defined for general matrices and also $CayTra : \mathbb{R}^{3\times3} \to \mathbb{R}^{3\times3}$ restricts to $Sk \to SO(3)$, $Q = CayTra([\mathbf{u}\times])$ where Sk is all 3×3 skew matrices $[\mathbf{u}\times]$ and Q is a proper rotation matrix. Moreover $Q \in SO(3)$ is a Cayley transform except for rotations through π.

1. Compute the explicit form of $CayTra(N)$ for $N = [\frac{1}{\alpha}\mathbf{u}\times]$.
2. For $N = [\frac{1}{\alpha}\mathbf{u}\times]$ compute the explicit version of equations 6 and 7 of Serie5, that give $\mathbf{u}$ in term of Q and α.
3. In the function `frames.m` of the cgDNA+ Matlab package, implement the Cayley transformation with the α scaling founded in part 1).

Exercise 4 of serie 5 shows that $\alpha = 2$ is one sensible choice. In the cgDNA+ model we use the scaling $\alpha = 10$ for reasons that will be explained later.

4 Proof of the change of reading strand transformation Part–2

Consider a fragment of DNA of length of N base pairs. Using the convention of the change of reading strand transformation and part I of this exercise (Ex. 2.1, Serie 4), write explicitly the change of variable for the internal coordinates (z) by considering the transformation between

$$\left\{(R_i, r_i)^C, (R_i, r_i)^W\right\}_{i=1}^N \mapsto \left\{(\overline{R}_k, \overline{r}_k)^C, (\overline{R}_k, \overline{r}_k)^W\right\}_{k=1}^N .$$

Note: $\overline{R}^C$, and $\overline{R}^W$ are defined in the same way as are defined in part I (Ex. 2.1, Serie 4). Here we choose W = "plus" and C = "minus".

5 On the symmetry of the coordinate system

For a base $X \in \{A, T, G, C\}$ we note its complementary base by $\overline{X}$. Moreover if $S = X_1 X_2 \dots X_{N-1} X_N$ is a sequence, we note by $\overline{S} = \overline{X}_N \overline{X}_{N-1} \dots \overline{X}_2 \overline{X}_1$ its complementary sequence. We define the $(24N - 18) \times (24N - 18)$ matrix

$$E_N = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \cdots & E \\ 0 & 0 & 0 & 0 & \cdots & I & \cdots \\ 0 & 0 & 0 & \cdots & E & \cdots & 0 \\ 0 & 0 & \cdots & \ddots & \cdots & 0 & 0 \\ 0 & \cdots & E & \cdots & 0 & 0 & 0 \\ \cdots & I & \cdots & 0 & 0 & 0 & 0 \\ E & \cdots & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where $E = diag(-1, 1, 1, -1, 1, 1), I = diag(1, 1, 1, 1, 1, 1)$. In fact, the matrix E_N is a block trailing–diagonal matrix with $2N - 1$ copies of E interspersed with $2(N - 1)$ copies of I.

1) Prepare Matlab script to compute E_N for given N. Consider $S_1 = GTGAAAAAAAAAGC$ and let $S = S_1$. Write down the sequence $\overline{S}$. Using the cgDNA+ package construct the shapes ($\mathbf{z}(S)$ and $\mathbf{z}(\overline{S})$) and the stiffness matrices ($K(S)$ and $K(\overline{S})$) and check that:

$$\begin{aligned} \mathbf{z}(S) &= E_N \mathbf{z}(\overline{S}), \\ K(S) &= E_N K(\overline{S}) E_N . \end{aligned}$$

2) Take a sequence S = CGCGAATTCGCG (Drew-Dickerson dodecamer) and construct the shape vector $\mathbf{z}(S)$ and the stiffness matrix $K(S)$, then compute $E_N \mathbf{z}(S)$ and $E_N K(S) E_N$. What is the number of entries in K, where the change between K and $E_N K(S) E_N$ is bigger than 1%? What is the value of Shift in junction 6 of $\mathbf{z}$(S) and why?