**Note:** Following exercises will use notation poly$(\alpha\beta)_N$ which was introduced in Qu4 Serie 7. Download here, `https://lcvmwww.epfl.ch/teaching/modelling_dna/protected_files/codes_exercises/cgDNA+ps1`, the parameter set to be used with cgDNApmc code.

# 1 Monte Carlo simulation with the cgDNA+ model part-1

## 1.1 Monte Carlo simulation of the cgDNA+ model with matlab

Monte Carlo simulation for a multivariate normal distribution can be easily done in matlab when the mean and the covariance are given. The only command needed is `mvnrnd` (use the help command in matlab to find out how to use it).

1. Sample 250 configurations from the cgDNA+ distribution for each of the six poly$(\alpha\beta)_{50}$, and plot the xyz position of all the base–pairs of each sampled configuration plus the one for the groundstate on the same figure.

2. For each poly-dinucleotide, plot the position of the last base–pair of each sampled configuration. What you can say about the shape of the point clouds?

**Remark**: We want now to sample a larger number ($\sim 10^6$) of configurations for each poly$(\alpha\beta)_{150}$ and we will use a very efficient Monte Carlo code written in C++.

## 1.2 The cgDNApmc code

The cgDNApmc code (cgDNAmc code is written by J. Glowacki (sequence-dependent persistence lengths of DNA , J. S. Mitchell, J. Glowacki, A. E. Grandchamp, R. S. Manning and J. H. Maddocks, Journal of Chemical Theory and Computation, 2017, 1539-1555) adopted to cgDNA+ by A. Patelli (A sequence-dependent coarse-grain model of B-DNA with explicit description of bases and phosphate groups parametrised from large scale Molecular Dynamics simulations, PhD dissertation # 9552, EPFL)) will allow you to sample $10^6$ configurations of a poly$(\alpha\beta)_{150}$ in a few minutes on a regular laptop. This includes reconstructions of the $(R_i, r_i) \in SE(3)$ base–pair configurations from the sampled cgDNA+ internal coordinates.

**How to run "run_cgDNAmc_from_seq":**

To run a simulation for a given sequence using cgDNApmc one needs to give following inputs:

Input 1) : -e "expectation you want to compute" (either t0 or r, will be discussed in detail in Serie 9)

Input 2) : -l "name you want to use for the generated files"

Input 3) : -i "path to a text file with a DNA sequence"

Input 4) : -p "path to a cgDNA+paramset.txt"

Input 5) : -g "requested number of generated configuration"

Input 6) : -d "number of base–pairs to drop from both ends" (for the exercises use always 0)

Input 7) : -j "this flag indicate whether the Jacobian should be used" (for the exercises use always **n**)

Example: ./run_cgDNAmc_from_seq -e t0 -l Result/test_seq -i sequences/test_seq.txt -p cgDNA+ps1.txt -g 10000 -d 0 -j n

Run a few tests on a poly-dinucleotide in order to assure that the code is working properly.

**Modified run_cgDNAmc_from_seq**

Modify the script run_cgDNAmc_from_seq.cpp in order to save a file with the last base-pair position for each sampled configuration. Then, load this file in matlab and visualize the obtained cloud of points. What can you say?

# 2 Monte Carlo simulation with the cgDNA+ model part-2

Download here, `https://lcvmwww.epfl.ch/teaching/modelling_dna/protected_files/codes_exercises/lambda_phage.mat`, the sequences you will work on in this question. The five sequences, called $\texttt{lambda}_{1...5}$ have 300 base-pairs and are all part of the $\lambda$ phage genome (see the details in Sanger et al. 1982).

## 2.1 cgDNA+ reconstruction of the five lambda phage segments

Using the cgDNA+ matlab package reconstruct the groundstate of the five sequences and compare the 3D reconstruction of all the groundstates.

## 2.2 Monte Carlo simulation of the cgDNA+ model with matlab

See Qu. 1.1 of this session for details about Monte Carlo simulation with matlab and sample 250 configurations for each of the five sequences, and plot each sampled configuration plus the ground state on one figure. Here you can just plot the xyz position of all the base–pairs for each configuration. Are the five plot similar? Compare them to the plot obtained with the six poly-dinucleotides.

## 2.3 Cloud of points of last base-pair positions

Use cgDNApmc to sample $10^5$ configuration for each sequence and save for each sample the last base-pair position. Plot using matlab the cloud of point, what can you say? Also compare the obtained clouds with the poly-dinucleotide ones from Qu. 1.1 of this Session.

# 3 Effect of the sparsity on the Monte Carlo simulation efficiency

Nowadays Monte Carlo simulation for univariate normal distribution can be done in an extremely efficient way. Hence the univariate case is also used to sample from a multivariate normal distribution, but for taking advantage of the univariate case one has to diagonalise the stiffness matrix. Here you have two methods to diagonalise the cgDNA+ stiffness matrix.

- Use the spectral decomposition to diagonalise the cgDNA+ stiffness matrix, i.e, write $K = M\Lambda M^T$.
- Use the Cholesky factorisation to decompose the cgDNA+ stiffness matrix, i.e, write $K = L^T L$.

1. Using the matlab function `mvnrnd` sample 500 configurations using both methods of diagonalisation on poly$(AT)_N$ with different $N$ and compare the computational efficiency of both the methods.

2. As discussed in the lecture this week and also will be discussed in detail in next week's lecture/exercise that the computation of persistence length depends only on the value of the inter variables. Thus, formally the expectation of these two quantities is with respect to the marginal distribution of the inter variables. More precisely, let $x = (y^C, y, y^W, z) \in \mathbb{R}^{24n-18}$ be a cgDNA+ configuration where $y \in \mathbb{R}^{6n}, y^C \in \mathbb{R}^{6(n-1)}, y^W \in \mathbb{R}^{6(n-1)}$ are all the intra, Crick phosphate, Watson phosphate variables respectively and $z \in \mathbb{R}^{6(n-1)}$ are all the inter variables. Denote by $\phi(z)$ a non linear function of the inters. Thus the expected value of $\phi$ with respect to the cgDNA+ distribution $\rho(y^C, y, y^W, z)$ is

$$\begin{aligned}\langle\phi(z)\rangle &= \int_{\mathbb{R}^{6(n-1)}}\int_{\mathbb{R}^{6n}}\int_{\mathbb{R}^{6(n-1)}}\int_{\mathbb{R}^{6(n-1)}} \phi(z)\rho(y^C, y, y^W, z)dzdy^W dydy^C \\ &= \int_{\mathbb{R}^{6(n-1)}} \phi(z)\rho(z)_{(y^C,y,y^W)}\, dz = \langle\phi(z)\rangle_z, \end{aligned} \tag{1}$$

where $\langle\cdot\rangle_z$ is the expectation with respect to the inter's marginal distribution $\rho(z)_{(y^C,y,y^W)}$. We want now to compare the efficiency of sampling from the whole cgDNA+ distribution against sampling from the marginal with respect to the intra and phosphates variables, using Monte Carlo simulation. For doing that consider different repetitions of poly(AT) and, as method of decomposition, use the Cholesky factorisation. For this exercise you can fix the number of samples to 500.

**Remark**: The marginal distribution, of a subset of variables, of a Gaussian distribution can be obtained by selecting, in the covariance and in the mean, the corresponding blocks. For example, in the case of a cgDNA+ distribution the marginal distribution of the inters is obtained by extracting all the inter-inter blocks of the covariance matrix, and the inter part of the mean.