

Faculty of Physics and Astronomy
University of Heidelberg

Diploma thesis in Physics

submitted by
Evrin Ilden
born in Heidelberg
2007

Analysis of the chromatin fiber conformation by Monte Carlo
simulations

This diploma thesis has been carried out by Evrim Ilden at the
Kirchhoff Institute of Physics
under the supervision of
PD Dr. Karsten Rippe
and
Prof. Dr. Dr. Christoph Cremer

Analyse der Chromatinfaserkonformation durch Monte Carlo Simulationen:

Im Zellkern von Eukaryoten wird die DNA durch Histon-Proteine in eine Kette von Nucleosomen gepackt, in denen 146 oder 147 Basenpaare von DNA um Histon-Oktamer Proteinkerne gewickelt sind. Diese einer Perlenkette ähnliche Struktur assoziiert unter physiologischen Bedingungen in eine kondensierte Chromatinfaser mit einem Durchmesser von 30 nm. Der Effekt der DNA-Nucleosomen-Geometrie auf die Konformation dieser Chromatinfaser wurde mit Monte-Carlo-Simulationen unter Verwendung des Metropolis-Algorithmus untersucht. Ein bereits entwickeltes „zwei-Winkel-Modell“ für die Konformation der DNA am Nucleosom wurde verwendet, um den Effekt der Nucleosomen-Geometrie auf die Organisation der Faser bei physiologischer Ionenstärke zu untersuchen. Eine Reihe von Fasern mit je 100 Nucleosomen wurde erstellt, bei denen zwei zentrale Parameter, der DNA Eintritts-Austrittswinkel und die Länge der Verbindungs-DNA systematisch variiert wurden. Es konnte gezeigt werden, dass kleine Änderungen der Winkel, die den DNA-Pfad bestimmen, zu überraschend grossen Änderungen von einer kompakten Faserkonformation mit hoher Massenbelegungsdichte zu einer ausgedehnten Konformation führen. Dieser Übergang könnte mit der Bildung von Zuständen zusammenhängen, bei denen der Zugang der DNA für biologische Prozesse erleichtert wird.

Analysis of the chromatin fiber conformation by Monte Carlo simulations:

In the cell nucleus of eukaryotes the DNA is packaged by histone proteins into a chain of nucleosomes, in which 146 or 147 base pairs of DNA are wrapped around octameric histone protein cores. This “beads on a string” like structure associates under physiological conditions into a condensed chromatin fiber with a diameter of about 30 nm. The effect of the DNA-nucleosome geometry on the conformation of these chromatin fibers was studied by Monte Carlo simulations using the Metropolis algorithm. A previously developed “two angle” model for the conformation of the DNA at the nucleosome was used to explore the effect of the nucleosome geometry on the organization of the chromatin fiber at physiological ionic strength. A series of fibers with 100 nucleosomes were created for which two of the central parameters, the DNA entry-exit angle and the linker DNA length, were systematically varied. It is shown that small changes in the angles that determine the DNA paths lead to surprisingly large changes from a compacted fiber conformation with high mass density into an extended conformation. This transition might be related to the formation of a state where access to the DNA is facilitated for biological processes.

Contents

1	Introduction.....	3
1.1	Biological Background	5
1.1.1	DNA	5
1.1.2	Chromatin.....	7
1.2	Numerical Simulations of chromatin	10
2	Method	13
2.1	The Dynamic Cross Linker model	13
2.1.1	Discretization	14
2.1.2	Elastic Energies.....	15
2.1.3	Electrostatic interactions of DNA	16
2.1.4	Internucleosomal interactions	17
2.2	Monte Carlo Simulation using the Metropolis Algorithm	18
2.2.1	Simple Sampling	19
2.2.2	Importance Sampling	20
2.2.3	Metropolis Algorithm	20
2.2.4	Monte Carlo Steps.....	24
2.3	Tools for the evaluations.....	25
3	Results.....	29
3.1	Initial observation and motivation	32
3.2	Simulations with higher resolution	36
3.3	Comparing versions of the simulation software	40
3.4	Influence of the stem.....	43
4	Discussion	49
5	Literature.....	55
6	Appendix.....	59

1 Introduction

The large amount of genetic information of a cell is contained in the linear order of the nucleotides in the DNA. In recent times we have gained some knowledge about the way this information is copied, its data integrity is ensured and the information is coded. Also the principles of how the information is retrieved and then translated into sequences of amino acids are understood. So the next challenge has been to find out more about the highly complex mechanisms of how all these processes are regulated. What we know is that regulation may take place at any step of gene expression - from the DNA-RNA transcription step to post-translational modifications of a protein.

One additional layer of the control of gene expression is the organization of DNA in a protein-DNA complex called chromatin [1]. The histone proteins in this complex can compact the DNA chain of ~2 m in humans so it can get packed into a nucleus with 10 μm diameter. In addition the cell must also be able to selectively grant access to genes for those DNA binding factors that are involved in processes of replication, transcription, repair and recombination. Thus, the histone proteins have to accomplish the somewhat contradictory task of both compacting and opening specific regions of the DNA in a defined manner.

In fact the organization of chromatin in the cell is an important factor for the regulation of transcription. For a number of genes it has been shown that chromatin conformation and gene expression are directly linked [2, 3]. The octameric protein complex of histones H2A, H2B, H3 and H4 all in two copies is responsible for the packaging of the DNA, and the histone complexes can be modified by acetylation, methylation and phosphorylation of certain amino acids [1]. These kinds of modifications are considered to be important regulators of DNA packaging and with it the gene expression levels.

Furthermore, changes to these factors that affect the chromatin conformation have been related to several forms of cancer and neurodegenerative diseases [2, 3]. It is also essential for the expression of recombinant genes for experimental or therapeutic purposes that the relevant chromatin region is in a transcriptionally competent state. There is, therefore, an urgent need to derive a complete understanding of the relationship between chromatin organization and gene expression.

When having a look at the levels of the chromatin organization we see that the histones together with DNA wrapped around them (together constituting a so called nucleosome) lead to a "beads on a string" structure which in turn coils into a 30 nm diameter helical structure known as the 30 nm fiber or filament. So the nucleosome is the basic building block of the chromatin fiber and changes of its conformation are likely to modulate the organization of the whole fiber. However, the structural principles that come into play during the transition from the loose chain of nucleosomes to a compact 30 nm chromatin fiber have been difficult to establish, and the arrangement of nucleosomes and linker DNA in condensed chromatin fibers has not been fully resolved.

One way of trying to find an answer is through numerical modeling the 30 nm chromatin fiber. The size of the molecule and its complexity does not allow for atom-scale molecular dynamics and therefore any model of the chromatin fiber is necessarily "coarse-grained". The zig-zag-structure of nucleosomes and DNA can be modeled with elastically connected beads and their configurations in thermal equilibrium can be sampled using a Monte Carlo algorithm.

The aim of this thesis was it to reveal local properties of the chromatin fiber as well as its higher order structural organization by building on an established simulation software [4, 5]. It was examined how the organization of the fiber was dependent on the local nucleosome geometry. We hoped to identify mechanisms of DNA compaction which are based on geometrical transformations of the

nucleosome that could be related to structural changes of the fiber geometry. These are of potential relevance for the regulation of gene expression.

1.1 Biological Background

The genome of a human cell comprises 6×10^9 DNA base pairs corresponding to a DNA chain with a total length of about two meter. Free in solution a DNA duplex of this length would occupy the volume of a sphere with a diameter of approximately 400 μm as estimated from its calculated radius of gyration. This amount of DNA is packaged into a nucleus with a typical diameter of 10-20 μm . The required compaction of the genome is obtained via complexation of the DNA with small strongly positively charged proteins, the histones, into a large nucleoprotein complex that is referred to as chromatin (Figure 1.1).

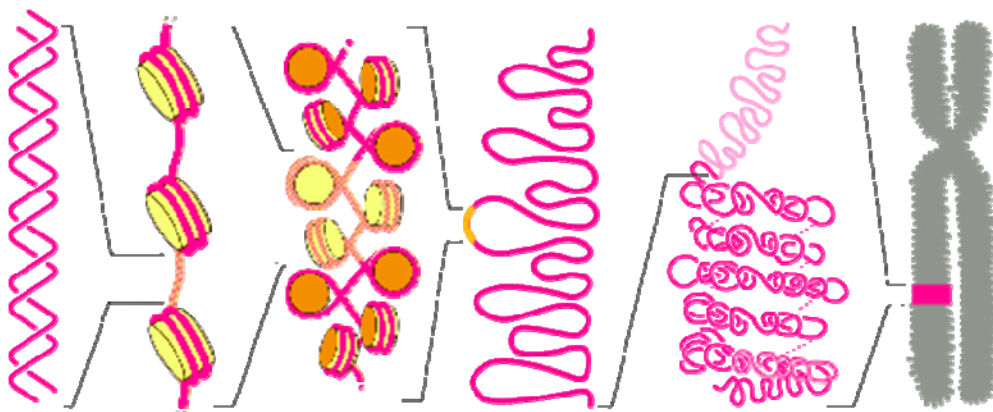


Figure 1.1: Different levels of chromatin packing. From left to right: The free DNA double helix becomes the 10-11 nm wide ‘beads-on-a-string’ form when coiled around the nucleosome cores. This string is then coiled into a fiber of about 30 nm diameter. In the higher order organization chromatin is folded into loops and further into distinguished domains which are subunits of the (in this case metaphase) chromosome.

1.1.1 DNA

DNA is a nucleic acid molecule that contains the instructions needed to construct other components of cells, such as proteins and RNA molecules. The main role of DNA is the long-term storage of information. The DNA segments that carry this

genetic information are called genes. Other DNA sequences have structural purposes or are involved in regulating the use of this genetic information.

DNA is a long polymer made from repeating units called nucleotides. The DNA chain is 2.2 to 2.4 nanometers wide, and one nucleotide unit is 0.33 nanometers long [6]. Although each individual repeating unit is very small, DNA polymers can be enormous molecules containing millions of nucleotides. For instance, the largest human chromosome, chromosome number 1, is 220 million base pairs or ~7.5 cm long, when fully extended.

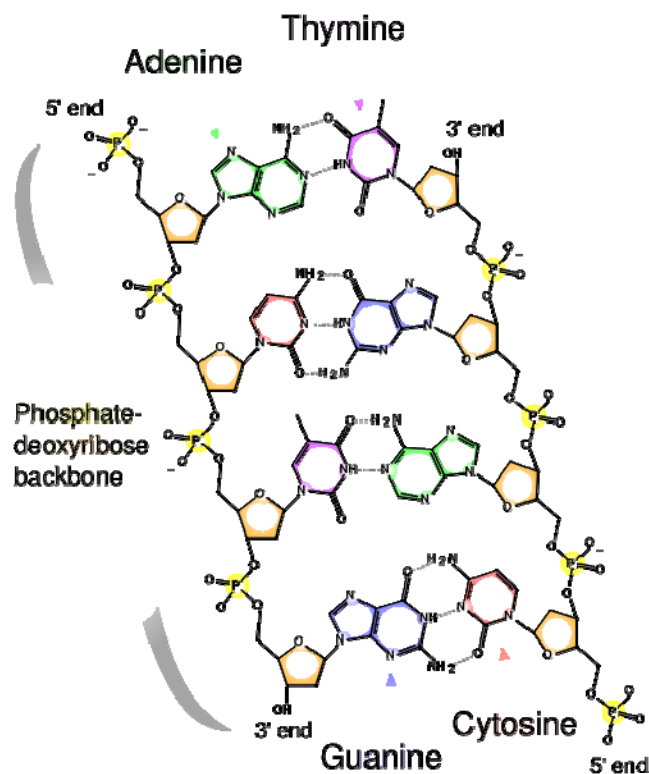


Figure 1.2: The chemical structure of the DNA double helix. Four complementary base pairs are shown. The two strands run in opposite directions. Each base pair is held together by either two or three hydrogen bonds.

In living organisms, DNA does not usually exist as a single molecule, but instead as a tightly-associated pair of two chains. These two long strands intertwine, in the shape of a double helix [1]. The nucleotide repeats contain both the segment of

the backbone of the molecule, which holds the chain together, and a base, which interacts with the other DNA strand in the helix (Figure 1.2). In general, a base linked to a sugar is called a nucleoside and a base linked to a sugar and one or more phosphate groups is called a nucleotide. The backbone of the DNA strand is made from alternating phosphate and sugar residues. The sugar in DNA is 2-deoxyribose, which is a pentose (five carbon) sugar. The sugars are joined together by phosphate groups that form phosphodiester bonds between the third and fifth carbon atoms of adjacent sugar rings. These asymmetric bonds mean a strand of DNA has a direction. In a double helix the direction of the nucleotides is antiparallel. The asymmetric ends of a strand of DNA bases are referred to as the 5' (five prime) and 3' (three prime) ends (also Figure 1.2).

The DNA double helix is stabilized by hydrogen bonds between the bases attached to the two strands. The four bases found in DNA are adenine (abbreviated A), cytosine (C), guanine (G) and thymine (T).

The double helix is a right-handed spiral. As the DNA strands wind around each other, they leave gaps between each set of phosphate backbones, revealing the sides of the bases inside. There are two of these grooves twisting around the surface of the double helix: one groove, the major groove, is 22 Å wide and the other, the minor groove, is 12 Å wide.

1.1.2 Chromatin

The basic repeat element of chromatin is the nucleosome, linked by sections of linker DNA. The nucleosome consists of 1.67 turns of DNA (146 or 147 base pairs), which are wrapped around the histone octamer complex in a left-handed superhelix [7-9]. There are four different types of core histone proteins which form the octamer, which is made up of two copies each of H2A, H2B, H3 and H4. The structure of the free histone octamer and that of the nucleosome complex have been determined by high resolution x-ray diffraction [7-9]. The nucleosome has a cylindrical shape with a diameter of about 11 nm and a height of 5.5 nm.

This small diameter around which the DNA bends is far smaller than can be reached by DNA in solution.

The nucleosomes bind DNA non-specifically, as required by their function in general DNA packaging. There is, however, some preference in the sequences the nucleosomes will bind. This is largely through the properties of DNA, adenosine (A) and thymine (T) bases are more favorably compressed into the inner minor grooves. This means nucleosomes bind preferentially at one position every 10 base pairs - where the DNA is rotated to maximize the number of A and T bases which will lie in the inner minor groove (Figure 1.3).

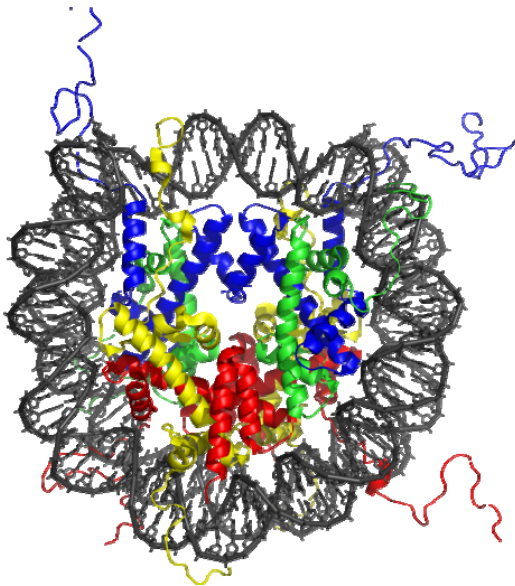
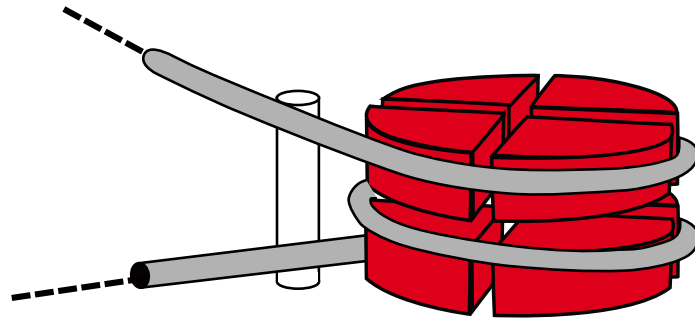


Figure 1.3: The crystal structure of the nucleosome core particle consisting of the histones H2A (yellow) , H2B (red) , H3 (blue) and H4 (green), two of each. Minor grooves of the DNA double helix are compressed on the inside of the turn.

A fifth histone, the so called linker histone H1 or H5, binds to the nucleosome core complex [10-13] and contacts the exit/entry of the DNA strand on the nucleosome. Although its exact binding site and orientation is still unknown, it seems to affect the trajectory of the DNA at the nucleosome and a structure derived from modeling has been published recently [14]. A nucleosome with one bound linker histone or in other words a histone octamer, one molecule of H1 and

166 bp of DNA is known as a chromatosome (Figure 1.4). Chromatosomes, connected by about 20 to 60 base pairs of linker DNA, form an approximately 10nm "beads-on-a-string" fiber.

Figure 1.4: Schematic of a chromatosome. Histone H1 determines the way the incoming and outgoing DNA is attached to the nucleosome core complex consisting of 8 histones.



The "beads-on-a-string" structure in turn coils into a 30nm diameter helical structure known as the 30nm fiber or filament [15]. The precise structure of the chromatin fiber in the cell is not known in detail, and there is still some debate over the exact structure.

This level of chromatin structure is thought to be the form of euchromatin, which contains actively transcribed genes. Electron microscopy studies have demonstrated the 30 nm fiber is highly dynamic such that it unfolds into a 10 nm fiber ("beads-on-a-string") structure when transversed by a RNA polymerase engaged in transcription.

In most of the fiber models the nucleosomes lie perpendicular to the axis of the fiber, the linker histones lie on the inside of the structure and the chain can unwind into the 10nm "beads-on-a-string" fiber [1].

The formation of a 30 nm fiber requires the regular positioning of nucleosomes along the DNA. The mechanical properties of DNA are such that the linker DNA is relatively resistant to bending and rotation. This makes the length of linker

DNA critical to the stability of the fiber. Nucleosomes must be separated by linker DNA which allows them to rotate and then fold into the required orientation without significant stress to the DNA.

Several lines of evidence indicate that the chromatin fiber adopts a higher order folding that organizes the interphase chromosome into domains containing roughly 1 MB of DNA [16, 17]

1.2 Numerical Simulations of chromatin

The first model describing the 30 nm fiber as a solenoid structure [18] was derived from low resolution diffraction data and many other models have been suggested since then on the basis of various types of experiments (Table 1.1).

	Model	Reference	Experimental Data
1	solenoid	[18]	x-ray scattering
2	zig-zag	[19]	cryo-electron microscopy
3	zig-zag	[20]	scanning force microscopy
4		[21]	analytical ultracentrifugation data
5		[22]	stretching single chromatin-molecules
6	DCL	[5]	linear mass density, persistence length, linear dichroism

Table 1.1: Models of the chromatin fiber conformation and the experimental data they explain. For older models see ref. [1]

The DCL model (Dynamic Cross Linker model, Table 1.1 entry 6), developed by Gero Wedemann [4] seems at the moment to be consistent with a number of different experimentally derived parameters for the chromatin fiber and is also in agreement with a recently presented crystal structure of a tetranucleosome [23].

Chromatin fiber models 1 to 3 in Table 1.1 are static models, while models 4 to 6 describe the fiber in thermal equilibrium. Model 4 is based on Brownian Dynamics simulations while model 5 and 6 involve Monte Carlo simulations. For larger systems the Brownian Dynamics approach is computationally too

expensive. Models 4 and 5 describe the nucleosomes as spheres. Since spheres can not be packed as well as cylinders, these models fail to reproduce the experimentally observed linear mass density. The DCL model describes the nucleosomes as oblate ellipsoids which is a better approximation of the true nucleosome shape. While model 5 computes the electrostatic interaction of the linker DNA by a Lennard-Jones potential, models 4 and 6 calculate this interaction explicitly by a Debye-Hückel potential including effects of the orientation of the DNA segments to each other and of different ionic strength of the solvent.

2 Methods

2.1 *The Dynamic Cross Linker model*

The Dynamic Cross Linker (DCL) model describes the chromatin fiber as a polymer with cylindrical units for DNA and oblate ellipsoids for chromatosomes [4, 5]. The units interact by stretching, torsion and bending forces modeled by harmonic potentials. The interaction of DNA by its negatively charged phosphate backbone is described by a Debye-Hückel potential, the interchromosomal interaction is described by a Gay-Berne-potential. Data from experiments are used to parameterize these potentials.

The geometry of the connection between the DNA and the chromatosome is sketched in Figure 2.1. We assume that the linker DNA is either connected directly to the chromatosome (defining the chromatosome as a histone core with exactly two superhelix turns) or alternatively to a nucleosome stem, which is rigidly coupled to the rest of the chromatosome (and Figure 3.3). In the following, we first analyze the situation where the linker DNA is connected via a stem (Figure 2.2). The entry and exit points of the DNA have a vertical distance (in the direction of the cylinder axis) of 3.1 nm and a radial distance of $d = 8$ nm from the symmetry axis of the core particle [24].

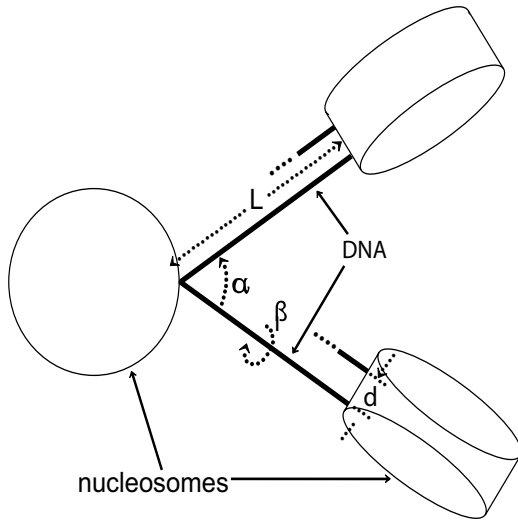


Figure 2.1: The geometric parameters that describe the model. These are the DNA entry-exit angle α , the twisting angle between two nucleosomes (or respectively chromatosomes if linker Histone H1 is bound) β and the linker DNA length L .

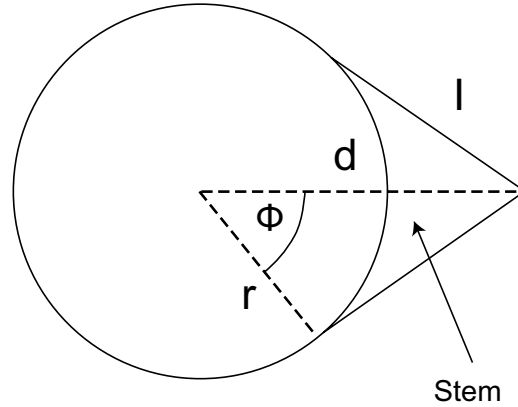


Figure 2.2: The core particle contains 146bp DNA in a superhelix with 1.65 turns, thus two turns correspond to 177bp. The nucleosome stem ends at a distance of $d=8\text{nm}$ from the center of the core particle

2.1.1 Discretization

DNA that is not bound to the chromatosome is described by segments. They are chosen much smaller than the persistence length, so they are approximately straight. For linker DNA which is 3.4 nm long or shorter, one segment was used to connect the chromatosomes. For longer linker DNAs, the length was divided into two segments. The position of each segment is determined by a position vector \vec{r}_i and a local coordinate system (f_i, u_i, v_i) describing its orientation in space. The coordinate vector \vec{u}_i corresponds to segment vector $\vec{u}_i = \vec{r}_{i+1} - \vec{r}_i$. The segment length is $b_i = u_i$

The position of a chromosome is described by a vector to its center of mass and a local coordinate system where \vec{u}_i points from the center of mass to the geometrical center of connection points of the linker DNA, and \vec{v}_i is parallel to the axis of the nucleosome core.

Linker DNA is coupled to the chromosome at two distinct points at a distance d . A chromosome therefore forms a short segment by itself. The distance of the two DNA coupling points to the symmetry axis of the cylinder is d . If the linker DNA is coupled directly to the nucleosome, the value of d is equal to the radius of the nucleosome. In the case of a nucleosome stem this radius is larger. Figure 2.3 and Figure 3.3 demonstrates the difference between the two ways DNA segments and nucleosomes can be connected.

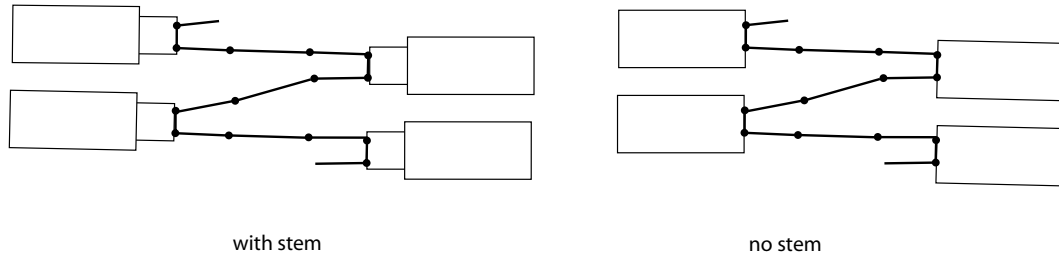


Figure 2.3: A Schematic showing the difference in the model of having a stem motif or not.

2.1.2 Elastic Energies

The potentials of the elastic interactions are assumed to be harmonic. The strength constants of the interactions are named $\alpha_{(Y)}^{(X)}$ in which X denotes the type of interaction, with s for stretching, b for bending and t for torsion. Y denotes the type of interacting partners, DNA or nucleosome.

The stretching potential of one segment is computed as

$$(2.1) \quad U_{\text{stretch}} = \frac{\alpha_Y^{(s)}}{b_i^0 (b_i - b_i^0)^2}$$

The bending angle is computed between the segment vector \vec{u}_{i+1} and its equilibrium position \vec{B}_i relative to the connected segment. \vec{B}_i is defined by the angles θ_i and ϕ_i .

$$(2.2) \quad \vec{B}_i = \vec{f}_i \sin \theta_i \cos \phi_i + \vec{v}_i \sin \theta_i \sin \phi_i + \vec{u} \cos \theta_i$$

In the case of no intrinsic bending $\theta_i = \phi_i = 0$ thus $\vec{B}_i = \vec{u}_i$. The bending potential is then

$$(2.3) \quad U_{\text{bend}} = \frac{\alpha_Y^{(b)}}{b_i^0 \theta_i^2}$$

The coordinate systems of the two connected segments i and $i+1$ can be mapped on each other by an Euler-transformation with the angles $(\alpha_i, \beta_i, \gamma_i)$ in which the rotation of the first angle is around \vec{u}_i . Then $\alpha_i + \gamma_i - \tau_i$ is the total torsion. τ_i is the intrinsic torsion. The torsion potential is then

$$(2.4) \quad U_{\text{tors}} = \frac{\alpha_Y^{(t)}}{b_i^0 (\alpha_i + \gamma_i - \tau_i)^2}$$

2.1.3 Electrostatic interactions of DNA

The electrostatic interaction between free DNA double helix segments is described by integrating the solution of the Debye-Hückel equation for a point charge over two charged line segments.

$$(2.5) \quad E_{ij}^{(e)} \text{ tors} = \frac{v^2}{D} \int d\lambda_i \int \frac{e^{-\kappa r_{ij}}}{r_{ij}}$$

D is the dielectric constant of water, and κ the inverse of the Debye length. r_{ij} is the distance between the current positions at the segments to which the integration parameters λ_i, λ_j correspond. The charge per unit length v is chosen such that the potential at the radius of the DNA coincides with the solution of the Poisson-

Boltzmann equation for a cylinder with charge per length v_0^* . For DNA in the presence of the Gouy layer of immobile counterions, this can be computed as $v_0^* = q v_0$ in which $v_0^* = -2e/\Delta$ is the charge per length of the naked DNA [25] with e as the proton charge, and $\Delta = 0.34$ nm as the distance between base pairs. Following Stigter, the value of q is 0.73 [26]. To save computation time, a tabulation of the double integral (2.5) is used. The table is parameterized by the distance of the segments and three values describing its relative orientation. During the simulation a linear interpolation of the tabulated values was used.

2.1.4 Internucleosomal interactions

The internucleosomal interaction is modeled by a Gay-Berne potential with modifications by Kabadi [27-29]. The interaction energy of two particles with a center-to-center distance \vec{r} is

$$(2.6) \quad V(\hat{u}_1, \hat{u}_2, \hat{r}) = 4 \varepsilon(\hat{u}_1, \hat{u}_2, \hat{r}) \left(\left(\frac{\sigma_0}{r - \sigma(\hat{u}_1, \hat{u}_2, \hat{r}) + \sigma_0} \right)^{12} - \left(\frac{\sigma_0}{r - \sigma(\hat{u}_1, \hat{u}_2, \hat{r}) + \sigma_0} \right)^6 \right)$$

The vectors \hat{u}_i and \hat{u}_2 point into the direction of the symmetry axis of the particles. σ_0 scales the potential width, and ε_0 scales the potential depth. The parameter χ defines the anisotropy of the potential width, χ' defines the anisotropy of the potential depth.

$$(2.7) \quad \sigma(\hat{u}_1, \hat{u}_2, \hat{r}) = \sigma_0 \left(1 - \frac{1}{2} \chi \left\{ \frac{(\hat{r}\hat{u}_1 + \hat{r}\hat{u}_2)^2}{1 + \chi(\hat{u}_1\hat{u}_2)} \right\} + \left\{ \frac{(\hat{r}\hat{u}_1 - \hat{r}\hat{u}_2)^2}{1 - \chi(\hat{u}_1\hat{u}_2)} \right\} \right)^{-1/2}$$

$$(2.8) \quad \chi = \frac{(\sigma_{\parallel}^2 - \sigma_{\perp}^2)}{(\sigma_{\parallel}^2 + \sigma_{\perp}^2)}$$

$$(2.9) \quad \varepsilon(\hat{u}_1, \hat{u}_2, \hat{r}) = \varepsilon^v(\hat{u}_1, \hat{u}_2) \varepsilon'^\mu(\hat{u}_1, \hat{u}_2, \hat{r})$$

$$(2.10) \quad \varepsilon(\hat{u}_1, \hat{u}_2) = \varepsilon_0 \left[1 - \chi^2 (\hat{u}_1\hat{u}_2)^2 \right]^{-1/2}$$

$$(2.11) \quad \varepsilon'(\hat{u}_1, \hat{u}_2, \hat{r}) = 1 - \frac{1}{2} \chi' \left[\frac{(\hat{r}\hat{u}_1 + \hat{r}\hat{u}_2)^2}{1 + \chi'(\hat{u}_1\hat{u}_2)} + \frac{(\hat{r}\hat{u}_1 - \hat{r}\hat{u}_2)^2}{1 - \chi'(\hat{u}_1\hat{u}_2)} \right]$$

$$(2.12) \quad \chi' = \frac{(\varepsilon_s^{1/\mu} - \varepsilon_e^{1/\mu})}{(\varepsilon_s^{1/\mu} + \varepsilon_e^{1/\mu})}$$

σ_{\parallel} is the relative potential width for particles oriented parallel and σ_{\perp} for particles oriented orthogonal. ε_s defines the relative potential width for particles in lateral (s, side-to-side) and ε_e in longitudinal (e, end-to-end) orientation. ν and μ are dimensionless parameters; generally one uses $\nu = 1$ and $\mu = 2$ [27]. For other values of ν and μ see [30].

2.2 Monte Carlo Simulation using the Metropolis Algorithm

When examining systems with many degrees of freedom we may describe the set of all possible states or configurations as the parameter space. Monte Carlo simulations use random moves to explore and characterize this parameter space. In statistical physics the expected value $\langle A \rangle$ of an observable A is of interest, in hope to derive a value for a measurable physical property of the observed system. The expectation value of A for a system described by the Hamiltonian H and a temperature T is given by

$$(2.13) \quad \langle A(x) \rangle = \frac{1}{Z} \int e^{-\frac{H(x)}{k_B T}} A(x) dx = \int p(x) A(x)$$

Here

$$(2.14) \quad Z = \int e^{-\frac{H(x)}{k_B T}} dx$$

is the partition function and

$$(2.15) \quad p(x) = \frac{1}{Z} e^{-\frac{H(x)}{k_B T}}$$

is the probability of finding the system in the particular state x at temperature T in thermal equilibrium. The interesting values for this study are the space of geometric configurations C with respective energies $\varepsilon(C)$ and properties $A(C)$.

2.2.1 Simple Sampling

When at each step a different region of the parameter space is sampled by randomly generated independent configurations C_1, C_2, \dots, C_N in a way that they are equally probable, it is referred to as simple sampling. Because in this kind of simulation all random moves are accepted configurations can be generated at random and (2.13) approximated by the expression

$$(2.16) \quad \begin{aligned} \langle A(C) \rangle &\approx \frac{1}{Z_d} \sum_{i=1}^N e^{-\frac{\varepsilon(C_i)}{k_B T}} A(C_i) \\ &\approx \sum_{i=1}^N \tilde{p}(C_i) A(C_i) \end{aligned} ,$$

replacing (2.14) by the discrete partition function

$$(2.17) \quad Z_d = \sum_{i=1}^N e^{-\frac{\varepsilon(C_i)}{k_B T}} .$$

Although simple sampling is easy to implement it has an important disadvantage. The property A is weighted statistically with the probability

$$(2.18) \quad \tilde{p}(C_i) = \frac{1}{Z_d} e^{-\frac{\varepsilon(C_i)}{k_B T}} .$$

When calculating the expected value $\langle A(C) \rangle$. But since the configurations C_i are generated randomly and with equal probabilities most of them will make a negligible contribution to the sum in (2.16). Therefore in order to simulate efficiently and with sensible resources it is vital to use the idea of importance sampling.

2.2.2 Importance Sampling

The method gets its name from emphasizing the statistically more "important" values by sampling them more frequently. Instead of moving to equally probable configurations C_i in the parameter space they shall be chosen according to a probability distribution $P(C_i)$. Then to take this into account (2.16) would have to be modified to

$$(2.19) \quad \langle A(C) \rangle \approx \frac{1}{Z'_d} \sum_{i=1}^N e^{-\frac{\varepsilon(C_i)}{k_B T}} P(C_i) A(C_i)$$

with

$$(2.20) \quad Z'_d = \sum_{i=1}^N \frac{e^{-\frac{\varepsilon(C_i)}{k_B T}}}{P(C_i)}$$

A good solution is to sample configurations with a probability given by the weight $P(C_i) \propto e^{-\frac{\varepsilon(C_i)}{k_B T}}$. Then the expression (2.19) for the expectation value would reduce to

$$(2.21) \quad \langle A(C) \rangle \approx \frac{1}{N} \sum_{i=1}^N A(C_i)$$

So the question is how to generate configurations with a distribution $P(C_i)$ that suits our needs [31, 32].

2.2.3 Metropolis Algorithm

In 1953, Nicholas Metropolis and coworkers proposed a new sampling procedure [33]. It is a realization of importance sampling and therefore allows to easily calculate the expected value of a property using (2.21). This modified Monte Carlo method is known as a Metropolis Monte Carlo simulation.

Rather than generating statistically independent configurations at random in this procedure new configurations are derived from existing ones. Configurations obtained by this fictitious dynamics constitute a so-called Markov chain of configurations.

To describe this dynamics we need to introduce considerations about transition probabilities. Let $P_A(t)$ be the probability of being in configuration A at time t. Let $W_{A \rightarrow B}$ be the probability per unit time, or transition probability, of going from A to B. Then

$$(2.22) \quad P_A(t+1) = P_A(t) + \sum_B [P_B(t) W_{B \rightarrow A} - P_A(t) W_{A \rightarrow B}]$$

Here t is marking the number of iterations of the procedure, not real time - our statistical system is considered to be in equilibrium, and thus time invariant.

A sufficient (but not necessary) condition for a equilibrium, i.e. time independent, probability distribution is the so-called detailed balance condition

$$(2.23) \quad P_A(t) W_{A \rightarrow B} = P_B(t) W_{B \rightarrow A}$$

In thermal equilibrium using the Boltzmann distribution this yields

$$(2.24) \quad \frac{W_{A \rightarrow B}}{W_{B \rightarrow A}} = \frac{P_B(t)}{P_A(t)} = \frac{e^{-\frac{\epsilon(B)}{k_B T}}}{e^{-\frac{\epsilon(A)}{k_B T}}} = e^{-\frac{\Delta \epsilon}{k_B T}}$$

There are many possible choices of the transition probabilities W which will satisfy detailed balance. A very simple one is:

$$(2.25) \quad W_{A \rightarrow B} = \begin{cases} e^{-\frac{\Delta \epsilon}{k_B T}} & \text{if } \Delta \epsilon > 0 \\ 1 & \text{if } \Delta \epsilon \leq 0 \end{cases}$$

So, if $\epsilon(B) > \epsilon(A)$

$$(2.26) \quad \frac{W_{A \rightarrow B}}{W_{B \rightarrow A}} = \frac{e^{-\frac{\varepsilon(B) - \varepsilon(A)}{k_B T}}}{1} = e^{-\frac{\Delta\varepsilon}{k_B T}}$$

and if $\varepsilon(B) \leq \varepsilon(A)$

$$(2.27) \quad \frac{W_{A \rightarrow B}}{W_{B \rightarrow A}} = \frac{1}{e^{-\frac{\varepsilon(A) - \varepsilon(B)}{k_B T}}} = e^{-\frac{\Delta\varepsilon}{k_B T}}$$

It was proven that the arbitrary probability distribution generated with this dynamic method fulfills the condition that allows using (2.21) to compute the expected value for an observable.

To describe a Metropolis Monte Carlo simulation, it is first necessary to define some notation. At any given time the current state of the system is denoted $State_0$. It has an energy of ε_0 and a property value of A_0 . After a random move it will be in $State_1$ with energy of ε_1 and a property value of A_1 .

At the beginning of a Monte Carlo simulation two parameters are set to zero: N (the number of Monte Carlo attempts) and A_{sum} (the sum of the property values). The system is currently in some $State_0$ and a temperature T , and the simulation proceeds as follows.

Step 1: Randomly change the system so that it is now in $State_1$, making sure that the configurations of both $State_0$ and $State_1$ are saved.

Step 2: Increment N by one. To determine what else happens you need to compare the energies of the two states and you need to generate a random number ξ which has a value between 0.0 and 1.0. The possible choices are given in the table below.

$\varepsilon_1 < \varepsilon_0$	$\varepsilon_1 > \varepsilon_0$	
	$e^{\frac{-(\varepsilon_1 - \varepsilon_0)}{k_B T}} > \xi$	$e^{\frac{-(\varepsilon_1 - \varepsilon_0)}{k_B T}} < \xi$
The energy of State ₁ is less than or equal to State ₀ . This means that the new state is accepted and becomes the new State ₀ . In addition, the property of the new state is added to the sum ($A_{\text{sum}} = A_{\text{sum}} + A_1$).	The energy of State ₁ is greater than State ₀ , but the energy difference is small enough that it is probabilistically accepted. This means that the new state is accepted and becomes the new State ₀ . In addition, the property of the new state is added to the sum ($A_{\text{sum}} = A_{\text{sum}} + A_1$).	The energy of State ₁ is greater than State ₀ , and the energy difference is large enough that it is probabilistically rejected. This means that the system stays in State ₀ . In addition, the property of the old state is (again) added to the sum ($A_{\text{sum}} = A_{\text{sum}} + A_0$).

Step 3. To continue the simulation, simply return to Step 1.

It is important to emphasize that every Monte Carlo attempt increases the value of N by 1 and adds a value to A_{sum} (either A_1 if the step is accepted or A_0 if it is rejected).

There are two factors that control how well the simulation samples the available search space of the system; the size of each Monte Carlo step and N the number of Monte Carlo steps attempted. If the size of a given Monte Carlo step is too large, the change in the energy between State₀ and State₁ may become very large and virtually all of the attempts will be probabilistically rejected. This causes the simulation to get "stuck" at a particular point in search space. Conversely, if the Monte Carlo step size is too small, the system may have a hard time sampling all of the available search space. In either case, the total number of attempts may have to become exceedingly large.

It is important to emphasize that proper statistics can only be obtained from a Metropolis or simple Monte Carlo simulation if the procedure allows that every configuration can be reached from every other configuration in a finite number of

iterations. If regions of search space are separated by an infinitely high barrier that is wider than the maximum allowed step size, it will be impossible to travel from one region to the other, and the resulting average of any property will be incorrect. A simulation that allows for the sampling of all finite regions of search space is called ergodic.

2.2.4 Monte Carlo Steps

The task is to effectively sample all of the states of the system that are accessible at the given temperature. To accomplish this a simulation step consisting of two transformations is used. These are a pivot and a rotation move (Figure 2.4 and Figure 2.5). As has been pointed out finding a good average step size at a particular temperature depends upon the system being modelled. In practice, the number of accepted Monte Carlo steps can be used as a guide. The intervals for the rotation angles δ and ϕ were chosen to yield acceptance rates of 30% and 50% respectively [4].



Figure 2.4: The pivot move: One segment is selected randomly as the origin of a rotation. of a random angle of the interval $[-\delta, \delta]$ around a random axis.

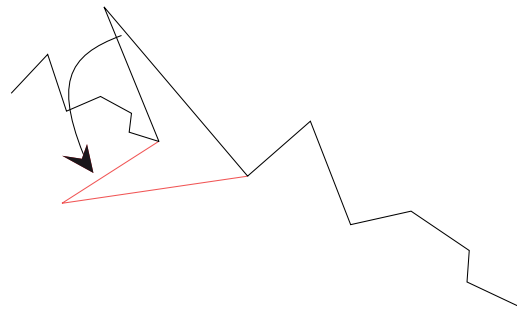


Figure 2.5: For a rotation move, a segment is chosen randomly. The end point of this segment is rotated around an axis through the start point of this segment and the end point of the next segment by an angle chosen randomly from the interval $[-\phi, \phi]$

2.3 Tools for the evaluations

The Monte Carlo simulation software is based on an elaborate object oriented design and implemented in C++ [34]. The implementation of the model described in ref. [4] has since then been improved by optimization of the code and incorporation of new C++ standards.

Additional software was created to systematically produce and analyze ensembles of fiber conformations. In the beginning this was done by using shell scripts which included the batch editing tools sed and awk [35]. These solutions proved to be not suited for the increasing complexity. Thus another scripting language was needed that would serve future developments in a better way. It was decided to use a contemporary programming language called Ruby. It combines syntax inspired by Perl with object-oriented features and is a single-pass interpreted language (i. e. it is not compiled before executing). Its official implementation is free [36]. Ruby supports running parallel tasks by splitting them into multiple threads within the program. This was implemented to use a 7 dual-core processor cluster more efficiently.

Starting simulations

Starting a simulation consists of calling the programs ‘chromCreat’, ‘createDNATable’ and ‘backebackekuchen’ in this order. ‘chromCreat’ takes the parameters for the fiber geometry and produces a starting configuration. Based on this configuration ‘createDNATable’ writes a table with interpolated values for the DNA electrostatic interactions (used to save calculation time). ‘backebackekuchen’ is the implementation of the Metropolis Monte Carlo Algorithm described in section 2.2.

The program ‘runsim’ consists of two parts. In the first part a stack is filled with all the combinations of start parameters needed for the particular set of simulations. The second part starts the simulation processes on the remote clients.

When a client finishes a simulation it gets assigned the next process in line. The source code for this is included in the appendix (heading ‘runsim.rb’).

Analyzing the simulation outcomes

All information the simulations yield are contained in so called trajectory-files. To process all this files a program with an object-oriented design was created. It consists of the four objects illustrated in Figure 2.6.

Evaluation objects controls the program and holds the information concerning the way the evaluation should be conducted, most importantly the location of the trajectory files. A result object ‘reads’ the analysis outcomes via an Analyzer object which is the interface to the information contained in the trajectory files. It also stores the data, and sorts and prepares it for output. This output can be in terms of diagrams via a plotter object or in the form of tabulated text files.

Thus a typical evaluation looks like this.

```
require 'evaluation'
require 'analyzer'
require 'result'
require 'plotter'

eval = Evaluation::new
eval.name = "other"
eval.traj_names =Dir['a*_l*.res']
eval.allfilesinonedir = true
eval.run_analysis_execs = %w(energies compEffAlpha)
eval.start
```

The first block loads the files containing the source code for the four objects and the second block starts the evaluation after setting all the attributes that control the way how.

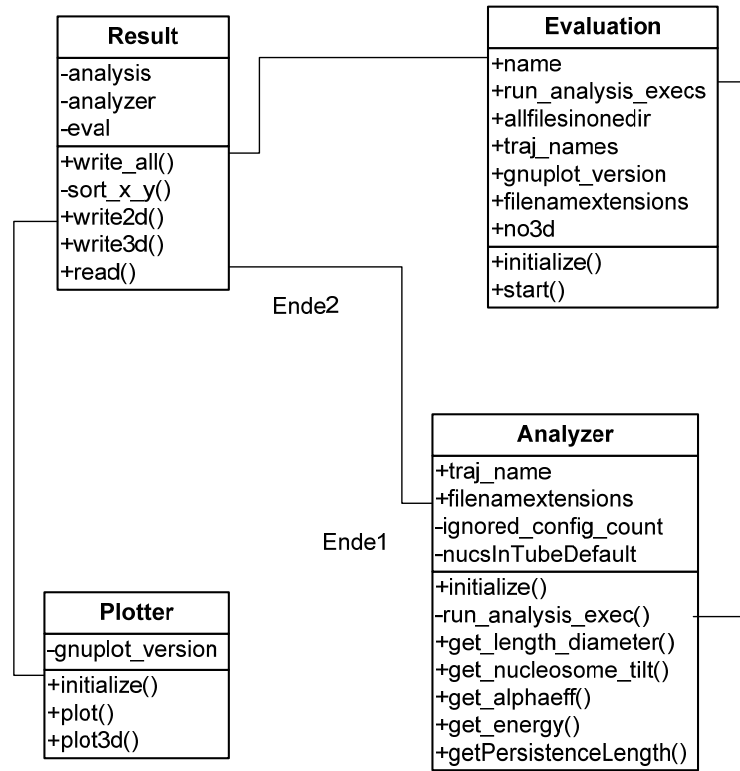


Figure 2.6: UML chart showing the attributes, methods and relationships of the four classes created to evaluate the data read from the trajectory files. The source code for these objects can be found in the appendix

For getting an expectation value the mean of results obtained from a trajectory-file has to be taken according to (2.21). This is done by the “get_”-methods of the Analyzer class. To calculate the deviations for the $N=400$ observations the formula for running sums

$$(2.28) \quad s = \sqrt{\frac{N \sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i \right)^2}{N(N-1)}}$$

is used.

3 Results

The pictures in Figure 3.1 show some chromatin fibers that are the result of the simulations carried out for this thesis. These are visualizations of a set of coordinates which are produced by the simulation software. In fact Figure 3.1 does not resemble four distinct fibers but different Monte Carlo steps of a simulation and hence different conformations of the same fiber.

As described in detail in the methods section the simulation is based on a model for the chromatin fiber which uses only two building blocks: Nucleosomes and the connecting DNA. Only by approximating the interactions of single atoms by effective potentials the number of interacting components for a fiber becomes computable. Therefore the connecting DNA is reduced to linear segments with a cylindrical Debye-Hückel-potential while the nucleosomes are described by ellipsoid Gay-Berne-potentials. Figure 3.2 shows a fiber section with three nucleosomes. It illustrates the three parameters that characterize how consecutive nucleosomes are connected with each other in this particular model. Later these parameters are varied in order to observe their influence on the whole of the fiber.

Furthermore we wanted to observe the influence of linker histone H1 on the fiber. To distinguish between the cases with and without H1 being present two different local geometries for the nucleosomes have been designed as illustrated in Figure 3.3 (details regarding the fiber model see chapter 1.2). The visualization tool uses plain cylinders to display the nucleosome cores. The stem is not drawn, which may sometimes be confusing when examining the pictures.

Figure 3.1 A shows a typical start conformation for a simulation. To create it a separate program called chromCreate is used which generates a fiber with regular geometry. The start conformations reflect the “ground state” values of the simulation parameters (i.e. entry-exit angle, torsion-angle and linker DNA length, temperature, parameters for the potentials, etc.).

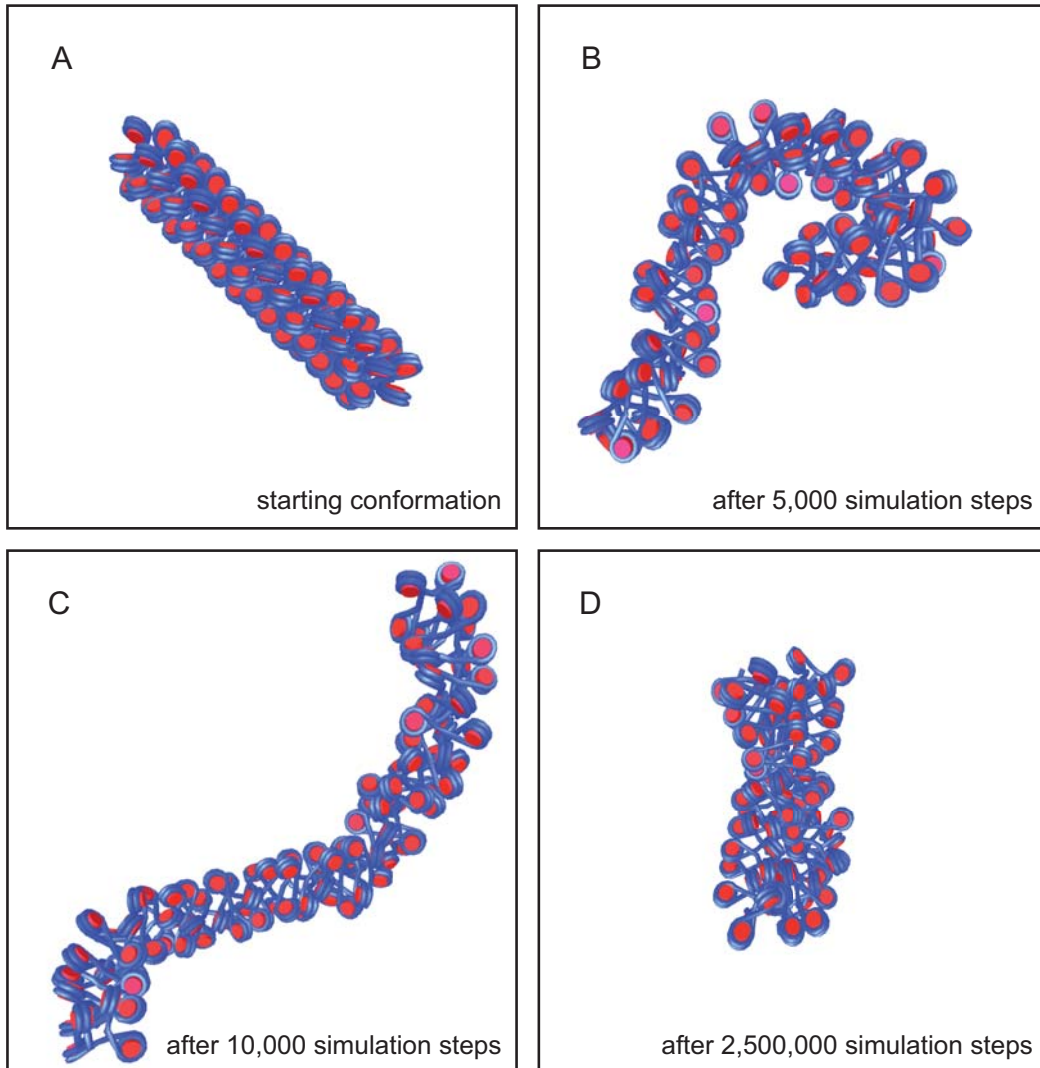


Figure 3.1. Monte Carlo Simulation of a chromatin fiber (A) Starting conformation with very regular arrangement of nucleosomes . (B)-(D) Changes in conformation of the same fiber as it is transformed by the indicated simulation steps. (D) Demonstrates a representative equilibrium conformation.

Figures 3.1 B,C,D demonstrate how the conformation of the fiber changes with progressing simulation steps. A simulation step is a geometrical transformation of the fiber which generates new conformations. After each simulation step the total of these energies is calculated and the new conformation then is either rejected or accepted according to the rules of the metropolis algorithm (see section 2.2) which at the end delivers an ensemble that represents the equilibrium distribution.

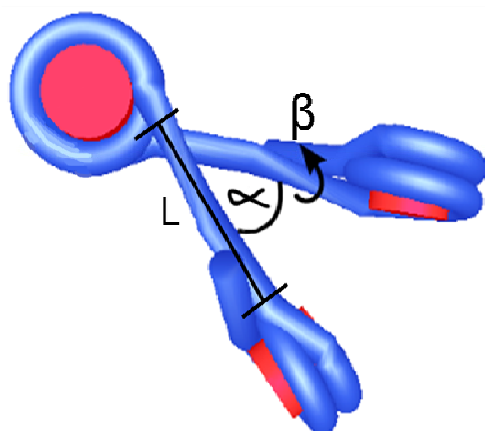


Figure 3.2. The geometric parameters that define the fiber structure: The red cylinders describe the nucleosome core with , 5.5 nm height and 11nm diameter. The blue tubes are the DNA and have a diameter of 2 nm. The DNA entry-exit angle α , the length of the linker DNA, L and the angle β that describes the torsion between two adjacent nucleosomes.

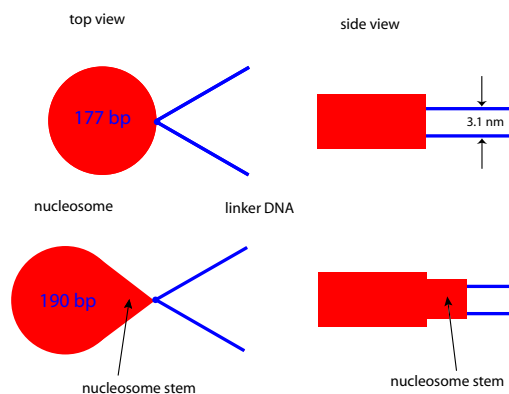


Figure 3.3. The two types of coupling of DNA to nucleosomes are shown. A nucleosome without stem corresponds to 177 bp (2 full turns of DNA) while adding the stem increases the attached base pairs to 190. According to this definition the repeat length is the linker DNA length divided by 0.34 nm plus the number of base pairs contained in the chosen nucleosome coupling type.

Therefore during the ongoing simulations conformations with either lower or similar energies are selected preferably. This can be seen when looking at the change from Figure 3.1 A to Figure 3.1 B. Because of the repelling forces between the constituents of the fiber they are obviously further apart from each other such that the fiber in Figure 3.1 B has a lower energy than the starting conformation in Figure 3.1 A. Obviously there are still enough attractive forces to hold the structure of a fiber. Although in Figure 3.1 D it seems like the order has collapsed and we only have a chunk of unarranged matter a more detailed examination reveals that it is as a U-shaped fiber which in addition is twisted along the symmetry axis of the “U”.

Figure 3.4 shows a typical progression of the interaction energies for the Monte Carlo simulations. It can be seen that the process starts with high total energies for

the starting conformation and then very quickly descends (note the logarithmic scale) for the following conformations. All evaluations were made in a way that this equilibrium state was reached by running the simulations until the energies remained constant.

When talking of a trajectory a set of fiber conformations is meant that were picked out in regular intervals during the ongoing simulations. The interval for picking out conformations was chosen such that they could be considered uncorrelated.

All the results presented here were obtained from fibers consisting of 100 nucleosomes. Simulations of 2,500,000 Monte Carlo steps each were done and conformations were saved every 5000 steps. That makes 501 ‘snapshots’ in a trajectory (including the starting conformation). For evaluation the first 101 (= 101 x 5,000 steps = 505,000 steps) of these were ignored so we could be sure to have reached the equilibrium state. This is evident from Figure 3.4 where after 100,000 monte carlo steps, which corresponds to the first 20 conformations of the associated trajectory, the equilibrium state is clearly reached.

3.1 Initial observation and motivation

This work was initiated by simulations made by Dietrich Foethke. His analysis of chromatin fibers indicated the existence of a local maximum of the mass density for a specific combination of entry-exit angle and linker DNA length (see Figure 3.5 A). Because the mass density is considered an indication for the accessibility of chromatin regions the suggested relation to the geometric parameters of the model seemed worth further investigation.

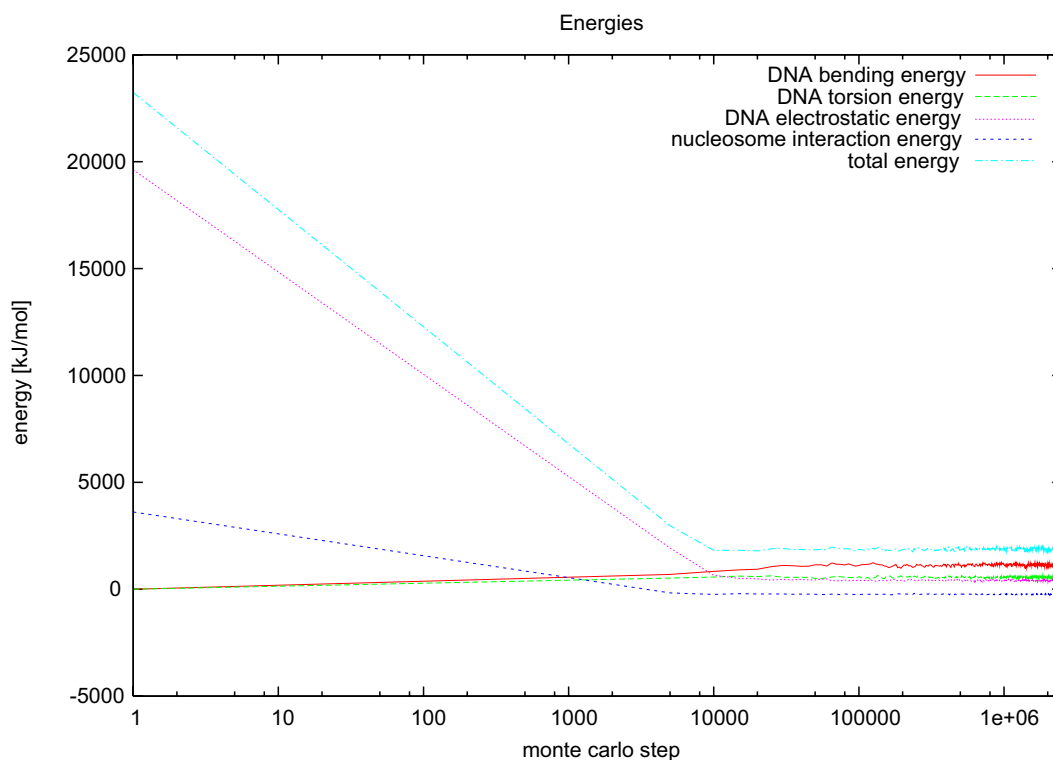


Figure 3.4. The contributing energies reach a steady state quickly. This graph shows the typical behavior of the interaction energies in the course of a simulation. The simulation steps are plotted on a logarithmic scale. After approximately 10,000 simulation steps, which is small compared to the total of 2.5 million steps, the total energies reach a steady state from which the energies of the following conformations don't deviate much. As can be seen in this case the component that is mainly responsible for the decay is the repulsive electrostatic force between the negatively charged parts of DNA. Also it can be seen that regarding bending and torsion of the DNA the starting conformation is very relaxed, although depending on the input parameters that are chosen to construct the starting conformation the distribution of these energies may vary.

In order to achieve a more detailed and quantitative description of the phenomenon we next intended to study the effect of the variations of the entry-exit angle α and the linker DNA length L had on other parameters. Therefore procedures that handle this task were integrated into the analysis programs. The results can be seen in Figure 3.5 (B-F)

For reasons of reproducibility as well as to confirm the former analysis new computer programs were created based on the methods used earlier. These analysis tools were then applied to the existing and new data sets.

To characterize the fiber conformation we chose six parameters and arranged them in panels corresponding to the ones in Figure 3.5. This style of display is kept throughout all the following diagrams for reason of consistency and clarity.

The parameters are:

- A. Linear mass density. This is the amount of nucleosomes per 1 nm contour length of the fiber.
- B. The total energy, which is the sum of the interaction, bending and torsion energies described earlier in connection with Figure 3.4.
- C. The average fiber diameter.
- D. Persistence length: Defined as the length over which correlations between the tangent vectors of the fiber segments is reduced by the factor $1/e$. It quantifies the stiffness of a polymer.
- E. Effective entry-exit angle. While undergoing the electrostatic and entropic forces the entry-exit angle changes during a simulation. The measured α_{eff} reflects the true entry-exit angle, while α is the input parameter.
- F. Average nucleosome tilt angle. The tilt angle is the angle between the axis of a nucleosome and the fiber axis.

It should be noted that one point in the diagrams is not the result of one singular fiber but represent an ensemble of 400 fibers with similar energies. For the persistence length, the mass density, and the total energy the final values are obtained straight forward by first computing the respective value for one fiber and then calculating the mean of the 400. As the diameter, the effective entry-exit angle and the nucleosome tilt are not the same along a fiber, these parameters are calculated by first taking the average over each fiber. So the final result obtained for a trajectory is actually an average of an average.

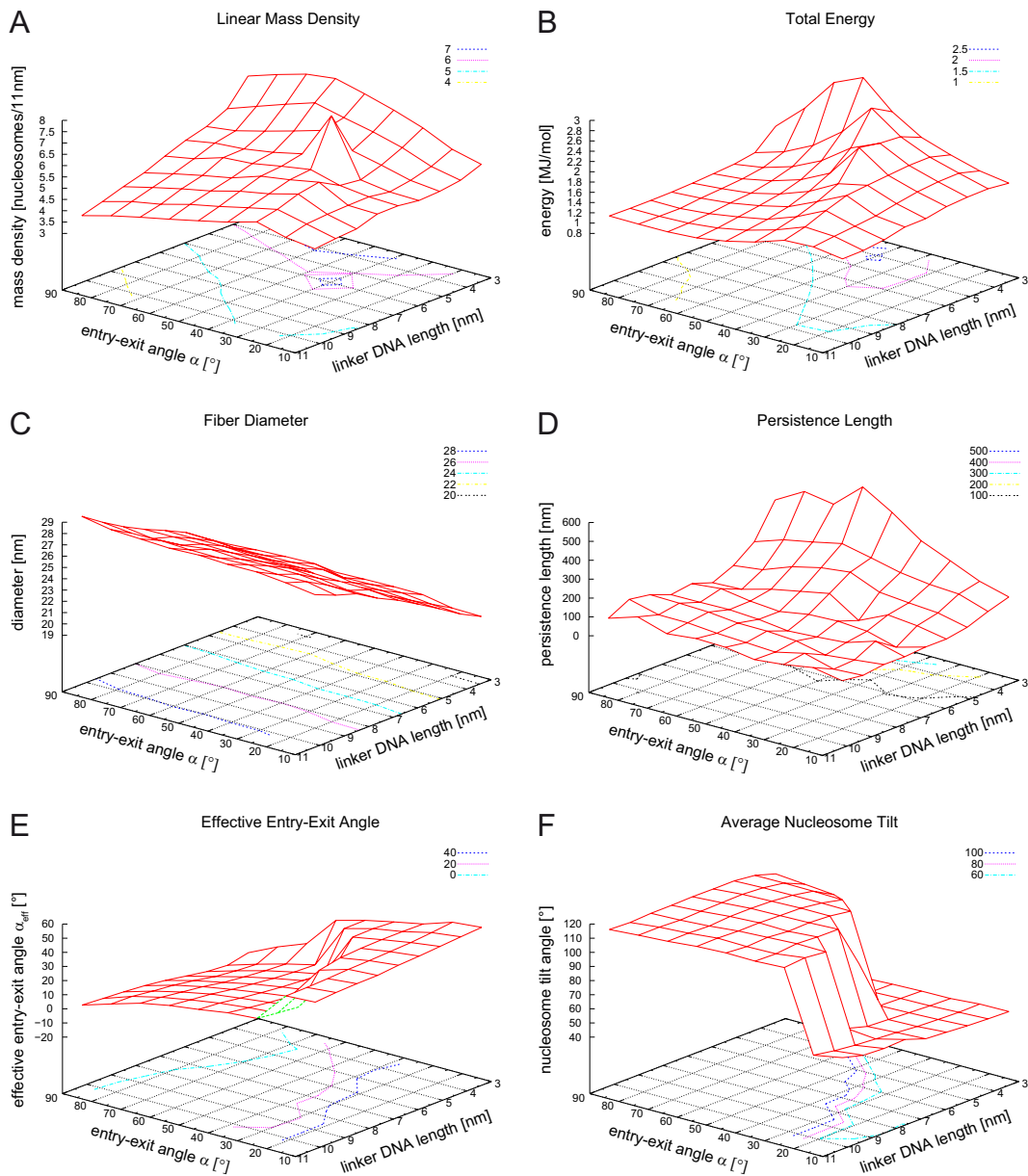


Figure 3.5. Systematic analysis of several parameters for miscellaneous combinations of entry-exit angle and linker DNA length. Fibers were simulated in the range from $\alpha = 20^\circ$ to $\alpha = 90^\circ$ in steps of 10° and $L = 3.4$ nm to $L = 10.2$ nm in steps of 0.85 nm, $\beta = 110^\circ$ (A) The initial observation (D. Foethke, G. Wedemann and K. Rippe, unpublished). For a DNA linker length of 6 nm and an entry-exit angle of 40° a steep increase of the chromatin fiber mass density occurs. (B) Total energy. (C) The diameter of the fiber grows proportional with the linker DNA length and shows no dependency on the effect seen in the other parameters. (D) Persistence length. (E) Effective entry-exit angle. (F) Average nucleosome tilt. Except for the fiber diameter the parameters retrieved from the same data indicate that the transition might not only occur at one specific combination but at contiguous combinations of entry-exit angle and DNA linker length.

3.2 Simulations with higher resolution

To confirm the significance of the peak in the mass density new simulations with higher resolution in the particular region were conducted.

The elastic and interaction parameters as well as the temperature from the initial observations were kept constant. Also for the geometric parameter β the formerly used value was used. This was defined due to tests [5] that showed that for $\beta=110^\circ$ the mass density of the fiber is in agreement with the experimental value of 6 nucleosomes/11 nm [37]

Scripts in the Ruby programming language were written for running multithreaded simulations on a cluster with 7 dual-core processor.

In previous simulations the entry-exit angle α was scanned between 20° and 90° in steps of 10 degrees and for each α the linker DNA lengths of $L=3.4$ nm to $L=10.2$ nm in steps of 0.85 nm were simulated (which makes 81 trajectories). For the higher resolution sampling the step sizes were reduced to 2.5° for α and 0.85 nm for L . Also the range for α was narrowed to the interval of 20° to 50° (resulting in 369 trajectories).

After analyzing and plotting the results (see Figure 3.6) the one peak from the initial observation transformed into multiple peaks which together can be identified as an edge. The other parameters support this observation by showing characteristic transitions for the same combinations of input parameters α and L .

Thus the existence of the transition was confirmed in the analysis. To see how steep the transition effect is a series of chromatin fibers that ‘slice’ through one of the major peaks in a once higher (16 times) resolution was simulated (see Figure 3.7).

To illustrate the sudden change in the mass density some fibers from the trajectories near the region of major effect are visualized (Figure 3.8) and their measured properties are displayed in Table 3.1. It is clear that the simple visual

inspection of the fiber geometry can provide only a first impression of the difference.

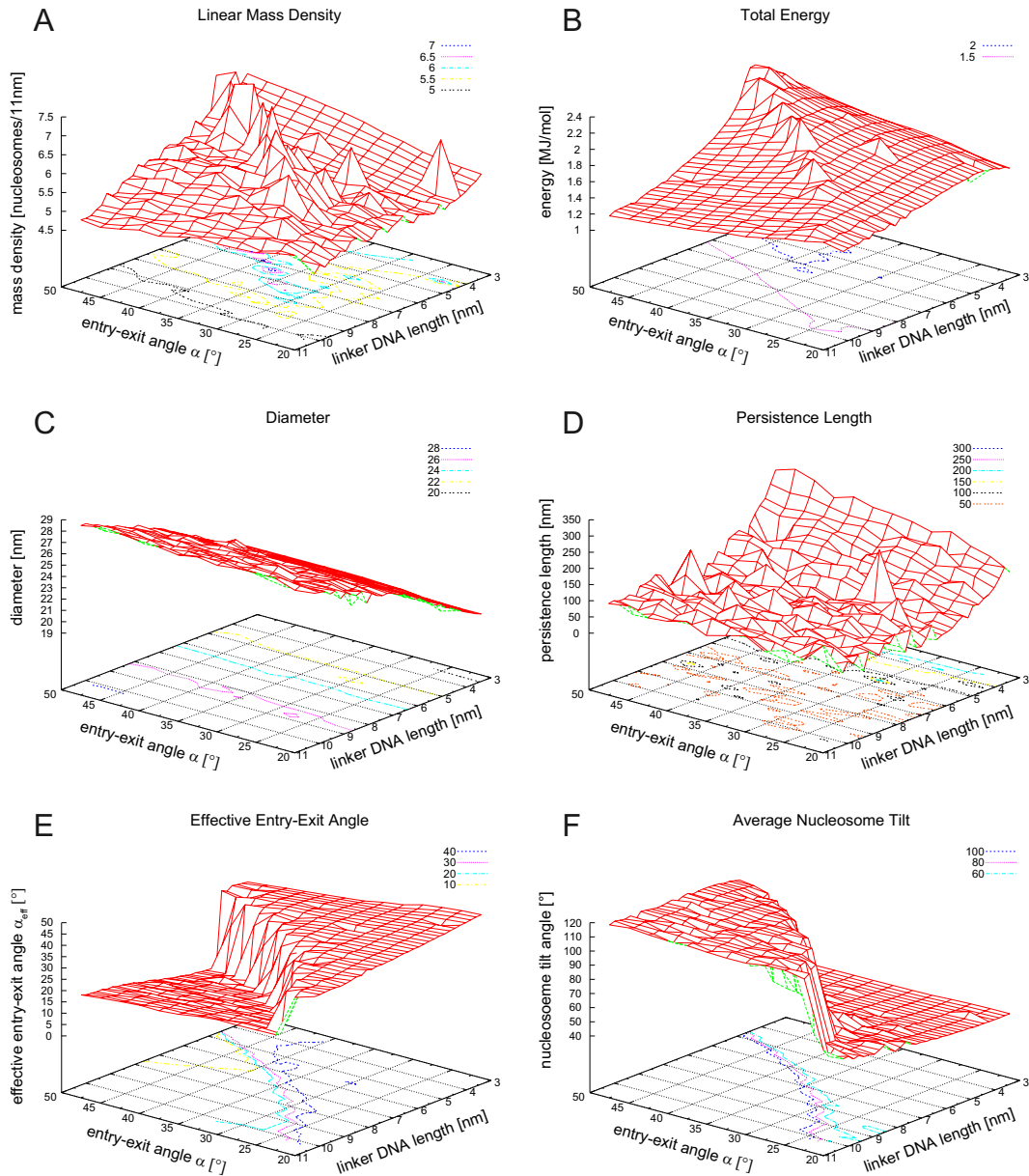


Figure 3.6. Output parameters of trajectories obtained from simulations in the region sampled from $\alpha=20^\circ$ to $\alpha=50^\circ$ in steps of 2.5° and from $L=3.4$ nm to $L=10.2$ nm in steps of 0.85 nm; $\beta = 110^\circ$. (A) Linear mass density. Multiple peaks in the chromatin fiber mass density constitute an edge rather than a peak. The observed scattering suggests that the extraordinary peak from the initial observation was one of these outliers, but still a transition effect is detected and confirmed by the behavior of the other parameters (B) Total energy. (C) Fiber diameter. (D) Persistence length. (E) Effective entry-exit angle. (F) Average nucleosome tilt.

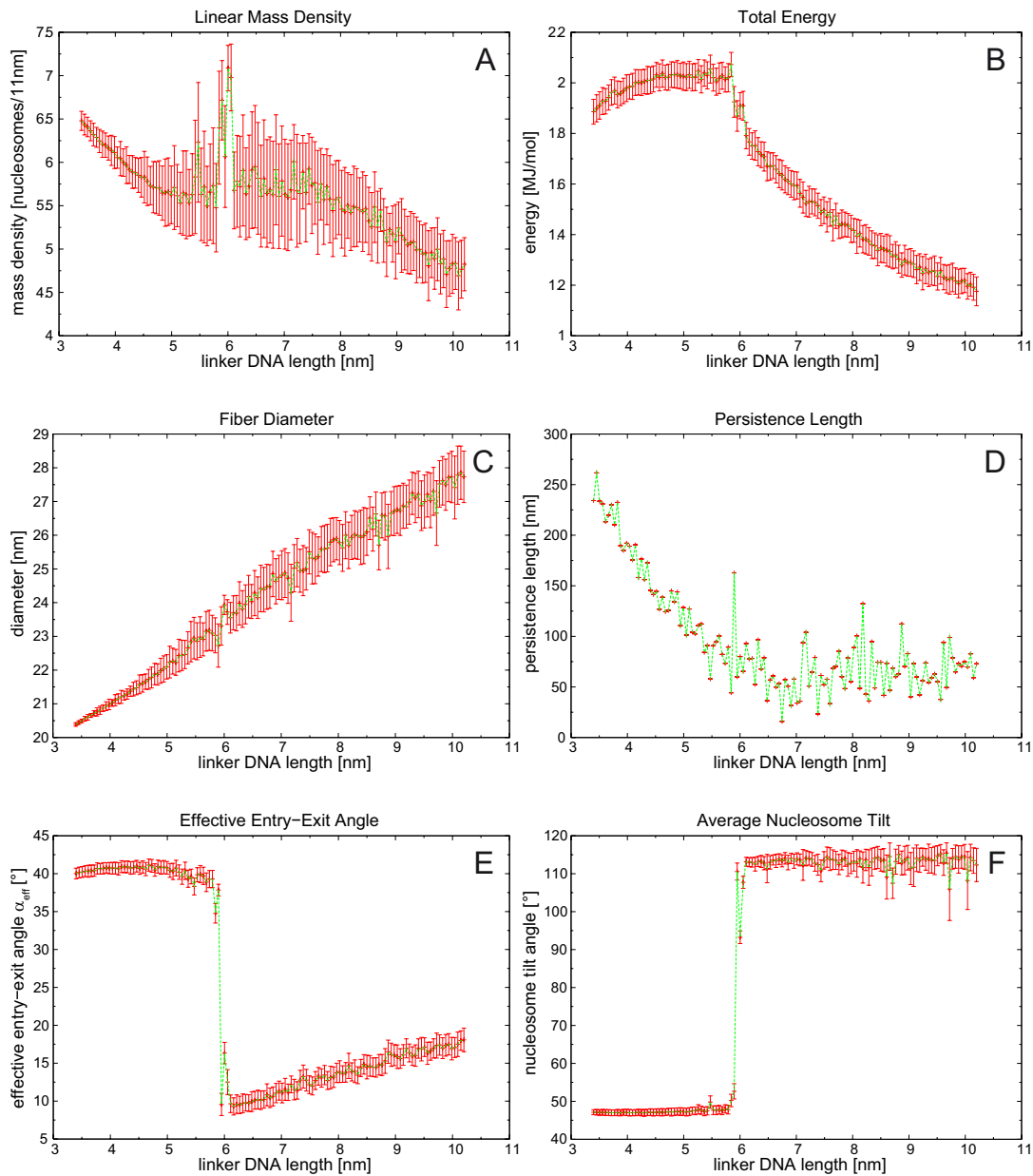


Figure 3.7. Simulations with starting parameters $\alpha=40^\circ$ and $\beta=110^\circ$. $L=3.4$ nm to $L=10.2$ nm in 129 steps of 0.05 nm. Plotted are the output parameters as functions of the linker DNA length L . (A) Linear mass density. For $L \sim 6$ nm a steep increase a mass density of ~ 7.5 nucleosomes/11 nm is seen. (B) Total energy. (C) Fiber diameter. The diameter grows linear to the linker DNA length. (D) Persistence length. (E) Effective entry-exit angle. α_{eff} is around the input angle $\alpha = 40^\circ$ for linker DNA lengths shorter then ~ 6 nm . There it decreases to approximately 10° . For bigger L it grows to $\sim 17^\circ$ for 10 nm linker DNA length (F) Average nucleosome tilt. It has a sharp increase from an average of $\sim 47.5^\circ$ to an average of 115° at ~ 6 nm

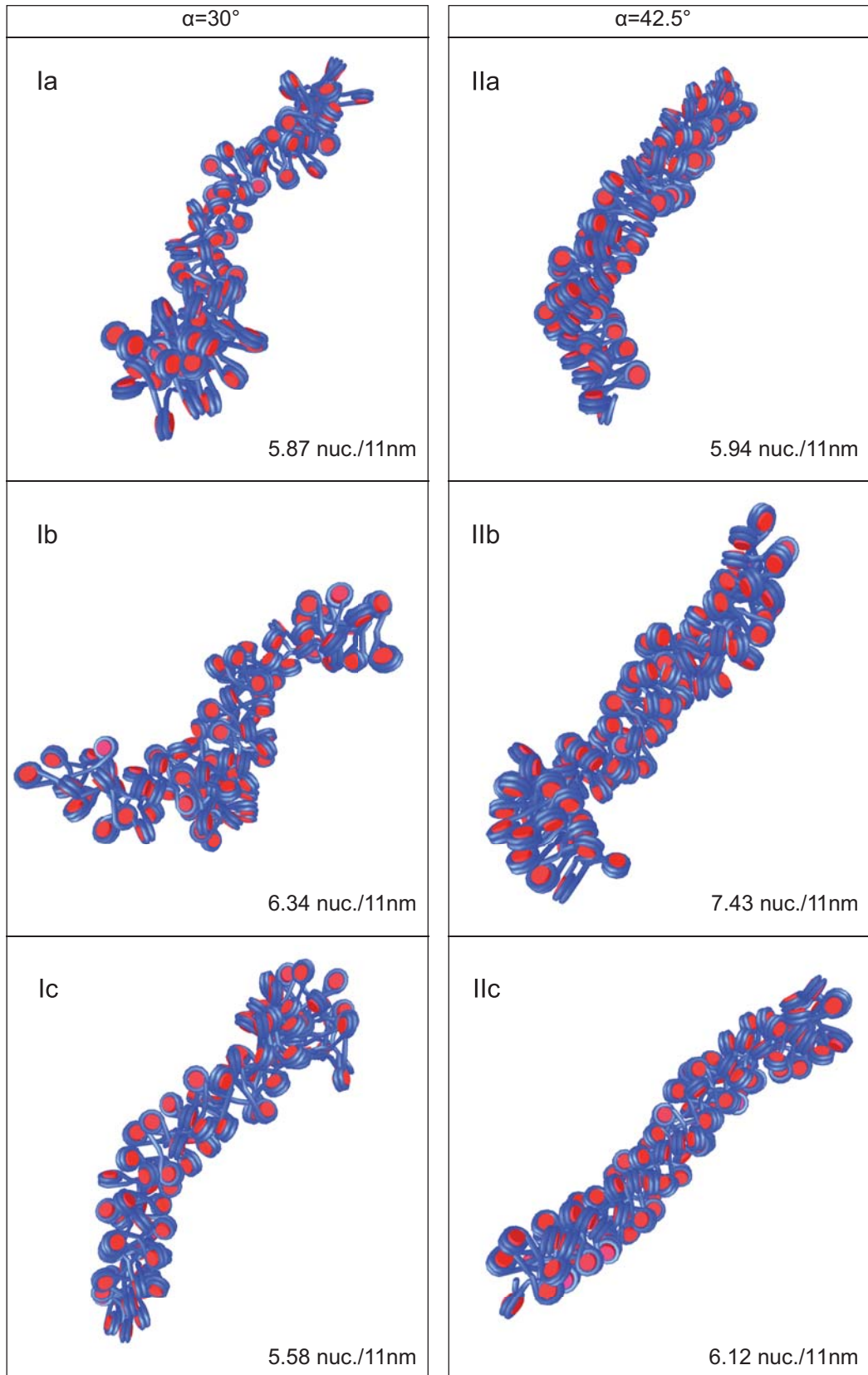


Figure 3.8 (opposite side). Examples from trajectories with parameters before, inside and after the linker DNA length interval where the transition occurs. The linear mass density of the corresponding trajectory is shown for each snapshot. It is difficult to detect the differences in the density, length or distances by pure sight. (Ia-Ic) linker DNA lengths: 7.9 nm \rightarrow 8.1 nm \rightarrow 8.3 nm; $\alpha = 30^\circ$ (IIa-IIc) linker DNA lengths: 5.3 nm \rightarrow 5.5 nm \rightarrow 5.7 nm; $\alpha = 42.5^\circ$

α [°]	30			42.5		
L [nm]	7.9	8.1	8.3	5.3	5.5	5.7
ϵ [MJ/mol]	1.7 ± 0.1	1.7 ± 0.1	1.6 ± 0.1	2.1 ± 0.1	2.2 ± 0.1	1.9 ± 0.1
λ [1/11nm]	5.9 ± 0.3	6.3 ± 0.3	5.6 ± 0.5	5.9 ± 0.6	7.4 ± 0.4	6.1 ± 0.5
δ [°]	58.8 ± 4.2	93.8 ± 3.7	113.4 ± 3.6	46.7 ± 1.1	60.4 ± 3.4	111.4 ± 1.9
α_{eff} [°]	38.7 ± 1.0	25.5 ± 1.2	16.2 ± 1.3	37.3 ± 1.3	32.9 ± 1.3	9.6 ± 1.9
\emptyset [nm]	25.3 ± 0.7	25.0 ± 0.7	26.0 ± 0.6	22.8 ± 0.5	22.6 ± 0.4	23.1 ± 0.5
l_p [nm]	66.1	60.2	62.3	74.8	36.4	68.1
l_c [nm]	185.4 ± 6.3	172.3 ± 5.4	191.3 ± 5.7	158.1 ± 3.8	146.1 ± 4.2	153.2 ± 4.5
N_{nucl}	98.8 ± 3.9	99.3 ± 2.2	97.0 ± 6.6	85.3 ± 8.0	98.5 ± 4.0	85.2 ± 6.0

Table 3.1. Output parameters of the trajectories from Figure 3.8. Abbreviations: α = entry-exit angle; L = linker DNA length; ϵ = total energy; λ = linear mass density; δ = average nucleosome tilt angle; α_{eff} = effective entry-exit angle; \emptyset = Fiber Diameter; l_p = persistence length; l_c = contour length of the fiber. N_{nucl} = number of nucleosomes used to calculate λ .

3.3 Comparing versions of the simulation software

To be able to analyze the complete parameter space the simulation software has been enhanced in collaboration with G. Wedemann, Rene Stehr and Nick Kepper by adding a method to heat up the system between simulation steps. This approach ensures that the complete conformation space is sampled as required for obtaining an equilibrium ensemble from the Monte Carlo simulation. Because this costs computing time the algorithm for computing interaction energies needed to be optimized further. To see whether results are still reproducible, and in particular to test if the new cube algorithm affected the results, simulations from different programs were compared (Figure 3.9 and Figure 3.10). MC2 refers to the newest version with the cube algorithm and MC0 to the version with which the

initial observation was made. Changes between MC0 and MC1 were optimizations of the code.

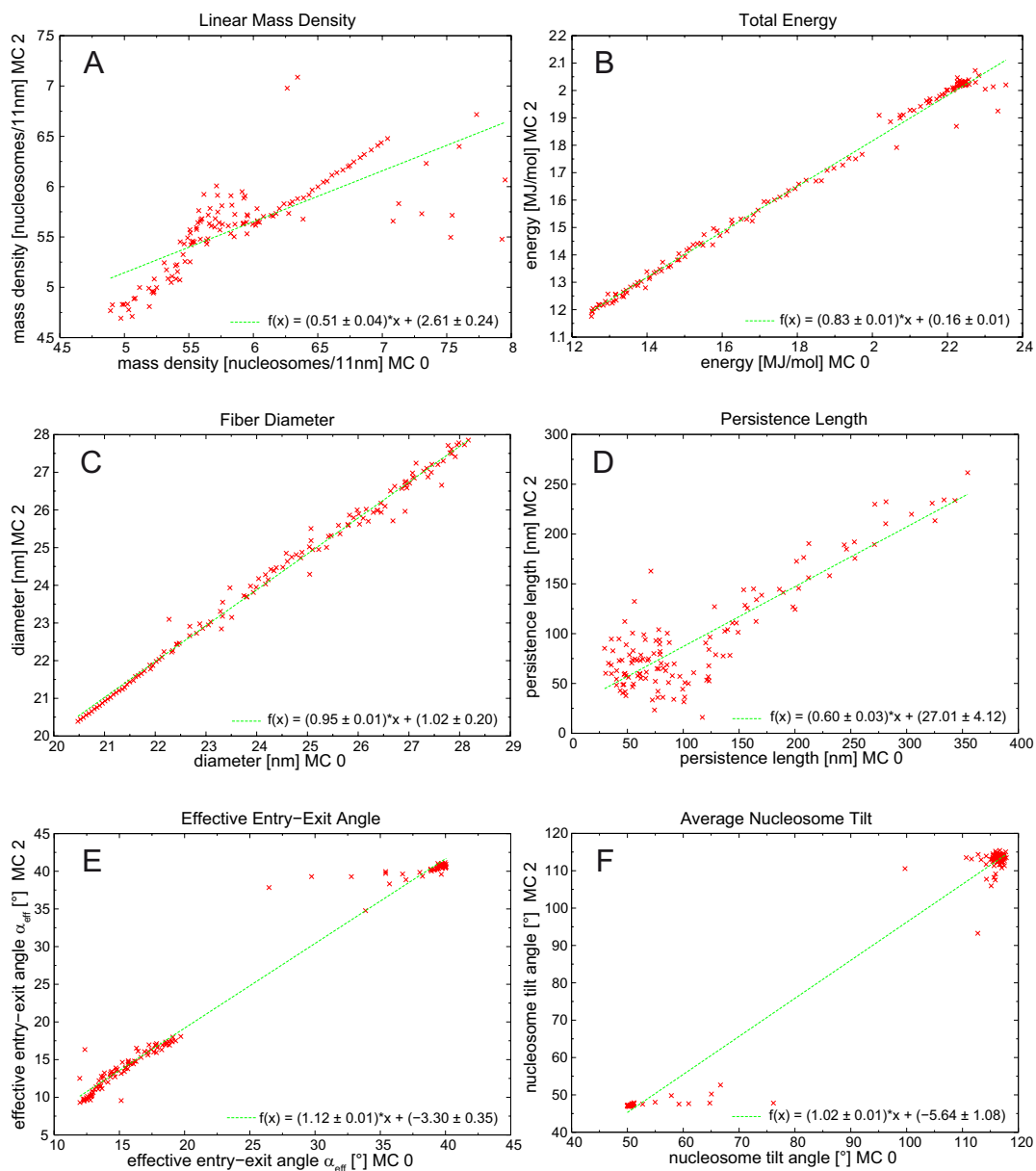


Figure 3.9. Comparison of data created with MC0 and MC2. Starting parameters for these simulations were an entry-exit angle of $\alpha=40^\circ$ and $\beta=110^\circ$. The length of the linker DNA was changed in steps of 0.05 nm between 3.4 nm and 10.2 nm. The corresponding parameters for same linker DNA lengths are plotted. (A) Linear mass density. (B) Total energy. (C) Fiber diameter. (D) Persistence length. (E) Effective entry-exit angle. (F) Average nucleosome tilt.

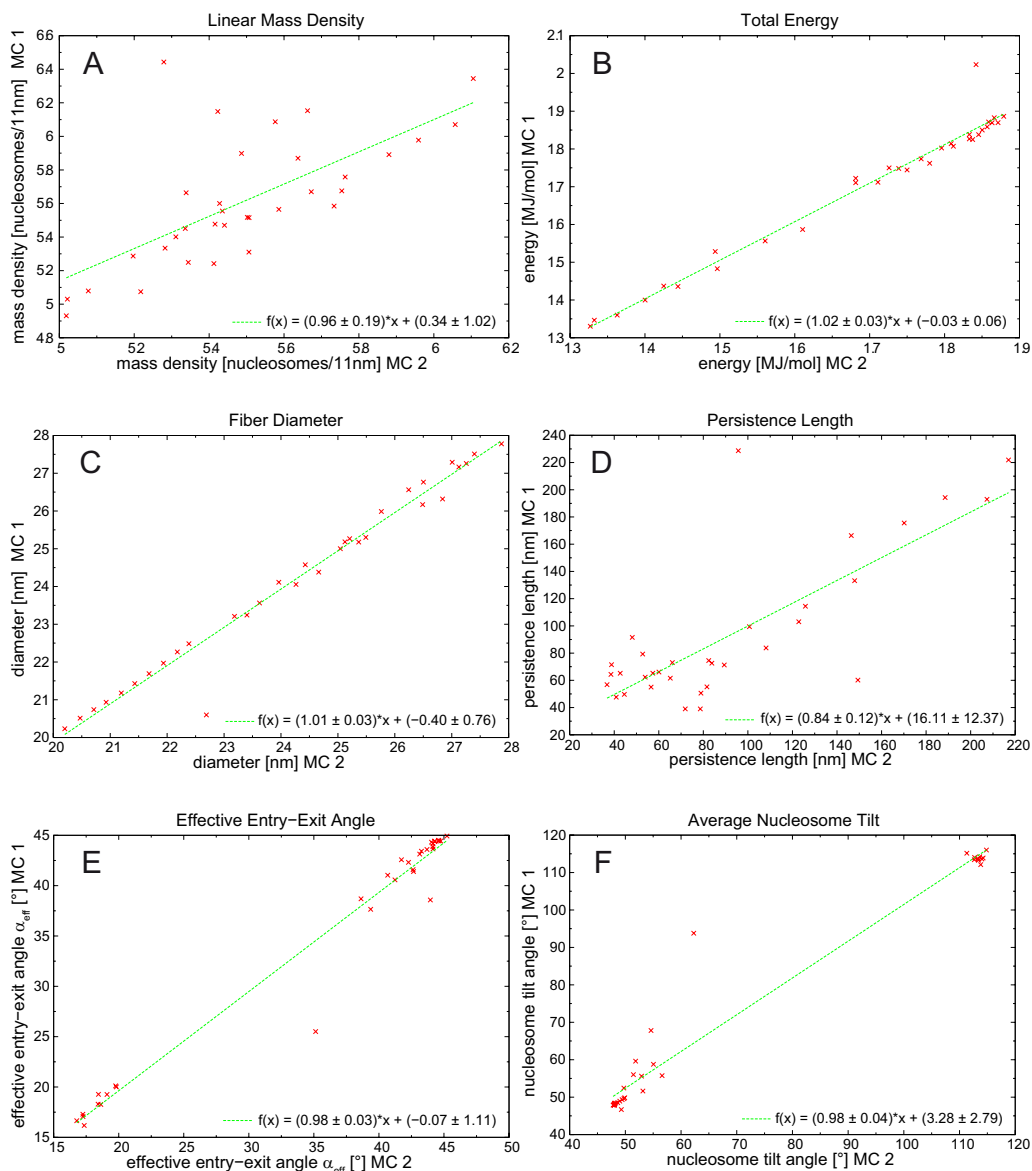


Figure 3.10. Comparison of data created with two consecutive versions of the simulation software (MC1 and MC2). Starting parameters for these simulations were an entry-exit angle of $\alpha=40^\circ$ and $\beta=110^\circ$. The length of the linker DNA was changed in steps of 0.85 nm between 3.4 nm and 10.2 nm. The corresponding parameters for same linker DNA lengths are plotted (A) Linear mass density. (B) Total energy. (C) Fiber diameter. (D) Persistence length. (E) Effective entry-exit angle. (F) Average nucleosome tilt.

3.4 Influence of the stem

In an additional set of simulations the effect of the stem motif, which substitutes for the linker Histone H1 including the attached linker DNA (see Figure 3.3 and Figure 2.3) was studied. We recognized that by only making a slight alteration in our simulations the tetranucleosome structure proposed by Schalch et. al. [23] (see Figure 3.15 B) could be incorporated. This structure has no associated H1 and its geometry corresponds to using the parameters $\alpha = 23^\circ$ and $L = 6\text{nm}$ and $\beta = -85^\circ$ in our chromatin model. Thus by changing the torsion angle β between consecutive nucleosomes additional information about a fiber conformation in the absence of H1 could be obtained.

We did simulations for the same combinations of parameters α and L as in section 3.2 but without a stem motif and with β changed to -85° . The results can be seen in the 3d-plots from Figure 3.11. When comparing this with Figure 3.6 it can be detected that the edge with the strong change in parameters is now missing but that except for the energies and the persistence length the overall tendencies seem to agree with each other. Therefore a more detailed look at a section with $\alpha = 40^\circ$ for $\beta = -85^\circ$ is plotted in Figure 3.12. In addition simulations for $\alpha = 40^\circ$ but with $\beta = 110^\circ$ were made too (Figure 3.13). This was done to rule out the possibility that the peak had only disappeared because of the changed torsion angle. The result of this analysis showed again no hint of a transition and no difference between $\beta = -85^\circ$ and $\beta = 110^\circ$ was apparent (Figure 3.14).

Figure 3.15 shows the qualitative and quantitative outcome for the trajectory with the parameters taken from the tetranucleosome geometry.

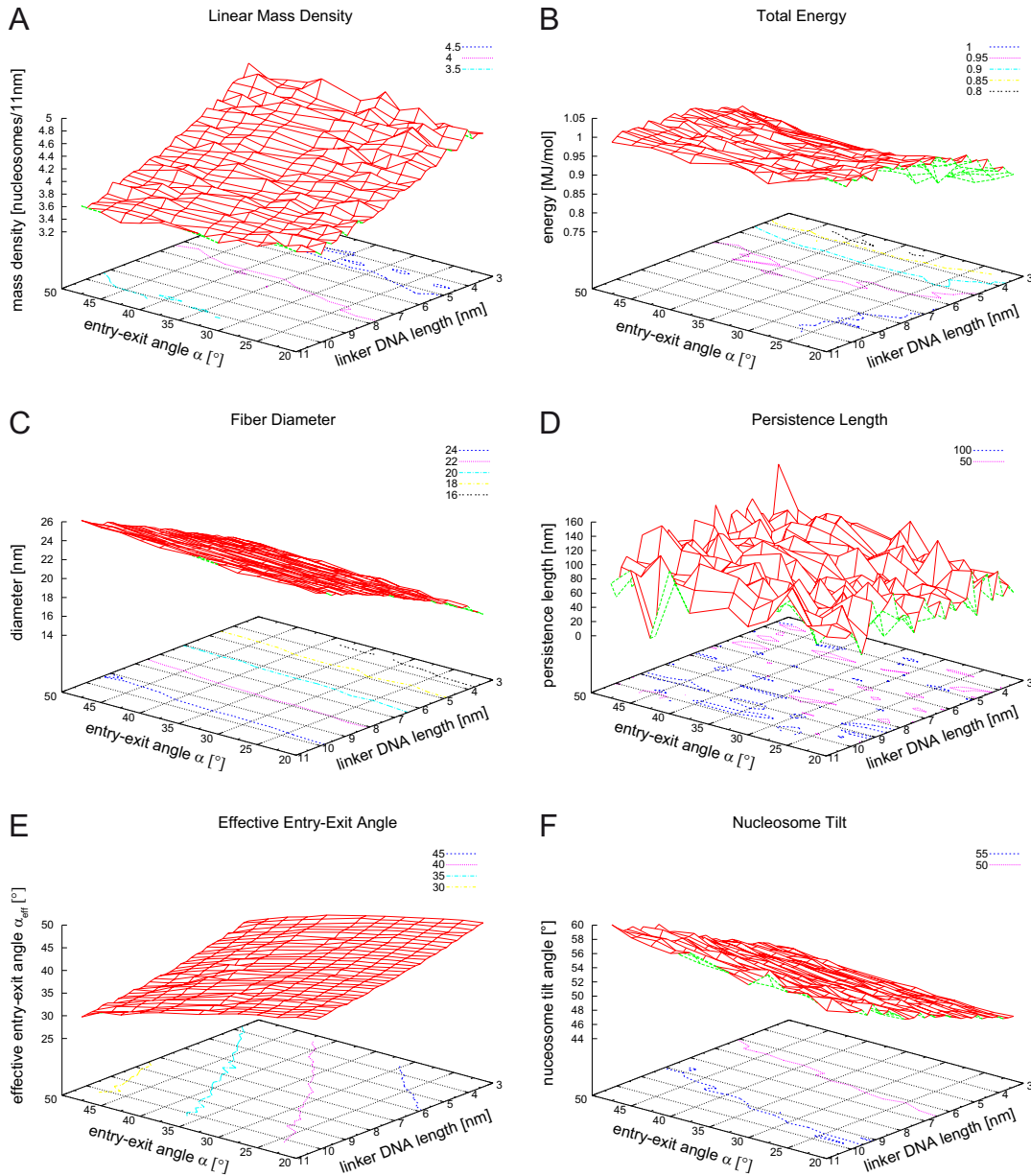


Figure 3.11. Results of simulated fibers without stem and $\beta=-85^\circ$. Output parameters of trajectories obtained from simulations in the region sampled from $\alpha=20^\circ$ to $\alpha=50^\circ$ in steps of 2.5° and from $L=3.4$ nm to $L=10.2$ nm in steps of 0.85 nm. (A) Linear mass density. No edge or peak to be detected. Also for the shorter linker DNA lengths the density doesn't seem to grow as fast as seen for fibers with stem. (B) Total energy. Shows inverse properties in relation to results from simulations with stem – lower energies for shorter linker DNAs (C) Fiber diameter. Because of the missing stem the diameters are smaller in average. (D) Persistence length. The scattering is striking compared to Figure 3.6 D because here the axis is half as high. (E) effective entry-exit angle. the measured output angle decreases with increasing input angle (F) average nucleosome tilt.

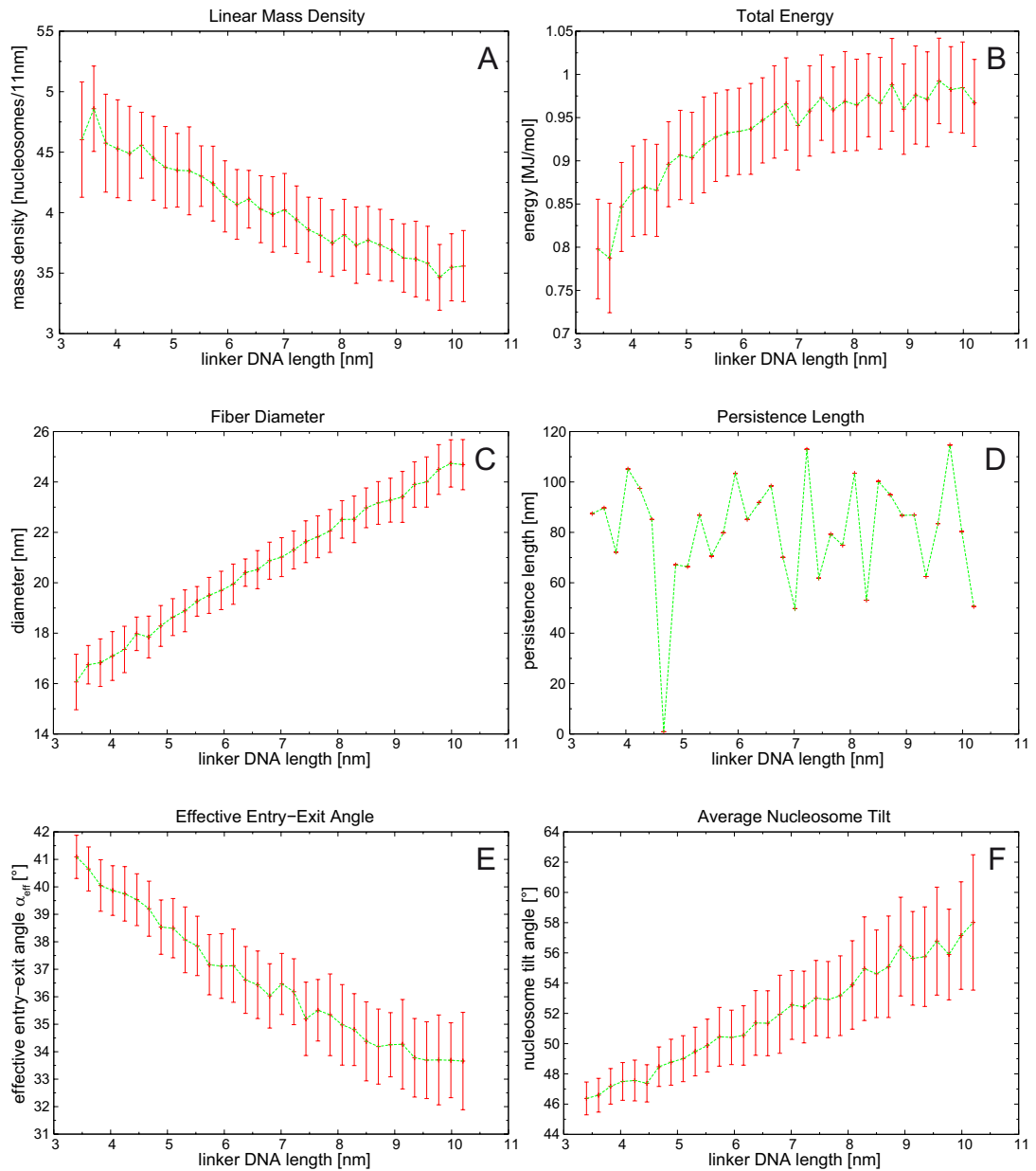


Figure 3.12. Simulations without stem using starting parameters $\alpha=40^\circ$ and $\beta=-85^\circ$. Plotted are the output parameters as functions of the linker DNA length L . (A) Linear mass density. Decreases linearly with increasing linker DNA lengths. (B) Total energy. lower energies for longer linker DNA lengths (C) Fiber diameter. The diameter grows linear to the linker DNA length. (D) Persistence length. (E) Effective entry-exit angle. the measured output angle decreases with increasing input angle (F) Average nucleosome tilt.

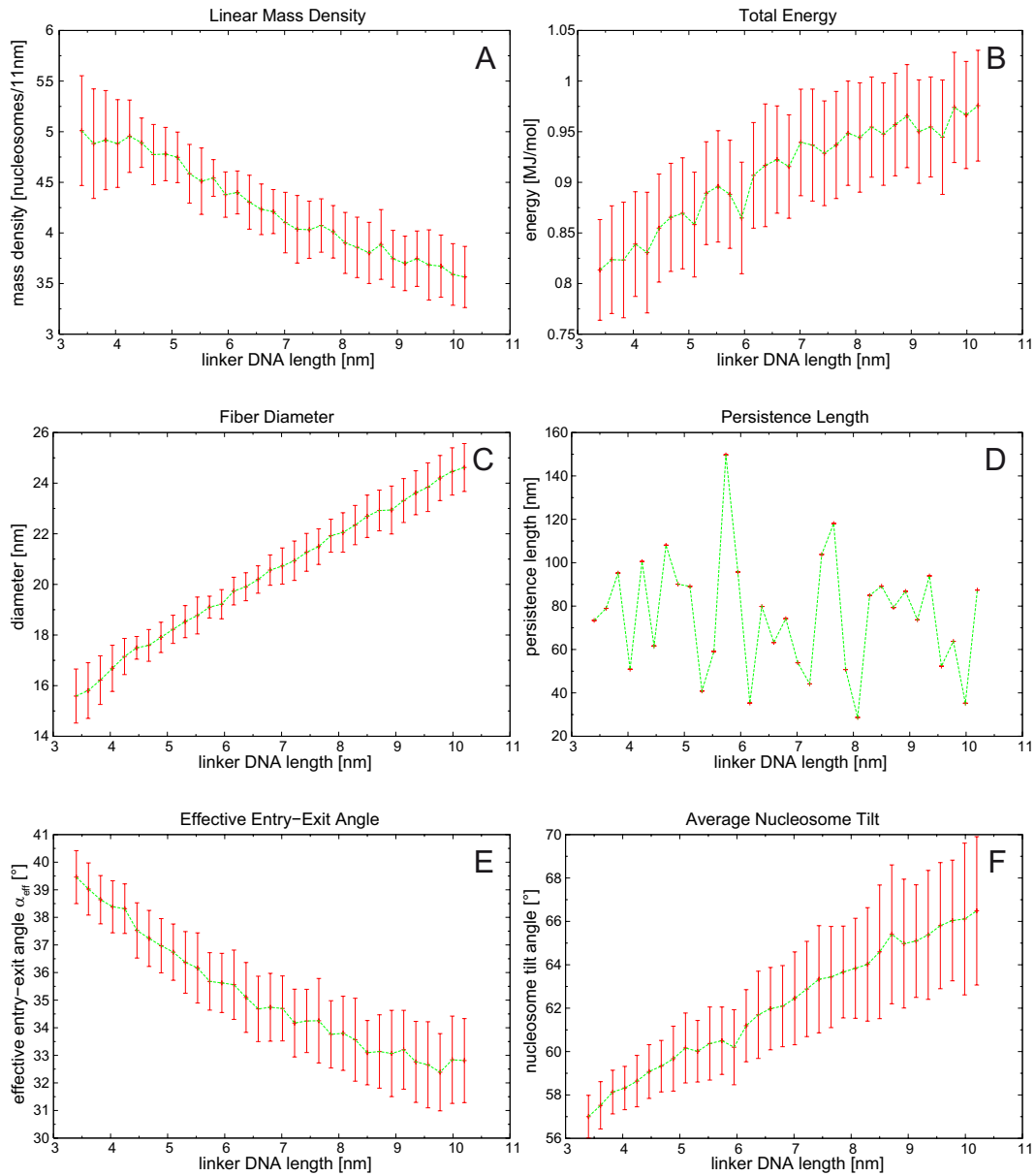


Figure 3.13: Simulations without stem using starting parameters $\alpha=40^\circ$ and $\beta=110^\circ$. Plotted are the output parameters as functions of the linker DNA length L . (A) Linear mass density. Decreases linearly with increasing linker DNA lengths. (B) Total energy. lower energies for longer linker DNA lengths (C) Fiber diameter. The diameter grows linear to the linker DNA length. (D) Persistence length. (E) Effective entry-exit angle. the measured output angle decreases with increasing input angle (F) Average nucleosome tilt.

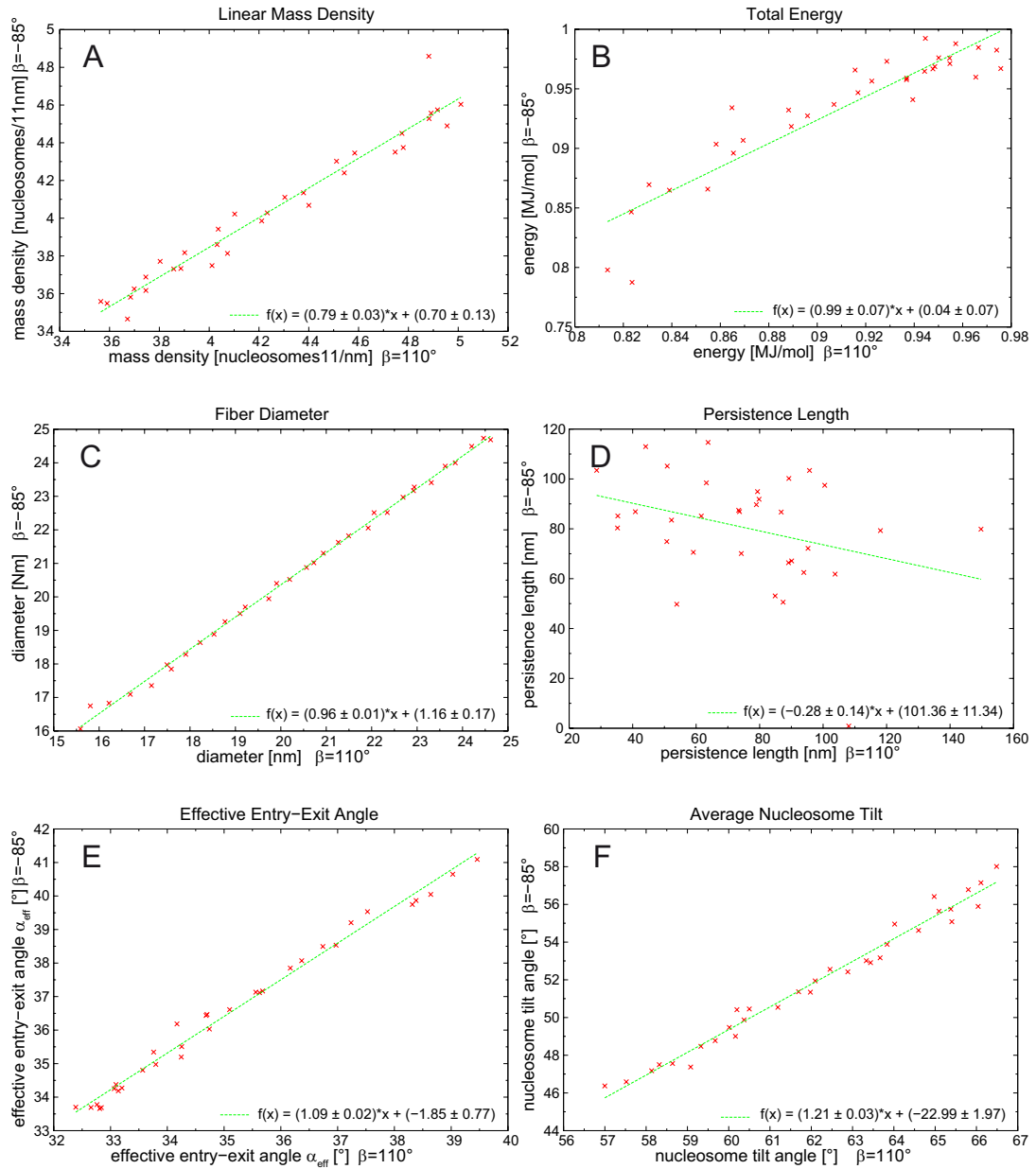


Figure 3.14: Results of simulations without stem and $\alpha=40^\circ$. Comparing values for $\beta=-85^\circ$ with those for $\beta=110^\circ$. Results with matching linker DNA length L were plotted. (A) linear mass density. (B) total energy. (C) fiber diameter. (D) persistence length. (E) effective entry-exit angle. (F) average nucleosome tilt. No significant differences can be observed except for (D) which comes from the scattering of the persistence length. In (A) the outlier (right side above) seems to falsify the value of the slope (F) Reasonably the different torsion angles lead to an offset in the nucleosome tilt angle.

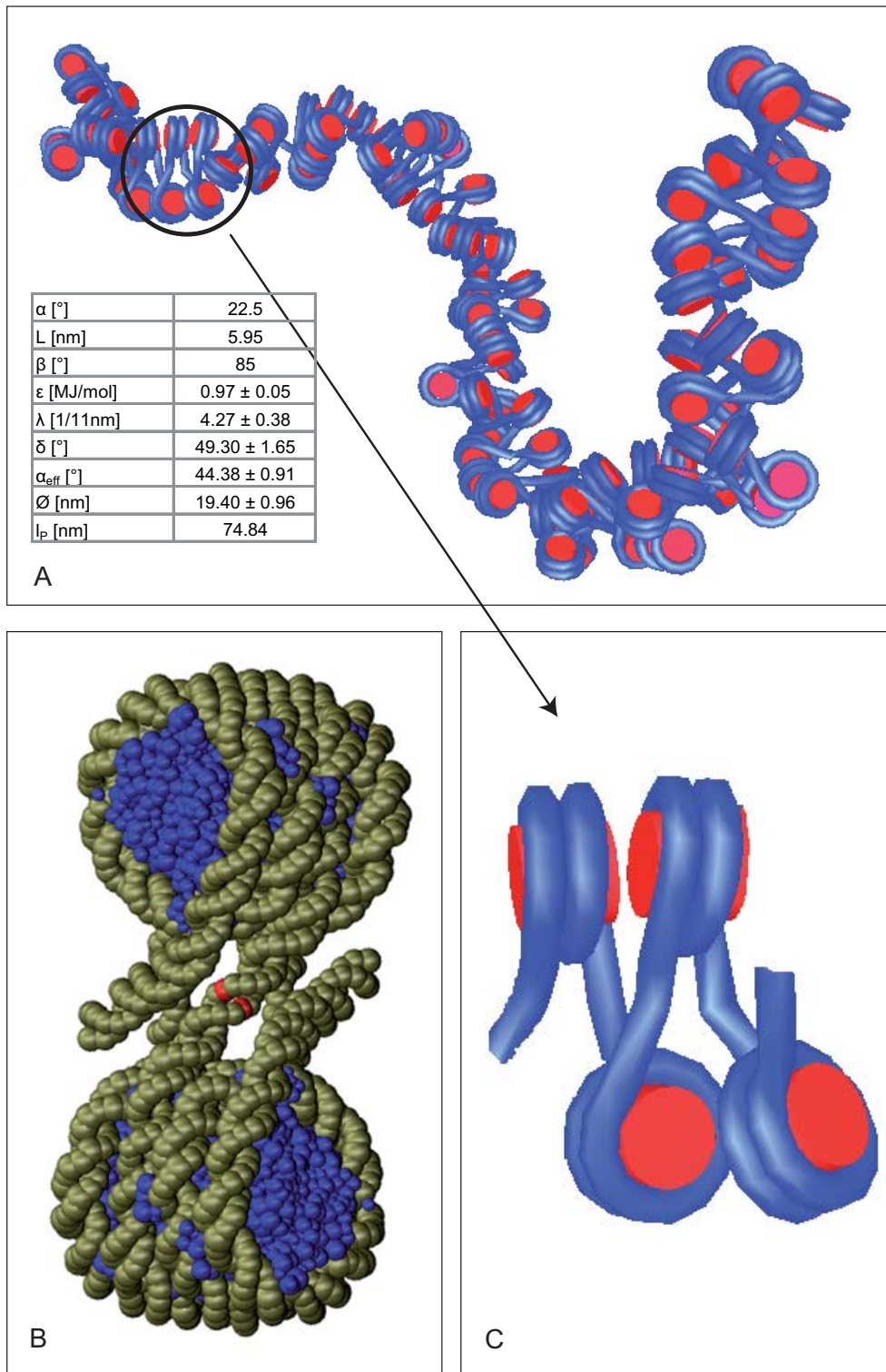


Figure 3.15: (A) A fiber conformation from simulations with nearly tetranucleosome parameters ($\alpha = 23^\circ$ and $L = 6\text{nm}$ and $\beta = -85^\circ$). (B) tetranucleosome structure according to Schalch et. al. [23] (C) enlarged section from (A) showing similarities to (B).

4 Discussion

The aim of this project was to examine the relation between the local geometry of the nucleosome and the organization of the chain of nucleosomes into a chromatin fiber. This is an important question as the packaging of the DNA into the fiber and its higher order organization has long been recognized as a mechanism to control the accessibility of the DNA. It is furthermore highly relevant for protein-DNA interactions involved in processes like transcription, replication repair and recombination [38, 39]. To examine this point a computer model of the 30 nm chromatin fiber was used. In this model the nucleosomes and DNA are modeled with elastically connected ellipsoids. Their configurations in thermal equilibrium are sampled using a Monte Carlo algorithm extending a previously used approach [4, 5]. In the simulations the units interact by stretching, torsion and bending forces modeled by harmonic potentials. The interaction of DNA by its negatively charged phosphate backbone is described by a Debye-Hückel potential and the interchromatosomal interaction by an ellipsoid potential. A classical Metropolis Monte Carlo procedure was then used to sample a statistical relevant set of configurations with pivotal and rotational moves at a certain temperature.

In Figure 4.1 two examples are shown from simulations of a chain with 100 nucleosomes as described in section 3.2. They represent ensembles with high (Figure 4.1 A) and low (Figure 4.1 B) linear mass densities. As evident from the results obtained here these large changes in compaction and mass density are achieved by relatively small changes of the entry-exit angle and the linker. The opening of the structure in Fig. 4.1 might reflect structural transitions in the fiber organization relevant for biological processes. The systematic analysis of the parameter space conducted here (Figure 3.6, 3.11) demonstrates that the dependence of the mass density on the entry-exit angle and the linker length show a peak for a specific combination of values. This observation indicates a

mechanism by which factors like histone acetylation/methylation, DNA methylation and linker histone H1 binding could modulate a conformational transition of the 30 nm fiber between two conformations. An inactive compacted state and a more flexible and an “open” conformation, in which the DNA can be transcribed [40, 41].

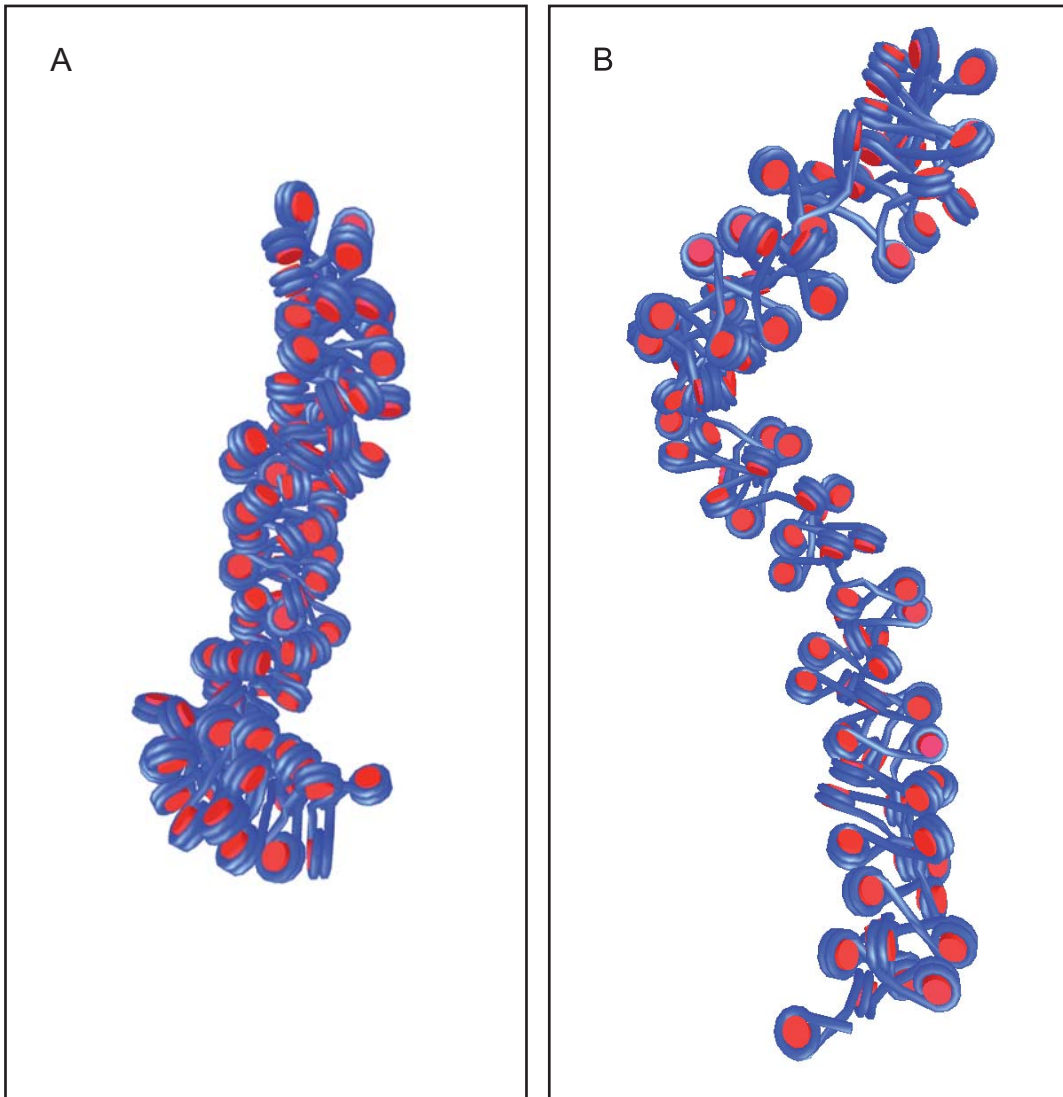


Figure 4.1. Two fibers of representative trajectories with high and the low mass density (see Figure 3.6). (A) entry-exit angle $\alpha_{\text{eff}}=32.9$, linker DNA length $L=5.5$ nm, mass density 7.4 nucleosomes/11 nm (B) $\alpha_{\text{eff}}=15.7^\circ$, $L=10.2$ nm, mass density 4.5 nucleosomes/11 nm. This means that the average contour length of fibers from trajectory B is approximately 1.6 times greater than that of fibers from trajectory A.

While the mass density is an important parameter to describe the overall fiber organization also other parameters like the total energy, fiber diameter, persistence length, effective entry-exit angle and the nucleosome tilt angle were analyzed to identify their relation with the fiber compaction state. The highest correlation with changes in the mass density were observed for the effective entry-exit angle (Figure 3.7E) and the nucleosome tilt angle (Figure 3.7F). Interestingly, these two parameters do not show a peak like behavior but a true transition at the regime where large changes of the mass density occur. Furthermore the transition occurred around $\alpha \sim 35^\circ$, which is exactly the value of the entry-exit angle determined experimentally by cryo-electron microscopy [24]. The diameter of 30 nm, which is regarded as the standard value for the chromatin fiber *in vivo*, is reached only for fibers with a stem (~ 28 nm diameter)(Fig. 3.7 C). For simulations without stem structure the fiber diameter was significantly smaller and below 25 nm (Fig. 3.12 C). This points to a role of linker histone H1 for establishing a fully compacted fiber structure. As expected the fiber diameter increased in a continuous manner as shown in Figure 3.7 and 3.12. For the persistence length of the chromatin fiber a decrease from a value of ~ 265 nm for 200 base pair repeat length to ~ 50 nm for 212 base pair repeat lengths was observed (Figure 3.7 D). This relatively large variation correlates with values obtained experimentally [42]. By fluorescence *in situ* hybridization (FISH) experiments the distance of dye labeled probes in fixed human fibroblast cells was measured by confocal laser scanning microscopy and compared to the genomic distance [43]. Analysis of the data provided values of $l = 137 - 440$ nm [44] and $l = 196 - 272$ nm [45], with an average value of $l \approx 250$ nm. The analysis of single chromatin fibers *in vitro* yielded a higher flexibility of $l = 60$ nm [46] that was also derived from *in vivo* cross-linking experiments [47].

In accordance with previous results the maximum values for the fibers with a nucleosome stem [37] are around 6 nucleosomes per 11 nm fiber. This is in good agreement with experimental values of the fiber mass density [5]. It suggests that

the nucleosome model and associated interaction energies are adequate for a faithful description of the global fiber structure. However, significant differences between theory and experiment are observed for the dependency of the linear mass density on the ionic strength of the solution (G. Wedemann, unpublished). This suggests that the electrostatic potential of the DNA shielded by salt ions as described by the Debye-Hückel approximation is not suitable to describe the interactions at short distances. In addition, a dependence of the nucleosome-nucleosome interaction potential with ionic strength needs to be implemented into the model.

The strong dependence of the fiber organization on the local nucleosome geometry observed here raises the question if the current model is sufficiently accurate to model the nucleosome. Some limitations are evident from a comparison of the Monte Carlo simulation model and the tetranucleosome structure derived from x-ray crystallography [23], as well as the results of single molecule stretching experiments [22, 46]. First, the nucleosome-nucleosome interaction energy used here was kept at a relatively low constant value ($\sim 3 RT$) and the effect of higher interaction energies on the fiber conformation needs to be examined. Second, the nucleosome is modeled as an ellipsoid in the simulations but its actual shape is that of a cylinder with 11 nm diameter and 5.5 nm height. It is expected that certain aspects of the mass density behavior of the fiber cannot be reproduced with the ellipsoid model due to geometric differences that become significant for close packaging of nucleosomes. Third, the geometry of the DNA at the nucleosome is much more flexible than it can be represented by the simple two-angle model. Accordingly, this description needs to be extended by additional degrees of freedom so that the trajectory of DNA entering and leaving the nucleosome can be described in a more realistic manner. This would allow us to study nucleosome and chromatosome conformations predicted from model building at atomic resolution [48, 49]. In continuation of the work presented here, the above points will be addressed in the development of improved chromatin

fiber models currently in progress. As shown in this thesis the use of numerical simulations provides valuable insights for identifying parameters that determine the dynamic conformation of the chromatin fiber. These are potential targets for biological factors that affect the conformation of the fiber and thus change transcription activity. Thus, the Monte Carlo simulations allow us to make predictions on how these would have to interact with the chromatin fiber to exert their biological function.

5 Literature

1. van Holde, K.E., *Chromatin*. 1989, Heidelberg: Springer.
2. Coffee, B., et al., *Acetylated histones are associated with FMR1 in normal but not fragile X-syndrome cells*. *Nature Genetics*, 1999. **22**(1): p. 98-101.
3. Kouzarides, T., *Histone acetylases and deacetylases in cell proliferation*. *Current Opinion in Genetics and Development*, 1999. **9**(1): p. 40-48.
4. Wedemann, G., *Simulation von Struktur und Strukturregulation von Chromatin*, in *Fachbereich Physik*. 1999, Bergische Universität - Gesamthochschule Wuppertal: Wuppertal.
5. Wedemann, G. and J. Langowski, *Computer simulation of the 30-nanometer chromatin fiber*. *Biophys J*, 2002. **82**(6): p. 2847-59.
6. Mandelkern, M., et al., *The dimensions of DNA in solution*. *J Mol Biol*, 1981. **152**(1): p. 153-61.
7. Luger, K., et al., *Characterization of nucleosome core particles containing histone proteins made in bacteria*. *Journal of Molecular Biology*, 1997. **272**(3): p. 301-311.
8. Davey, C.A., et al., *Solvent mediated interactions in the structure of the nucleosome core particle at 1.9 Å resolution*. *Journal of Molecular Biology*, 2002. **319**(5): p. 1097-1113.
9. Harp, J.M., et al., *Asymmetries in the nucleosome core particle at 2.5 Å resolution*. *Acta Crystallogr. D Biol. Crystallogr.*, 2000. **56 Pt 12**: p. 1513-1534.
10. Graziano, V., et al., *Histone H1 is located in the interior of the chromatin 30-nm filament*. *Nature*, 1994. **368**(6469): p. 351-354.
11. Zhou, Y.B., et al., *Position and orientation of the globular domain of linker histone H5 on the nucleosome*. *Nature*, 1998. **395**(6700): p. 402-405.
12. Ramakrishnan, V., *Histone H1 and chromatin higher-order structure*. *Critical Reviews in Eukaryotic Gene Expression*, 1997. **7**(3): p. 215-230.
13. Zlatanova, J. and K. van Holde, *The linker histones and chromatin structure: new twists*. *Progress in Nucleic Acid Research and Molecular Biology*, 1996. **52**: p. 217-259.
14. Bharath, M.M., N.R. Chandra, and M.R. Rao, *Molecular modeling of the chromatosome particle*. *Nucleic Acids Res*, 2003. **31**(14): p. 4264-4274.

15. Woodcock, C.L. and S. Dimitrov, *Higher-order structure of chromatin and chromosomes*. Current Opinion in Genetics and Development, 2001. **11**(2): p. 130-135.
16. Cremer, M., et al., *Three dimensional analysis of histone methylation patterns in normal and tumor cell nuclei*. Eur J Histochem, 2004. **48**(1): p. 15-28.
17. Cremer, T., et al., *Chromosome territories - a functional nuclear landscape*. Current opinion in cell biology, 2006. **18**(3): p. 307-16.
18. Finch, J.T. and A. Klug, *Solenoidal model for superstructure in chromatin*. Proceedings of the National Academy of Sciences of the USA, 1976. **73**(6): p. 1897-1901.
19. Woodcock, C.L., et al., *A chromatin folding model that incorporates linker variability generates fibers resembling the native structures*. Proceedings of the National Academy of Sciences of the USA, 1993. **90**(19): p. 9021-9025.
20. Leuba, S.H., et al., *Three-dimensional structure of extended chromatin fibers as revealed by tapping-mode scanning force microscopy*. Proceedings of the National Academy of Sciences of the USA, 1994. **91**(24): p. 11621-11625.
21. Ehrlich, L., *Simulation der Struktur und Dynamik von Chromatin mit Brownscher Dynamik*, in *Fakultät für Physik und Astronomie*. 1996, Ruprecht-Karls-Universität: Heidelberg. p. 75.
22. Katritch, V., C. Bustamante, and W.K. Olson, *Pulling chromatin fibers: computer simulations of direct physical micromanipulations*. Journal of Molecular Biology, 2000. **295**(1): p. 29-40.
23. Schalch, T., et al., *X-ray structure of a tetranucleosome and its implications for the chromatin fibre*. Nature, 2005. **436**(7047): p. 138-41.
24. Bednar, J., et al., *Nucleosomes, linker DNA, and linker histone form a unique structural motif that directs the higher-order folding and compaction of chromatin*. Proceedings of the National Academy of Sciences of the USA, 1998. **95**(24): p. 14173-14178.
25. Schellman, J.A. and D. Stigter, *Electrical double layer, zeta potential, and electrophoretic charge of double-stranded DNA*. Biopolymers, 1977. **16**: p. 1415-1434.
26. Stigter, D., *Interactions of highly charged colloidal cylinders with applications to double stranded DNA*. Biopolymers, 1977. **16**: p. 1435-1448.

27. Gay, J.G. and B.J. Berne, *Modification of the overlap potential to mimic a linear site--site potential*. Journal of Chemical Physics, 1981. **74**: p. 3316--3319.
28. Kabadi, V., *Molecular dynamics of fluids: the Gaussian overlap model II*. Ber. Bunsenges. Phys. Chem., 1986a. **90**: p. 327-332.
29. Kabadi, V., *Statistical mechanics of non-spherical molecules: spherical harmonic expansions on non-spherical surfaces II: Gay-Berne Gaussssian overlap potential*. Ber. Bunsenges. Phys. Chem., 1986b. **90**: p. 332-339.
30. Luckhurst, G.R. and P.S.J. Simmonds, *Computer Simulation studies of anisotropic systems. XXI. Parametrization of the Gay-Berne potential for model mesogens*. Molecular Physics, 1993. **80**(2): p. 233-252.
31. Binder, K. and D.W. Herrmann, *Monte Carlo Simulation in Statistical Physics*. Springer Series in Solid-State Sciences, ed. P. Fulde. Vol. 80. 1992, Heidelberg: Springer-Verlag.
32. Allen, M.P. and D.J. Tildesley, *Computer simulations of liquids*. 1989, Oxford: Oxford University Press.
33. Metropolis, N., et al., *Equation of state calculations by fast computing machines*. The Journal of Chemical Physics, 1953. **21**: p. 1087-1092.
34. Stroustrup, B., *The C++ Programming Language*. Special 3rd ed. 2000, Amsterdam: Addison-Wesley Longman.
35. Powers, S., et al., *Unix Power Tools*. 3rd ed. 2002: O'Reilly.
36. Thomas, D., C. Fowler, and A. Hunt, *Programming Ruby*. 2nd ed. 2004: Pragmatic Programmers.
37. Gerchman, S.E. and V. Ramakrishnan, *Chromatin higher-order structure studied by neutron scattering and scanning transmission electron microscopy*. Proceedings of the National Academy of Sciences of the USA, 1987. **84**(22): p. 7802-7806.
38. Woodcock, C.L., *Chromatin architecture*. Curr Opin Struct Biol, 2006. **16**(2): p. 213-20.
39. Cremer, T. and C. Cremer, *Chromosome Territories, Nuclear Architecture and Gene Regulation in Mammalian Cells*. Nature Reviews Genetics, 2001. **2**(4): p. 292-301.
40. Kadonaga, J.T., *Eukaryotic transcription: an interlaced network of transcription factors and chromatin-modifying machines*. Cell, 1998. **92**(3): p. 307-313.
41. Strahl, B.D. and C.D. Allis, *The language of covalent histone modifications*. Nature, 2000. **403**(6765): p. 41-45.

42. Rippe, K., *Making contacts on a nucleic acid polymer*. Trends in Biochemical Sciences, 2001. **26**(12): p. 733-740.
43. van den Engh, G., R. Sachs, and B.J. Trask, *Estimating genomic distance from DNA sequence location in cell nuclei by a random walk model*. Science, 1992. **257**: p. 1410-1412.
44. Hahnfeldt, P., et al., *Polymer models for interphase chromosomes*. Proceedings of the National Academy of Sciences of the USA, 1993. **90**: p. 7854-7858.
45. Ostashevsky, J.Y. and C.S. Lange, *The 30 nm chromatin fiber as a flexible polymer*. Journal of Biomolecular Structure & Dynamics, 1994. **11**: p. 813-820.
46. Cui, Y. and C. Bustamante, *Pulling a single chromatin fiber reveals the forces that maintain its higher-order structure*. Proceedings of the National Academy of Sciences of the USA, 2000. **97**(1): p. 127-132.
47. Dekker, J., et al., *Capturing chromosome conformation*. Science, 2002. **295**(5558): p. 1306-1311.
48. Rippe, K., J. Mazurkiewicz, and N. Kepper, *Interactions of histones with DNA: nucleosome assembly, stability and dynamics*, in *DNA interactions with polymers and surfactants*, R. Dias and B. Lindman, Editors. 2007, Blackwell Publishing: London.
49. Robinson, P.J., et al., *EM measurements define the dimensions of the "30-nm" chromatin fiber: evidence for a compact, interdigitated structure*. Proceedings of the National Academy of Sciences of the USA, 2006. **103**(17): p. 6506-11.
50. Lopez, A., et al., *Fluorescence Recovery After Photobleaching (Frap) Experiments Under Conditions of Uniform Disk Illumination - Critical Comparison of Analytical Solutions, and a New Mathematical Method for Calculation of Diffusion Coefficient-D*. Biophysical Journal, 1988. **53**(6): p. 963-970.

6 Appendix

runsim.rb

```
#!/usr/local/bin/ruby -w

#set field- and record-separators used by print method
$,="\t"
$\ = "\n"

SimSteps = 2_500_000
SaveAfterSteps = 5000

CLIENT_NUM=7
CLIENT_NUM_START=0
PROCESSES_PER_CLIENT=2

joblist = Array.new

20.step(50.5) do |alpha|
  3.4.step(10.2,0.2125) do |llength|
    job = []

    strAlpha = alpha.to_s
    strLlength = llength.to_s

    dirname1 = 'a' + strAlpha
    dirname2 = '_l' + strLlength
    dirname = dirname1 + dirname2 + '_r1'

    job << dirname
    job << strAlpha
    job << strLlength
    joblist << job
  end
end

joblist.sort! {|a,b| a.values_at(1,2) <=> b.values_at(1,2)}

#open client_num*processes_per_client threads

threads = []

(CLIENT_NUM_START..CLIENT_NUM+CLIENT_NUM_START-1).each do
|client_num|
  1.upto(PROCESSES_PER_CLIENT) do |proc_num|
    threads << Thread.new(client_num, proc_num) do
      until joblist.empty?
        job = joblist.pop
        unless client_num==0
          client = "kn" + format("%02d",client_num)

```

```

else
  client = "master"
end
print "Start thread #{client}_#{proc_num} with", job
rundir = "~/runningsims/"
dirname = job[0]
strAlpha = job[1]
strLlength = job[2]
Dir.mkdir(dirname)
`rsh #{client} "~/Manie/bin/chromCreat t 100 #{strLlength}
  #{strAlpha} 85 gb > #{rundir + dirname
  + "/" + dirname}.trj"`
`rsh #{client} "cp #{rundir + dirname + "/" + dirname}.trj
  #{rundir + dirname + "/" + dirname}_start.trj"`
`rsh #{client} "~/Manie/bin/createDNATable #{rundir
  + dirname + "/" + dirname}.trj"`
`rsh #{client} "cp estatdna_* #{rundir + dirname}"`
`rsh #{client} "~/Manie/bin/backebackekuchen mc #{rundir
  + dirname + "/" + dirname}.trj cube #{SimSteps}
  #{SaveAfterSteps}" > #{rundir + dirname + "/"
  + dirname}.msgout `
print "End thread #{client}_#{proc_num}"
end
end
end
end

threads.each {|th| th.join}

```

evaluation.rb

```

class Evaluation
  attr_accessor :name, :run_analysis_execs, :traj_names,
    :allfilesinonedir, :gnuplot_version,
    :filenameextensions, :no3d

  def initialize
    @name = "evaluation"
    @run_analysis_execs = false
    @allfilesinonedir = false
    @traj_names = Dir['a*_l*_*']
    @filenameextensions = Hash.new
    @no3d = false
  end

  def start
    analyzer = Analyzer.new
    analyzer.filenameextensions.merge!(self.filenameextensions)
    #...set analyzer attributes

    unless run_analysis_execs == false
      @traj_names.each do |trajName|

```

```
Dir.chdir(trajName) unless @allfilesinonedir == true
analyzer.traj_name = trajName
@run_analysis_execs.each do |progr|
  analyzer.run_analysis_exec(progr)
end
Dir.chdir('.') unless @allfilesinonedir == true
end
exit
end

res = Result.new(analyzer,self)
res.read(traj_names)

begin
  eval_dir = Dir.mkdir(@name)
rescue SystemCallError
  if $!.class == Errno::EEXIST then
    puts "An evaluation named '#@name' already exists!"
    puts "If you want to overwrite it press enter or else"
    puts "enter another name for the evaluation:"
    eval_name = gets
    unless eval_name == "\n"
      @name = eval_name.chomp
      retry
    end
  else
    $stderr.print "System Call Error: " + $!
    raise
  end
end
Dir.chdir(@name)
res.write_all

unless @no3d
  res.write3d("alpha", "llength", "len_avg")
  res.write3d("alpha", "llength", "dia_avg")
  res.write3d("alpha", "llength", "mbd_avg")
  res.write3d("alpha", "llength", "mbdActual_avg")
  res.write3d("alpha", "llength", "nucsInTube_avg")
  res.write3d("alpha", "llength", "tiltavg_avg")
  res.write3d("alpha", "llength", "perslen")
  res.write3d("alpha", "llength", "alphaeffavg_avg")
  res.write3d("alpha", "llength", "energy_avg")
  res.write2d("llength", "alphaeffavg_avg",
    "energy_avg", "energy_dev")
  res.write2d("llength", "alphaeffavg_avg",
    "dia_avg", "dia_dev")
  res.write2d("llength", "alphaeffavg_avg",
    "mbd_avg", "mbd_dev")
  res.write2d("llength", "alphaeffavg_avg",
    "len_avg", "len_dev")
  res.write2d("llength", "alphaeffavg_avg",
    "mbdActual_avg", "mbdActual_dev")
end
```

```

    res.write2d("llength", "alphaeffavg_avg",
               "tiltavg_avg", "tiltavg_dev")
    res.write2d("llength", "alphaeffavg_avg", "perslen")
    res.write2d("llength", "alphaeffavg_avg",
               "alphaeffavg_avg", "alphaeffavg_dev")
    res.write2d("llength", "alphaeffavg_avg",
               "nucsInTube_avg", "nucsInTube_dev")
end

res.write2d("alpha", "llength", "energy_avg", "energy_dev")
res.write2d("alpha", "llength", "dia_avg", "dia_dev")
res.write2d("alpha", "llength", "mbd_avg", "mbd_dev")
res.write2d("alpha", "llength", "len_avg", "len_dev")
res.write2d("alpha", "llength", "mbdActual_avg",
            "mbdActual_dev")
res.write2d("alpha", "llength", "tiltavg_avg", "tiltavg_dev")
res.write2d("alpha", "llength", "perslen")
res.write2d("alpha", "llength", "alphaeffavg_avg",
            "alphaeffavg_dev")
res.write2d("alpha", "llength", "nucsInTube_avg",
            "nucsInTube_dev")
Dir.chdir("../")
end
end

```

analyzer.rb:

```

class Analyzer
  attr_accessor :traj_name, :filemextensions

  def initialize(nucsInTubeDefault = 80, ignoredConfigs = 101)
    @ignored_config_count = ignoredConfigs
    @nucsInTubeDefault = nucsInTubeDefault
    @filemextensions =
      Hash['trajectory', '.trj', 'chromLengthDiameter', '.lendia',
          'persistenceLength', '.pers', 'nucleosomeTilt', '.tilt',
          'energies', '.energy', 'compEffAlpha', '.effAlpha',
          'compEffAlpha', '.effAlpha', 'energy', '.energyOutput']
  end

  def run_analysis_exec(progname)
    res_file = @traj_name + @filemextensions[progname]
    traj_file = @traj_name + @filemextensions['trajectory']
    print "running '#{progname}' on:", traj_file
    `#{progname} #{traj_file} > #{res_file}`
  end

  def get_length_diameter
    h = Hash.new
    a=%w(mbd_avg mbd_dev len_avg len_dev mbdActual_avg
        mbdActual_dev nucsInTube_avg nucsInTube_dev dia_avg dia_dev)
    a.each {|key| h[key]=0}
  end
end

```

```

res_file = @traj_name +
@filenameextensions['chromLengthDiameter']
#start the block for error handling
begin
  f = File::open(res_file)
  #read data into an array
  a = f.readlines
  len=0; lenSq = 0; mbd = 0; mbdSq = 0; rad=0; radSq = 0
  lmd_actual=0; lmd_actual_sq=0; nucs_in_tube = 0;
  nucs_in_tube_sq = 0
  #loop through the results file and sum up values
  a[@ignored_config_count..-1].each do |l|
    lenRad = l.split
    length = lenRad[0].to_f
    radius = lenRad[1].to_f
    radiusMax = lenRad[2].to_f
    nucsInTube = lenRad[3].to_i

    len += length
    lenSq += length*length
    mbd += @nucsInTubeDefault/length*11
    mbdSq += (@nucsInTubeDefault/length*11)**2
    lmd_actual += nucsInTube/length*11
    lmd_actual_sq += (nucsInTube/length*11)**2
    nucs_in_tube += nucsInTube
    nucs_in_tube_sq += nucsInTube**2
    rad += radius
    radSq += radius**2
  end
  configCount = a.size

  #calculate the averages and standard deviations
  #mass-density
  mbdAvg = mbd/(configCount-@ignored_config_count)
  mbdDev = Math.sqrt(((configCount-@ignored_config_count)
    *mbdSq-mbd**2)/(configCount-@ignored_config_count)/
    ((configCount-@ignored_config_count)-1))
  lenAvg = len/(configCount-@ignored_config_count)
  lenDev = Math.sqrt(((configCount-@ignored_config_count)
    *lenSq-len**2)/(configCount-@ignored_config_count)/
    ((configCount-@ignored_config_count)-1))
  #mass density the 'supposed to be' way
  lmd_actual_avg = lmd_actual/
    (configCount-@ignored_config_count)
  lmd_actual_dev = Math.sqrt(((configCount-
    @ignored_config_count)*lmd_actual_sq-lmd_actual**2)/
    (configCount-@ignored_config_count)/
    ((configCount-@ignored_config_count)-1))
  nucs_in_tube_avg = nucs_in_tube.to_f/(configCount-
    @ignored_config_count)
  nucs_in_tube_dev = Math.sqrt(((configCount-
    @ignored_config_count)*nucs_in_tube_sq-
    nucs_in_tube**2)/(configCount-@ignored_config_count)/
    ((configCount-@ignored_config_count)-1))

```

```

rad_avg = rad/(configCount-@ignored_config_count)
rad_dev = Math.sqrt(((configCount-@ignored_config_count)*
  radSq-rad**2)/(configCount-@ignored_config_count)/
  ((configCount-@ignored_config_count)-1))

dia_avg = rad_avg*2
dia_dev = rad_dev*2

h["mbd_avg"] = mbdAvg.round(6)
h["mbd_dev"] = mbdDev.round(6)
h["len_avg"] = lenAvg.round(6)
h["len_dev"] = lenDev.round(6)
h["mbdActual_avg"] = lmd_actual_avg.round(6)
h["mbdActual_dev"] = lmd_actual_dev.round(6)
h["nucsInTube_avg"] = nucs_in_tube_avg
h["nucsInTube_dev"] = nucs_in_tube_dev.round(6)
h["dia_avg"] = dia_avg.round(6)
h["dia_dev"] = dia_dev.round(6)

rescue
  if $!.class == Errno::ENOENT then
    $stderr.print "There is no file called '#{res_file}'"
  else
    $stderr.print "Standard Error: " + $!
  end
  $stderr.print "Still a record with zeros instead of the
    expected values will be returned!"
ensure
  return h
end
end

def get_nucleosome_tilt
  h = Hash.new
  h["tiltavg_avg"] = 0
  h["tiltavg_dev"] = 0

  res_file = @traj_name + @filenamextensions['nucleosomeTilt']
  begin
    f = File::new(res_file)
    a = f.readlines
    sum = 0; sumSq = 0 #--; sumDev = 0; sumDevSq = 0
    a[@ignored_config_count..-1].each do |l|

      angle = l.split
      angleAvg = angle[0].to_f
      #angleDev = angle[1].to_f

      sum += angleAvg
      sumSq += angleAvg**2
      #--sumDev += angleDev
      #--sumDevSq += angleDev*angleDev
    end
  end
end

```

```
end
configCount = a.size
avg = sum/(configCount-@ignored_config_count)
dev = Math.sqrt(((configCount-@ignored_config_count)
  *sumSq-sum**2)/(configCount-@ignored_config_count)/
  ((configCount-@ignored_config_count)-1))

avg = avg/Math::PI*180
dev = dev/Math::PI*180

h["tiltavg_avg"] = avg.round(4)
h["tiltavg_dev"] = dev.round(4)
rescue
  if $!.class == Errno::ENOENT then
    $stderr.print "There is no file called '#{res_file}'"
  else
    $stderr.print "Standard Error: " + $!
  end
  $stderr.print "Still a record with zeros instead of the
    expected values will be returned!"
ensure
  return h
end
end

def get_alphaeff
  h = Hash.new
  h = Hash["alphaeffavg_avg", 0,"alphaeffavg_dev", 0]

  res_file = @traj_name + @filenameextensions['compEffAlpha']
  begin
    f = File::new(res_file)
    a = f.readlines
    sum = 0; sumSq = 0 #--; sumDev = 0; sumDevSq = 0
    a[@ignored_config_count..-1].each do |l|

      angle = l.split
      angleAvg = angle[0].to_f
      #angleDev = angle[1].to_f

      sum += angleAvg
      sumSq += angleAvg**2
    end
    configCount = a.size
    avg = sum/(configCount-@ignored_config_count)
    dev = Math.sqrt(((configCount-@ignored_config_count)
      *sumSq-sum**2)/(configCount-@ignored_config_count)/
      ((configCount-@ignored_config_count)-1))
    avg = avg/Math::PI*180
    dev = dev/Math::PI*180

    h["alphaeffavg_avg"] = avg.round(4)
    h["alphaeffavg_dev"] = dev.round(4)
  rescue
```

```

    if $!.class == Errno::ENOENT then
      $stderr.print "There is no file called '#{res_file}'"
    else
      $stderr.print "Standard Error: " + $!
    end
    $stderr.print "Still a record with zeros instead of the
      expected values will be returned!"
  ensure
    return h
  end
end

def get_energy
  h = Hash.new
  h = Hash["energy_avg", 0, "energy_dev", 0]

  res_file = @traj_name + @filenamextensions['trajectory']+
    @filenamextensions['energy']
  begin
    f = File::new(res_file)
    a = f.readlines
    a.shift #discard first line
    sum = 0; sumSq = 0
    a[@ignored_config_count..-1].each do |l|

      values = l.split
      value = values[1].to_f

      sum += value/1000
      sumSq += (value/1000)**2

    end
    configCount = a.size
    avg = sum/(configCount-@ignored_config_count)
    dev = Math.sqrt(((configCount-@ignored_config_count)
      *sumSq-sum**2)/(configCount-@ignored_config_count)/
      ((configCount-@ignored_config_count)-1))

    h["energy_avg"] = avg.round(4)
    h["energy_dev"] = dev.round(4)
  rescue
    if $!.class == Errno::ENOENT then
      $stderr.print "There is no file called '#{res_file}'"
    else
      $stderr.print "Standard Error: " + $!
    end
    $stderr.print "Still a record with zeros instead of the
      expected values will be returned!"
  ensure
    return h
  end
end

```



```

def getPersistenceLength
  perslength = 0
  res_file = @traj_name+ @filenamextensions['persistenceLength']
  begin
    #read output from persistenceLength into an array
    f = File::new(res_file)
    pers = f.readlines

    #the last line (=the last Object in the array)
    #is the value of the persistence length
    perslength = pers[pers.length-1].chomp[1..-1].to_f.round(4)
  rescue
    if $!.class == Errno::ENOENT then
      $stderr.print "There is no file called '#{res_file}'"
    else
      $stderr.print "Standard Error: " + $!
    end
    $stderr.print "Still a record with zeros instead of the
      expected values will be returned!"
  ensure
    return {"perslen" => perslength}
  end
end
end

class Float
  def round(nachkomma)
    tmpfloat = self*10**nachkomma
    return (tmpfloat+0.5).floor.to_f/10**nachkomma if
      tmpfloat > 0.0
    return (tmpfloat-0.5).ceil.to_f/10**nachkomma if
      tmpfloat < 0.0
    return 0.0
  end
end
end

```

result.rb:

```

class Result
  attr_reader :analysis, :analyzer
  def initialize(analyzer,eval)
    #A hash instead of an array is used to store the data in
    #case of need to access a record by its key. Though it wasn't
    #necessary yet.
    @analysis = Hash::new
    @analyzer = analyzer
    @eval = eval
  end

  #Write all results into a file
  def write_all
    f = File::new(@eval.name + ".dat","w")
    results = self.sort_x_y("alpha","llength")
  end
end

```

```

    f.print results[0].keys
    results.each {|hauraus| f.print(hauraus.values)}
end

def sort_x_y(x,y)
  # '@analysis' is a hash with a hash holding the results:
  # discard the keys of the hash with 'analysis.values'.
  # This will return
  # a hash with elements that are the result and a
  # corresponding key
  # Then sort the array by the values assigned to alpha
  # And the linker length
  analysis = @analysis.values
  analysis.sort! {|a,b| a.values_at(x,y) <=> b.values_at(x,y)}
  return analysis
end

def write2d(const, x, y, y_error=0)
  # für jeweils const konstant
  konst = nil
  f = nil
  plotname2d = nil
  folder2d = @eval.name+"-2d"
  Dir.mkdir(folder2d) unless (FileTest.exists?(folder2d) &&
    FileTest.directory?(folder2d))
  Dir.chdir(folder2d)

  name2d = folder2d + "_" + x + "-" + y + "-" + const + "_const"
  Dir.mkdir(name2d) unless (FileTest.exists?(name2d) &&
    FileTest.directory?(name2d))
  Dir.chdir(name2d)

  plotter = Plotter.new(@eval)
  result_hash = self.sort_x_y(const,x)
  result_hash.each_with_index do |res,idx|
    # when value of res[const] changes then close file
    # and plot it (skip first time changing)
    # and create next file
    if konst!=res[const]

      unless idx==0
        # close (or at least flush) the connection so
        # gnuplot can read the data
        f.close unless f.closed? # means unless the first time or
          # h is closed already
        plotter.plot(f.path,plotname2d,x,y)
      end

      plotname2d = name2d + "-" + res[const].to_s
      filename2d = plotname2d + ".dat"
      f = File::new(filename2d,"w")
      # first adding a Header Line
      # (with # at the beginning so
      # gnuplot will ignore that line)
    end
  end
end

```

```

        f.write("#" + x + "\t" + "\t" + "\n")

        konst = res[const]
    end
    f.print(res.values_at(x,y,y_error)) unless res[y]==0
    #if that was the last result then don't go back to start
    #but close and plot file
    if idx == result_hash.length-1
        f.close unless f.closed? #means unless the first time
                                #or f is closed already
        plotter.plot(f.path,plotname2d, x,y)
    end
end
end
Dir.chdir("../..")
end

#Write Data into file for 3-d plotting
#(formatted to be read by gnuplot)
#and start plotting
def write3d(x, y, z)

    folder3d = @eval.name+"-3d"
    name3d = folder3d + "_" + x + "-" + y + "-" + z
    filename3d = name3d + ".dat"
    Dir.mkdir(folder3d) unless (FileTest.exists?(folder3d) &&
        FileTest.directory?(folder3d))
    Dir.chdir(folder3d)

    #first adding a Header Line
    #(with # at the beginning so gnuplot will ignore that line)
    g = File::new(filename3d,"w")
    g.write("#" + x + "\t" + y + "\t" + z + "\n")
    x_res = 0
    results = self.sort_x_y(x,y)

    #then writing the results into the file, printing a blank line
    #everytime alpha changes
    results.each do |res|
        g.print("") if x_res!=res[x] && x_res!=0
        g.print(res.values_at(x, y, z)) #unless res[z]==0
        x_res = res[x]
    end
    g.close

    plotter = Plotter.new(@eval)
    plotter.plot3d(filename3d,name3d, x, y, z)

    Dir.chdir("../..")
end

def read(traj_names)
    counter = 1
    traj_names.each do |trajName|
        result = Hash::new
    end
end

```

```

puts "processing file(s): " + trajName
Dir.chdir(trajName) unless @eval.allfilesinonedir == true

params = trajName.scan(/\d+\.\?\d*/).collect
        {|param| param.to_f }
result = {"alpha" => params[0], "llength" => params[1]}

@analyzer.traj_name = trajName

result.merge!(@analyzer.get_length_diameter)
result.merge!(@analyzer.get_nucleosome_tilt)
result.merge!(@analyzer.getPersistenceLength)
result.merge!(@analyzer.get_alphaeff)
result.merge!(@analyzer.get_energy)

Dir.chdir('..') unless @eval.allfilesinonedir == true

counter += 1

#Append results to the hash
@analysis[trajName] = result

#empty the hash
result = nil
end
end
end

```

plotter.rb:

```

class Plotter

  def initialize(eval)
    @gnuplot_version = eval.gnuplot_version
  end

  def plot(datafile, title, xlabel, ylabel)
    outputfile_eps = title + ".eps"
    outputfile_png = title + ".png"
    outputfile_gnuplot = title + ".gnuplot"
    gnuplot = IO.popen("gnuplot -persist", "w+")

    if @gnuplot_version == 3.7
      plotcommand = <<-END_OF_COMMAND
      set title '#{50}'
      set xlabel '#{xlabel}'
      set ylabel '#{ylabel}'
      set mxtics 2
      set mytics 2
      set terminal png
      set output '#{outputfile_png}'
      plot '#{datafile}' notitle with errorbars, '#{datafile}'
        title '' with lines
    end
  end
end

```

```
    set terminal postscript eps enhanced color
    set output '#{outputfile_eps}'
    replot
    set terminal x11
    replot
    END_OF_COMMAND
else
    plotcommand =<<-END_OF_COMMAND
    set style data lines
    set title '#[50]'
    set xlabel '#{xlabel}'
    set ylabel '#{ylabel}'
    set mxtics 2
    set mytics 2
    set terminal png
    set output '#{outputfile_png}'
    plot '#{datafile}' with errorbars , '#{datafile}' notitle
        with lines
    set terminal postscript eps enhanced color
    set output '#{outputfile_eps}'
    replot
    set terminal x11
    replot
    END_OF_COMMAND
end

g = File::new(outputfile_gnuplot,"w")
g.write(plotcommand)
g.close

gnuplot.puts plotcommand

end

def plot3d(datafile, title, xlabel, ylabel, zlabel)

    #wd = Dir.getwd
    #datafile = wd + "/" + datafile
    outputfile_eps = title + ".eps"
    outputfile_png = title + ".png"
    outputfile_gnuplot = title + ".gnuplot"
    gnuplot = IO.popen("gnuplot -persist","w+")

    if @gnuplot_version == 3.7
        plotcommand =<<-END_OF_COMMAND
        set hidden3d
        set contour
        set grid
        set view 60,220,1
        set title '#[50]'
        set ylabel '#{ylabel}'
        set xlabel '#{xlabel}'
        set zlabel '#{zlabel}'
        set mxtics 2
```

```

set mytics 2
set terminal x11
splot '#{datafile}' notitle with lines
set terminal png
set output '#{outputfile_png}'
replot
set encoding iso_8859_1
set terminal postscript eps enhanced color
set xlabel
set ylabel
set zlabel
set label 1 '{/Arial entry-exit angle}&{s}{/Symbol a
[\260]}' at screen 0.2,0.17 font 'Arial'
rotate by -15
set label 2 'linker DNA length [nm]' at screen 0.66,0.11
font 'Arial' rotate by 20
set label 3 '#{zlabel}' font 'Arial' at screen 0.05,0.4
rotate by 90
show label
set output '#{outputfile_eps}'
replot
END_OF_COMMAND
else
plotcommand =<<-END_OF_COMMAND
set style data lines
set hidden3d
set contour
set grid
set view 60,220,1
set title '#{50}'
set xlabel '#{xlabel}'
set ylabel '#{ylabel}'
set zlabel '#{zlabel}'
set mxtics 2
set mytics 2
#set pm3d
set terminal x11
splot '#{datafile}' notitle
set terminal png
set output '#{outputfile_png}'
replot
set encoding iso_8859_1
set terminal postscript eps enhanced color
set xlabel
set ylabel
set zlabel
set label 1 '{/Arial entry-exit angle}&{s}{/Symbol a
[\260]}' at screen 0.2,0.17 font 'Arial'
rotate by -15
set label 2 'linker DNA length [nm]' at screen 0.66,0.11
font 'Arial' rotate by 20
set label 3 '#{zlabel}' font 'Arial' at screen 0.05,0.4
rotate by 90
show label

```

```
        set output '#{outputfile_eps}'
        replot
        END_OF_COMMAND
    end

    g = File::new(outputfile_gnuplot, "w")
    g.write(plotcommand)
    g.close

    gnuplot.puts plotcommand
end
end
```

Erklärung:

Ich versichere, dass ich diese Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den

Unterschrift