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

1.1 Monte Carlo simulation of the cgDNA+ model with matlab

Let $\mathcal{P}$ be a cgDNA+ parameter set and S a n base-pair long sequence. The cgDNA+ matlab package reconstructs then the groundstate $\hat{\mathbf{x}} \equiv \hat{\mathbf{x}}(S,\mathcal{P}) \in \mathbb{R}^{24n-18}$ and the stiffness $K \equiv K(S,\mathcal{P}) \in \mathbb{R}^{(24n-18)\times(24n-18)}$, thus it predicts the probability density function

$$\rho(\mathbf{x};S,\mathcal{P}) = \frac{1}{Z}\exp\left\{\frac{1}{2}(\mathbf{x}-\hat{\mathbf{x}})\cdot K(\mathbf{x}-\hat{\mathbf{x}})\right\}.$$

Using the latter probability density function we can explore the set of possible configuration for the given sequence S and parameter set $\mathcal{P}$. The Monte Carlo method, of the cgDNA+ model, consist in evaluating a deterministic function on an ensemble of configurations $\{\mathbf{x}_i\}_{i=1}^{M}$ sampled from $\rho(x;S,\mathcal{P})$. For sampling from a cgDNA+ probability density function we can take advantage from the sparsity pattern of the cgDNA+ stiffness matrix K. In fact one can observe that the Cholesky factorisation of a block diagonal matrix preserve the block structure, i.e, let $K = L^T L$ the Cholesky factorisation of the cgDNA+ stiffness matrix K, where L is a upper triangular sparse matrix whose sparsity pattern correspond to the half, with respect to the main diagonal, of the one of K. Please check by your self the latter statement. We can use now the Cholesky factorisation to perform the following change of variable:

$$\begin{aligned}\rho(\mathbf{x};S,\mathcal{P}) &= \frac{1}{Z}\exp\left\{\frac{1}{2}(\mathbf{x}-\hat{\mathbf{x}})\cdot K(\mathbf{x}-\hat{\mathbf{x}})\right\}\\ &= \frac{1}{Z}\exp\left\{\frac{1}{2}(\mathbf{x}-\hat{\mathbf{x}})\cdot L^T L(\mathbf{x}-\hat{\mathbf{x}})\right\}\\ &= \frac{1}{Z}\exp\left\{\frac{1}{2}L(\mathbf{x}-\hat{\mathbf{x}})\cdot L(\mathbf{x}-\hat{\mathbf{x}})\right\},\end{aligned}$$

define $\mathbf{y} = L(\mathbf{x}-\hat{\mathbf{x}}) \in \mathbb{R}^{24n-18}$, to get the following distribution

$$\rho(\mathbf{y};S,\mathcal{P}) = \frac{1}{Z}\exp\left\{\frac{1}{2}\mathbf{y}\cdot\mathbf{y}\right\} = \frac{1}{Z}\prod_{i=1}^{24n-18}\exp\left\{\frac{1}{2}y_i^2\right\} \tag{1}$$

Finally we can sample $\mathbf{y}$ component wise from an uni-dimensional standard normal distribution $\mathcal{N}(0,1)$ and recompute the configuration $\mathbf{x}$ by solving the sparse system

$$\mathbf{x} = L^{-1}\mathbf{y} + \hat{\mathbf{x}}. \tag{2}$$

1. In Fig. (1) we plot only 250 configuration samples from the cgDNA+ distribution for $\text{poly}(AA)_{50}$ and $\text{poly}(AT)_{50}$ using the matlab function `mvnrnd`. You can use script `main_mc.m` given in the cgDNA+ matlab package in order to generate Fig. (1). To understand the use of `mvnrnd` see matlab function `cgDNAp_MonteCarlo.m` provided in cgDNA+ matlab package. We stress that even if the groundstate is remarkably straight, few sampled configurations are incredibly bended. Moreover one can visualise that $\text{poly}(AA)_{50}$ is stiffer than $\text{poly}(AT)_{50}$.

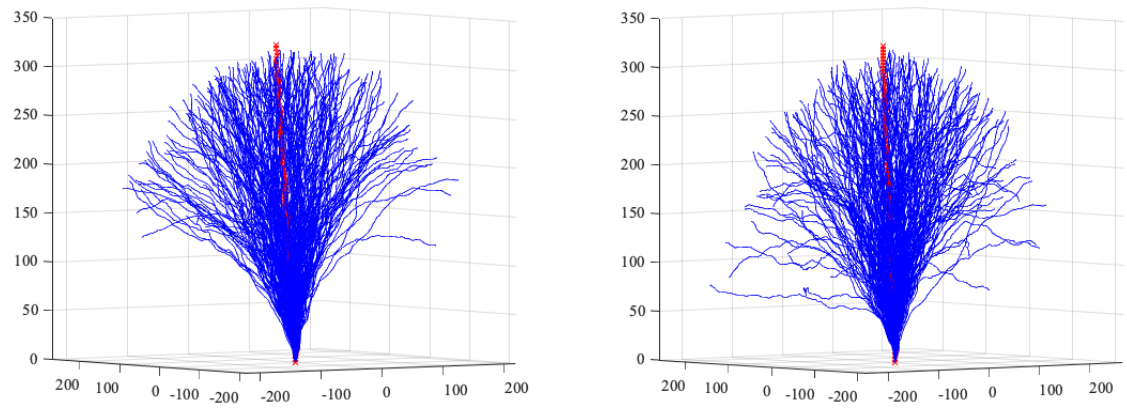

Figure 1: 250 sampled configurations (blue) of poly$(AA)_{50}$ (left figure) and poly$(AT)_{50}$ (right figure) and corresponding ground states (red).

2. Just by plotting the last base-pair position of the 250 sample configurations of Fig. (1) we can observe that the cloud of points seems to define a half sphere like shape around the groundstate as shown in Fig. (2). It is now interesting to ask ourself if this cloud of point is "converged" in the sense that the real distribution of this point is uniform on a half sphere which may, roughly speaking, is the ground state. For doing that we need to sample more configuration to reconstruct the last base–pairs position, thus we need to use the cgDNApmc C++ code because it is much faster then using matlab.

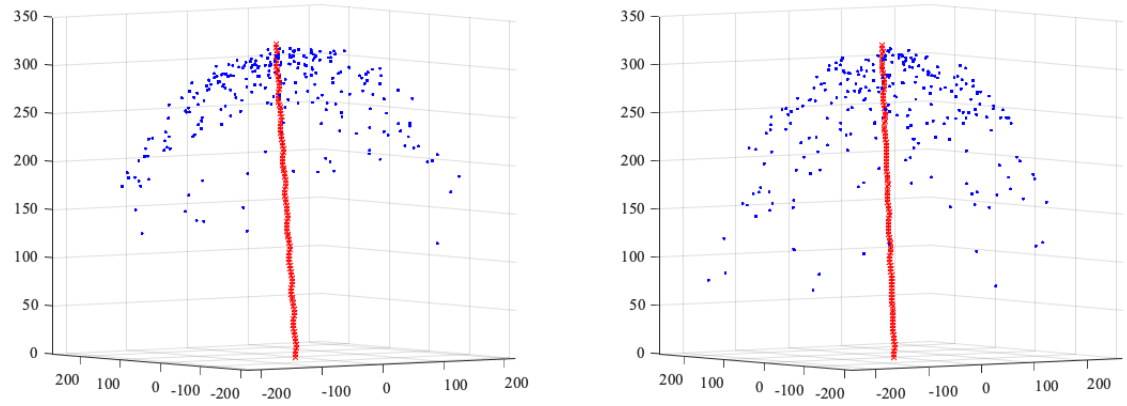

Figure 2: Cloud of points (blue) defined by the last base–pair position (of 250 sampled configurations in Fig. (1)) of poly$(AA)_{50}$ (left figure) and poly$(AT)_{50}$ (right figure) and corresponding ground states (red).

1.2 The cgDNApmc code

Modified run_cgDNAmc

By doing a minor modification in the script run_cgDNAmc_from_seq.cpp one can compute and save all the last base-pairs positions of all the sampled configurations. These modification is already done in script (given inside cgDNApmc code package) mod_run_cgDNAmc_from_seq.cpp. Then by using it one can draw 10^5 configurations of a poly$(\alpha\beta)_{150}$ in a couple of minutes. In Fig. (3) one can observe that the cloud of points of last base–pair positions form almost a sphere which "ray" is

the ground state. The isotropic behaviour of the poly$(AT)_{150}$ is due to his high twisted structure and his almost straight ground state. The two side views, right column in Fig. (3), show a more concentrated region of point which in this two dimensional view seems like a banana shaped region.

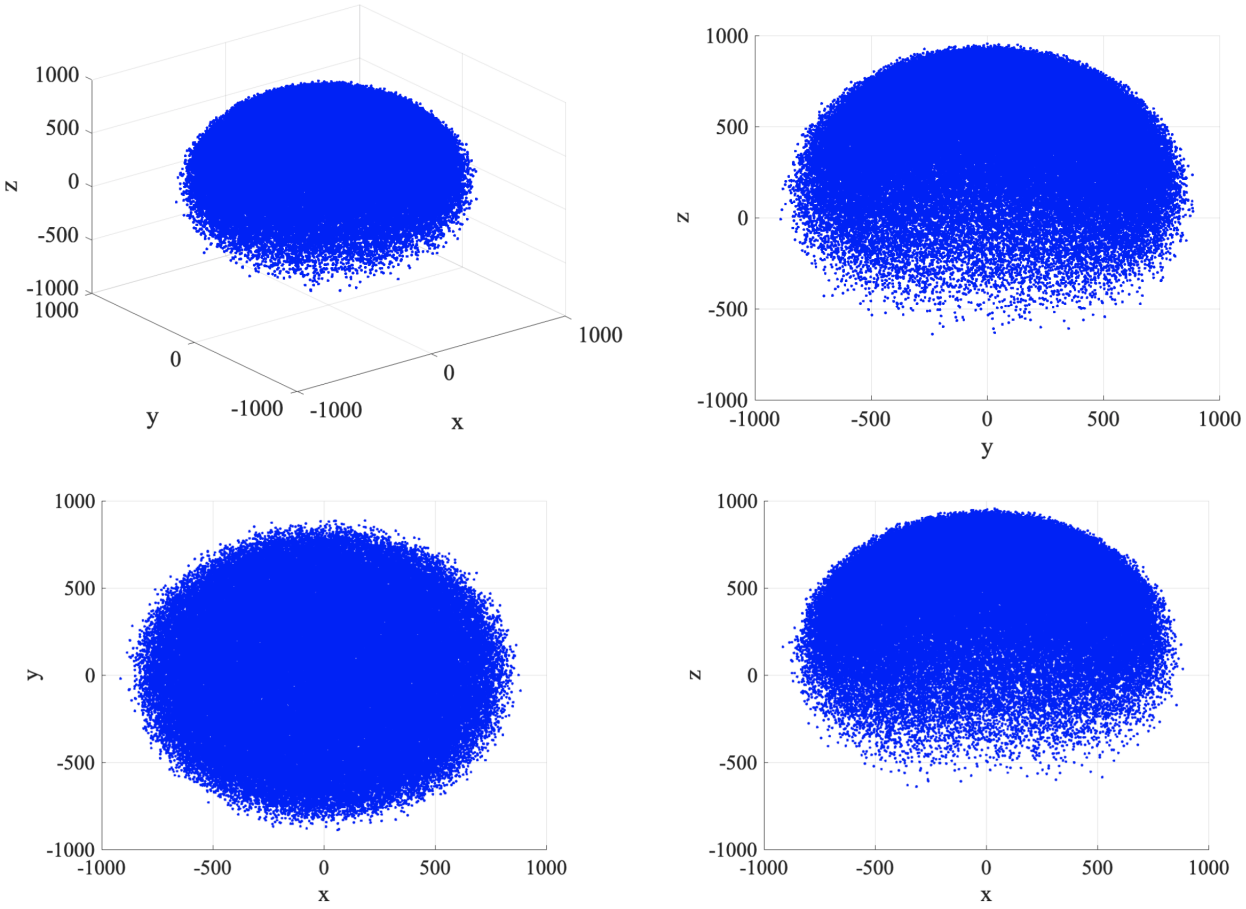

Figure 3: Cloud of 10^5 points (blue) which represent all the last base–pair positions sample using cgDNApmc for poly$(AT)_{150}$. We show here four different views in order to observe the region with the highest concentration of points. In the left column we plotted a global view of all the points (top), and the view of the plane (X,Y), while in the right column we plotted the side views, plane (Z,Y), and plane (Z,X).

This banana shape region is more visible when considering less repetitions of the dinucleotide AT, for example in Fig. (4) we show the cloud of 10^5 points obtained for poly$(AT)_{50}$ and the over all shape looks more like an umbrella, this is due to the fact that poly$(AT)_{50}$ is more rigid than poly$(AT)_{150}$. In the two side views in Fig. (4) the banana shape regions is more evident.

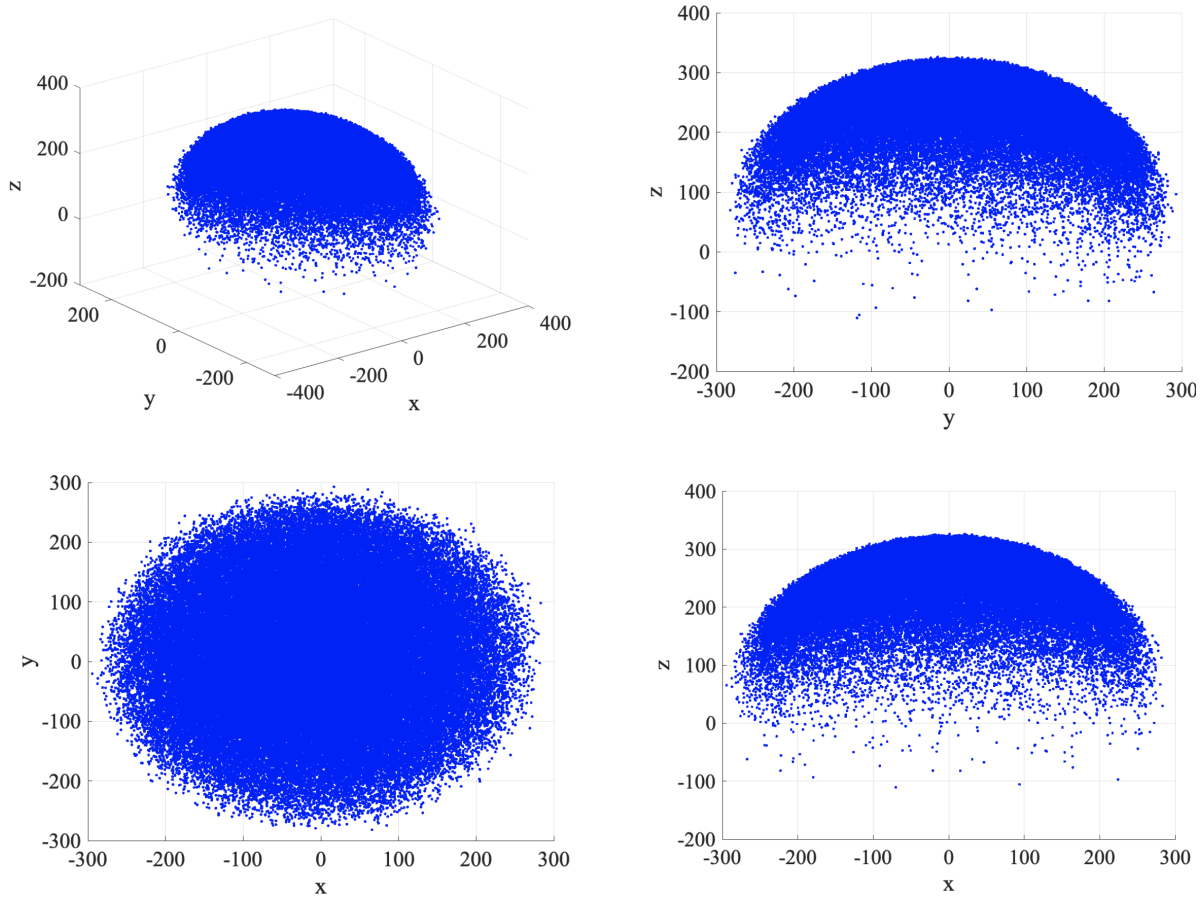

Figure 4: Cloud of 10^5 points (blue) which represent all the last base–pair positions sample using cgDNApmc for poly$(AT)_{50}$. We show here four different views in order to observe the region with the highest concentration of points. In the left column we plotted a global view of all the points (top), and the view of the plane (X,Y), while in the right column we plotted the side views, plane (Z,Y), and plane (Z,X).

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

2.1 cgDNA+ reconstruction of the 5 lambda phage segments

In Fig.(5) one can see the 3D reconstruction of the five lambda sequences. We want to stress that the five groundstates are all different and remarkably bended.

2.2 Monte Carlo simulation of the cgDNA+ model with matlab

In Fig.(6) we plot 250 sampled configurations for λ_2 and poly$(AT)_{150}$. From this plot one can observe that groundstate for λ_2 is more bended in comparison to groundstate of poly$(AT)_{150}$. Moreover one can also observe that rosette for λ_2 is also more bended than rosette of poly$(AT)_{150}$.

2.3 Cloud of points of last base-pair positions

In Fig. (7–9) we plot the cloud of points for λ_1, λ_2, and λ_4. Here it is interesting to see that each cloud has a different shape of the most concentrated region which is strongly correlated to its ground state. In all the three cases we can remark an anisotropic behaviour of all the DNA fragments.

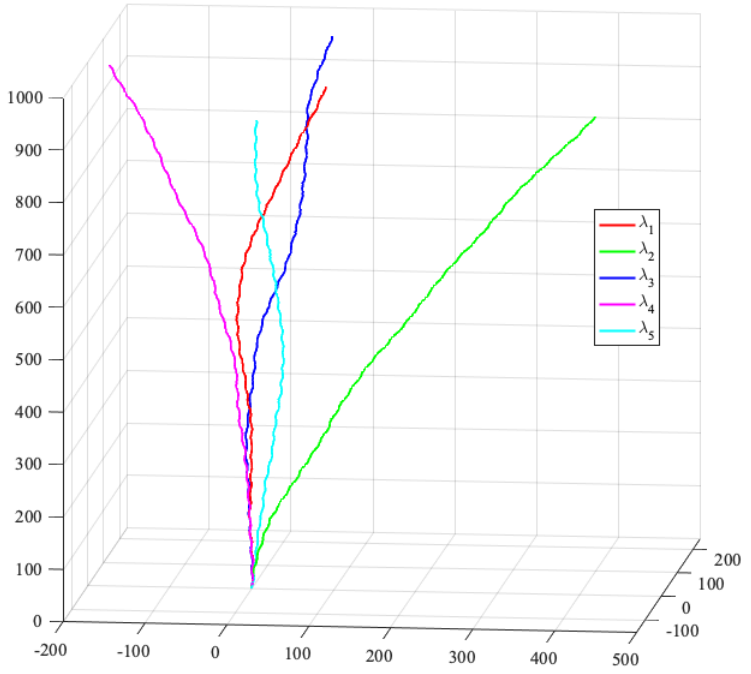

Figure 5: 3D reconstruction of the groundstates of each λ_i, $i = 1, \ldots, 5$.

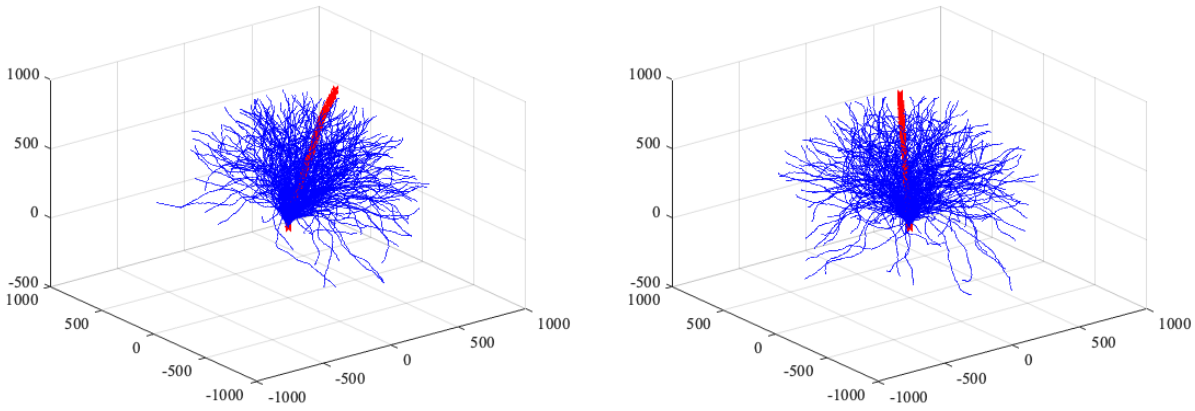

Figure 6: 250 sampled configurations (blue) of λ_2 (left figure) and poly$(AT)_{150}$ (right figure) and corresponding groundstates (red).

3 Effect of the sparsity on the Monte Carlo simulation efficiency

i) For the Spectral decomposition one can use the matlab command $[M, \Lambda]$ =`eigs`(K), where K is the cgDNA+ stiffness matrix. Be careful that the function `eig` does not take as argument sparse matrices. For the Cholesky factorisation the matlab function is `chol`. Notice from Fig.(10) that Spectral decomposition is computationally very expensive for longer sequences.

ii) To get the inter's marginal distribution you have to compute the marginal mean and the marginal covariance. To get the marginal mean you have to select all the inter–base–pair

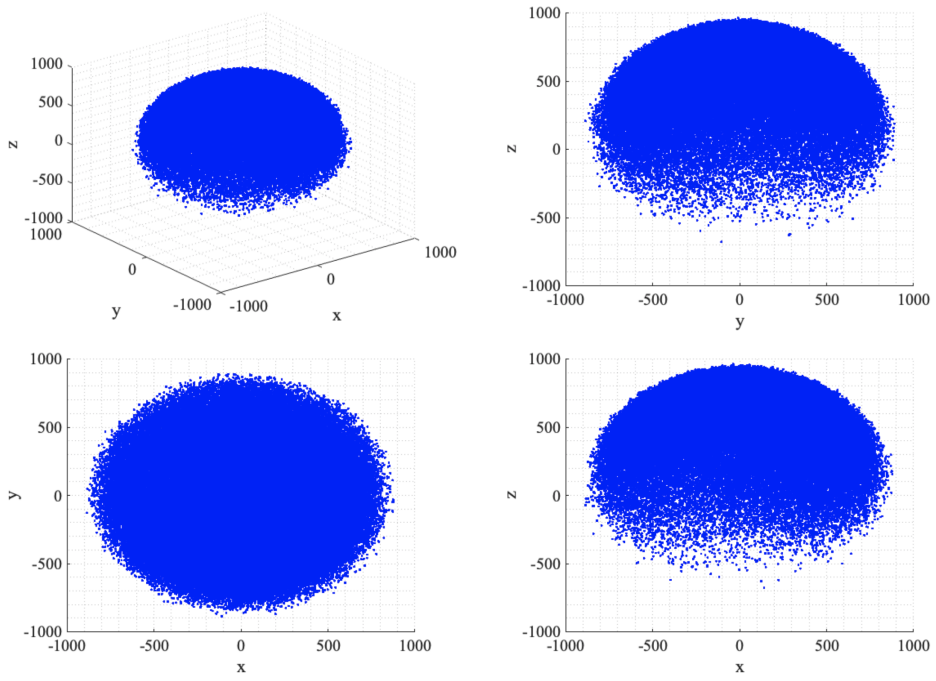

Figure 7: Cloud of 10^5 points (blue) which represent all the last base–pair positions sampled using cgDNApmc for λ_1. We show here four different views in order to observe the region with the highest concentration of points. In the left column we plotted a global view of all the points (top), and the view of the plane (X,Y), while in the right column we plotted the side views, plane (Z,Y), and plane (Z,X).

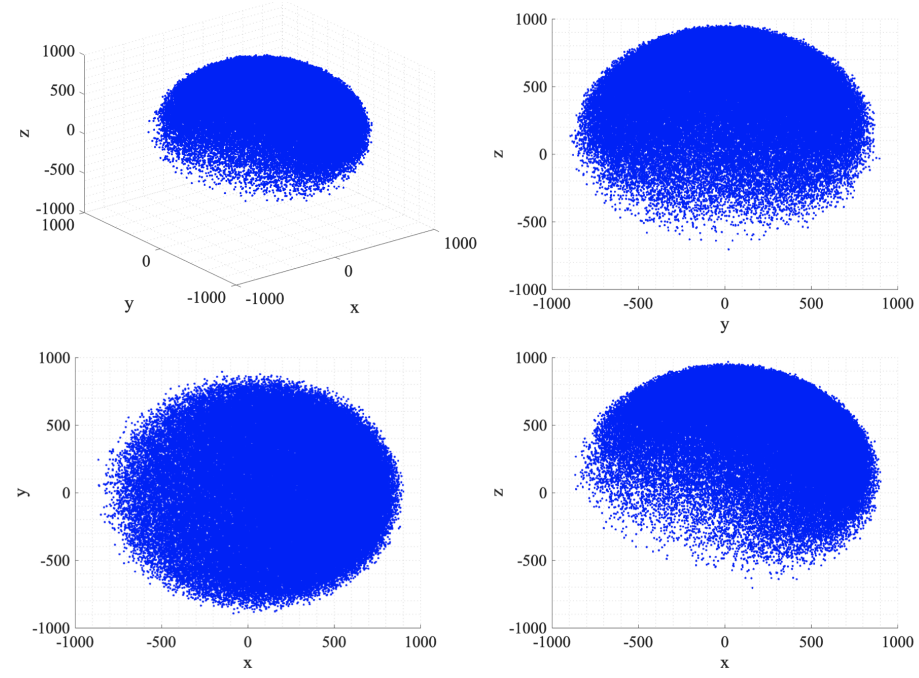

Figure 8: Same as Fig.(7) but for λ_2

components from the cgDNA+ mean. A same approach has to be used to get the marginal covariance. Starting from the cgDNA+ stiffness computes its covariance (by computing the inverse) and select only the inter–inter correlation entries of the matrix. One should obtain a

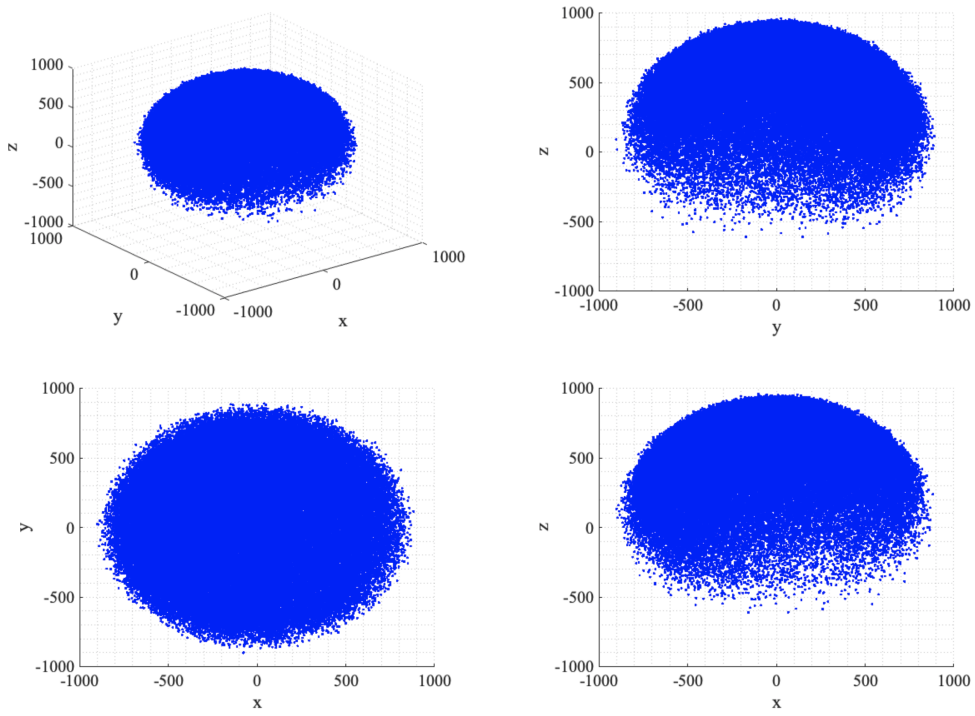

Figure 9: Same as Fig.(7) but for λ_4

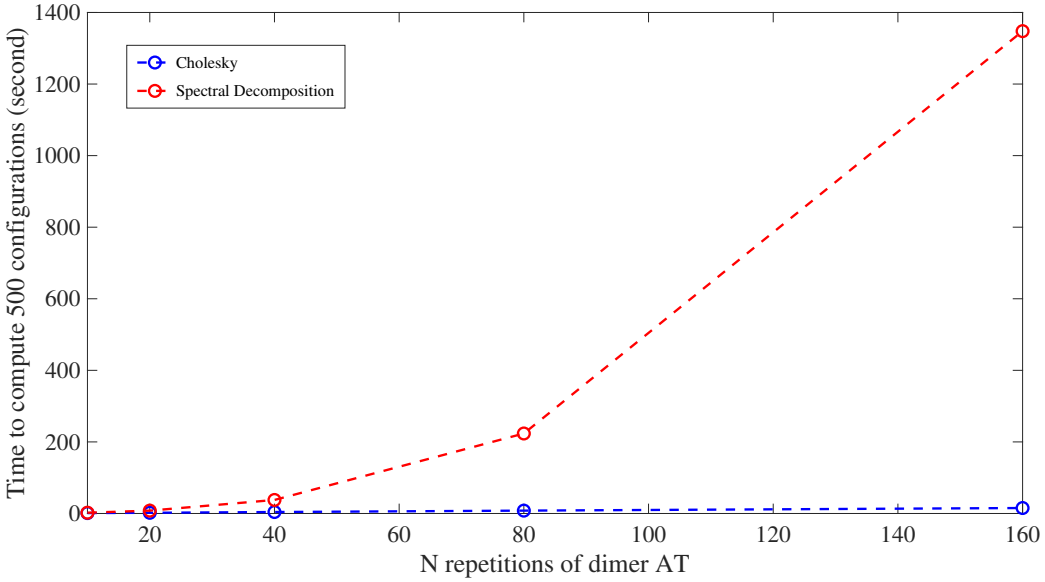

Figure 10: Comparison of Matrix decomposition method: We compare the Cholesky decomposition (in blue) and Spectral decomposition (in red) for sampling 500 configurations. We tested the performance of the two methods for different poly$(AT)_N$.

dense marginal covariance matrix with dense inverse matrix. In Fig.(11), time taken to sample 500 configurations from cgDNA+ distribution and from marginal distribution are shown.

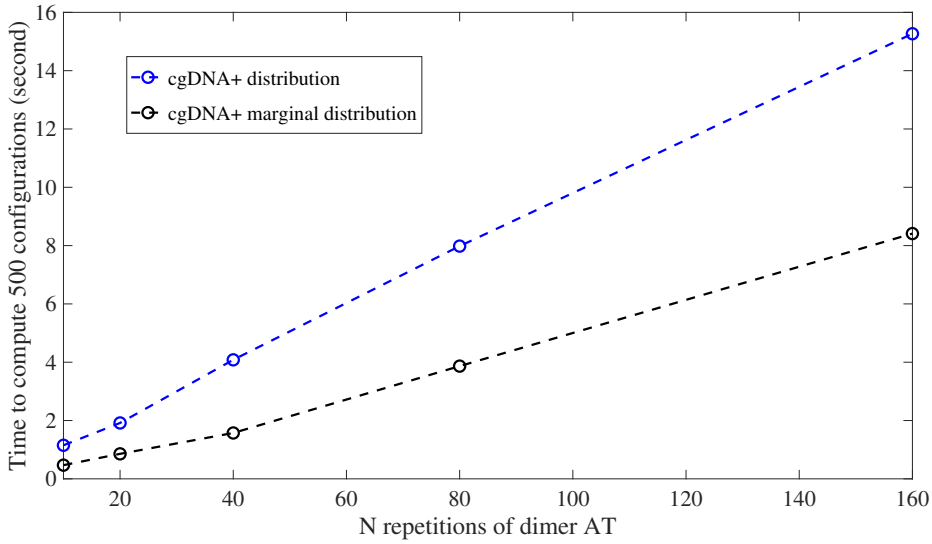

Figure 11: Comparison between distributions: We compare the performance of the Monte Carlo method when the whole cgDNA+ distribution is considered (in blue) and when the marginal distribution of the inter is used (in black). We sample 500 configurations for each method and tested the performance for different poly$(AT)_N$.