Freedom of additional signals on genes: on the combination of DNA mechanics, genetics and translation speed

Zuiddam, M.

Appendix A

The physics behind the mechanical nucleosome positioning code

A.1 Energy contributions of twist and cross terms

In section 2.2 we define the energy of a dinucleotide as the sum of the energy of roll and tilt. Keeping in mind the observation that the basic nucleosome positioning rules can be rationalized by discussing energy costs involved in the roll and tilt degrees of freedom [14], as well as our goal to reduce our model to its bare essentials, we chose to neglect the contribution of twist and of the cross terms between roll, tilt, and twist. In this appendix we will show that including the twist and cross terms does not change the qualitative agreement of our model with well-known positioning rules (see Fig. 2.3).

We start by defining the energy of twist and the cross-terms:

$$E^{\text{twist}}(a,b) \equiv \frac{1}{2} Q^{\text{twist}}(a,b) \left[q^{\text{twist}} - \bar{q}^{\text{twist}}(a,b) \right]^2 , \tag{A.1}$$

and

$$E_p^{\text{cross}}(a,b) \equiv \sum_{\substack{i,j \in \{\text{roll,tilt,twist}\} \\ i \neq j}} \frac{1}{2} Q^{i,j}(a,b) \left[q_p^i - \bar{q}^i(a,b)_p \right] \left[q_p^j - \bar{q}^j(a,b)_p \right] . \tag{A.2}$$

The bp-step dependent stiffnesses are now given by $Q^i(a,b)$, $i \in \{\text{roll}, \text{tilt}, \text{twist}\}$ and the corresponding intrinsic values by $\bar{q}^i(a,b)$, $i \in \{\text{roll}, \text{tilt}, \text{twist}\}$. The cross terms depend on the cross stiffnesses $Q^{i,j}(a,b)$, $i,j \in \{\text{roll}, \text{tilt}, \text{twist}\}$, $i \neq j$. (Note that, because of the constant twist, the energy associated with twist does not depend on position p but only on the dinucleotide step.) For the twist and cross terms, too, the hybrid parametrization [38] is used. We can redefine our energy as

$$E_p(a,b) = E_p^{\text{roll}}(a,b) + E_p^{\text{tilt}}(a,b) + E^{\text{twist}}(a,b) + E_p^{\text{cross}}(a,b). \tag{A.3}$$

Fig. A.1 was created using this redefined energy. We see that the relative behaviour of the dinucleotide probabilities at different positions is the same as without the cross terms, see Fig. 2.3.

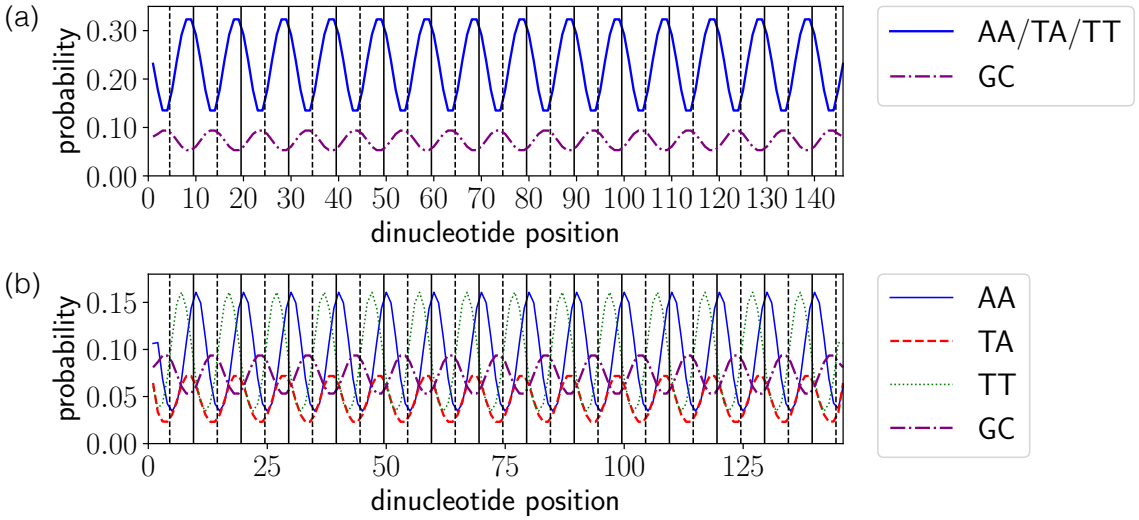


Figure A.1: Same as Fig. 2.3 but with including the cross-terms (see Eq. A.3). The positioning rules (i.e. the relative behaviour of the dinucleotide probabilities at different position) has stayed the same.

A.2 Validity of the average neighbour energy approximation

The average neighbour energy approximation of the probability works extremely well. We checked it for all dinucleotides and found that the largest error of this approximation occurs for the probability distribution of dinucleotide AA. Fig. A.2 depicts both the exact probability and its approximation for this dinucleotide. The difference between the values is always smaller than 3.5%.

To understand why this error is so small, one needs to consider the function $C_p(x, y)$, defined in Eq. 2.28. The average neighbour energy approximation is exact if this function is a constant (i.e., independent of x and y for each p). The approximation works well if the function is almost constant. That this is true is best seen by inspecting the standard deviation of $C_p(x, y)$, divided by its mean, and checking whether this quantity is much smaller than one. Here the standard deviation and mean are defined as:

$$\mathrm{std}[C_p] \equiv \sqrt{\langle \{C_p(x,y) - \mathrm{mean}[C_p]\}^2 \rangle_{x,y}} \tag{A.4}$$

with

$$\mathrm{mean}[C_p] \equiv \langle C_p(x,y) \rangle_{x,y}. \tag{A.5}$$

Fig. A.3 shows that this ratio is indeed much smaller than one for all dinucleotide positions.

A.3 Effect of temperature on the probability

The probabilities shown in the results section have all been obtained at room temperature $\beta = 1/k_B T_r$. Here we study how these probabilities change with temperature,

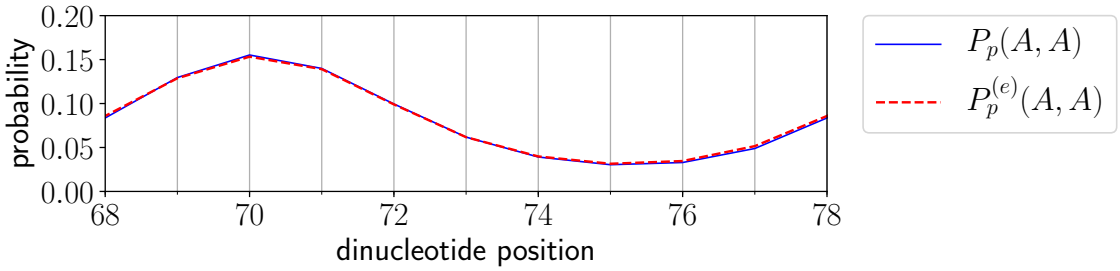


Figure A.2: The exact probability and its average neighbour energy approximation to find AA steps at all dinucleotide positions. The approximation introduces an error that is nowhere larger than 3.5%.

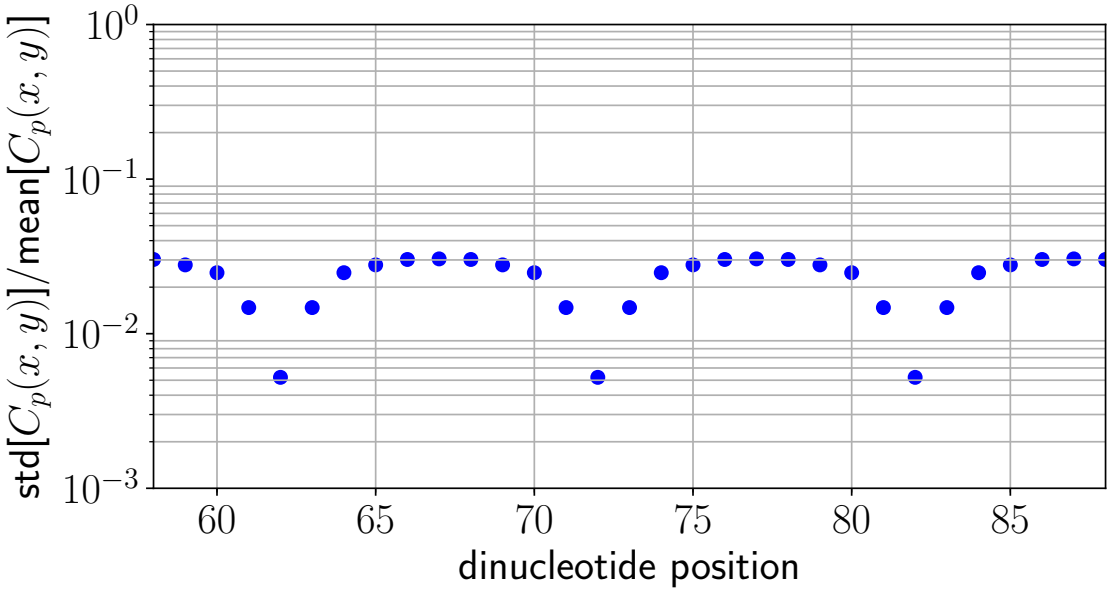


Figure A.3: The standard deviation (Eq. A.4) divided by the mean (Eq. A.5) of $C_p(x,y)$. As this ratio is very small at all positions p the function $C_p(x,y)$ is nearly constant, explaining the high accuracy of the average neighbour energy approximation.

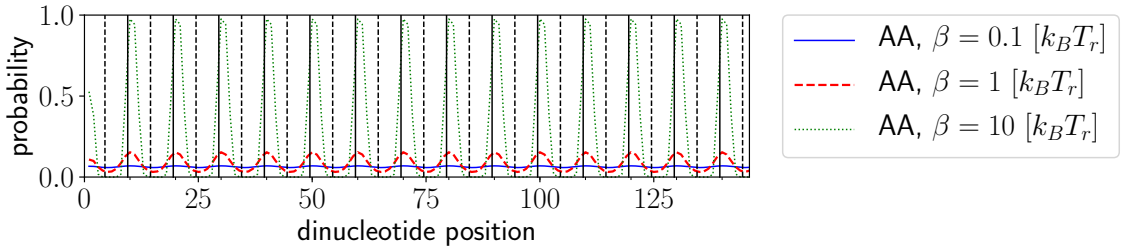


Figure A.4: The probability to obtain AA at all dinucleotide positions at several temperatures.

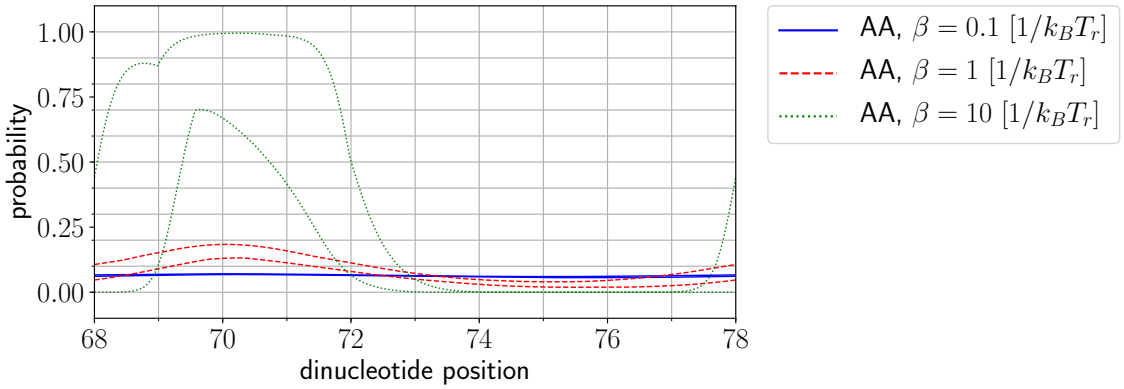


Figure A.5: The first-order bounds of the probability of encountering an AA step are shown at five different temperatures. At low temperatures, the bounds become significantly far apart from each other and only provide a qualitative description of the behaviour of the probability as a function of position.

focusing on dinucleotide AA. Its exact probability distribution for different temperatures is shown in Fig. A.4. We find at temperature $\beta = 0$ a constant value $1/16$ for the probability. This is the high temperature limit where all steps are equally probable. At low temperatures the probability varies between values close to 0 and 1, reflecting the fact that the ground state sequences becomes exceedingly important.

We also evaluated the first- and second-order bounds of the AA probability distribution at five different temperatures: $\beta = 0$, 0.1, 1, 10, and 100 (in units of $[1/k_B T_r]$), see Fig. A.5, and Fig. A.6. At high temperatures (low β) the bounds for both orders are very close to each other enclosing values close to $1/16$. With decreasing temperature the quality of the first-order bounds becomes poorer, giving only a rough qualitative estimate whereas the second-order bounds continue to work well for relatively low temperatures. Note that at $\beta = 100$ the probability takes values close to 0 and 1 at most places.

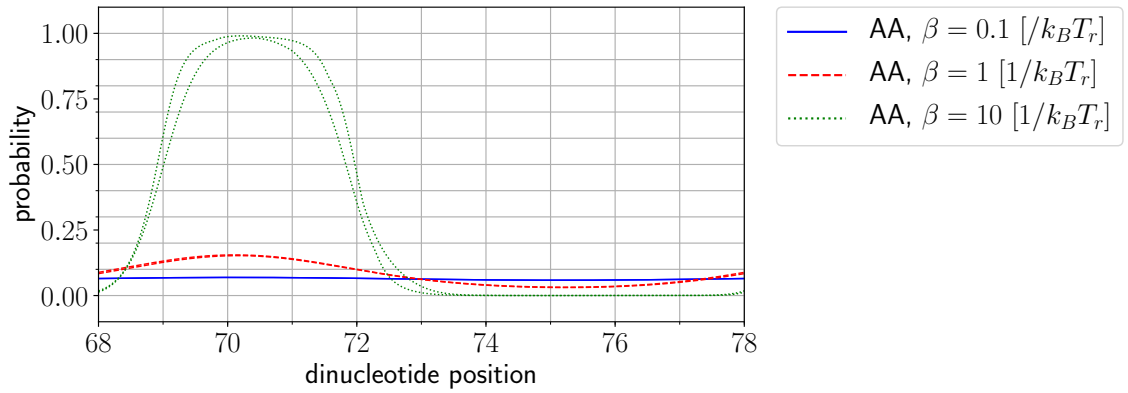


Figure A.6: The second-order bounds of the probability of encountering an AA step are shown at five different temperatures. At all temperatures the method provides a quantitative description of the probability, clearly outperforming the second-order bounds.

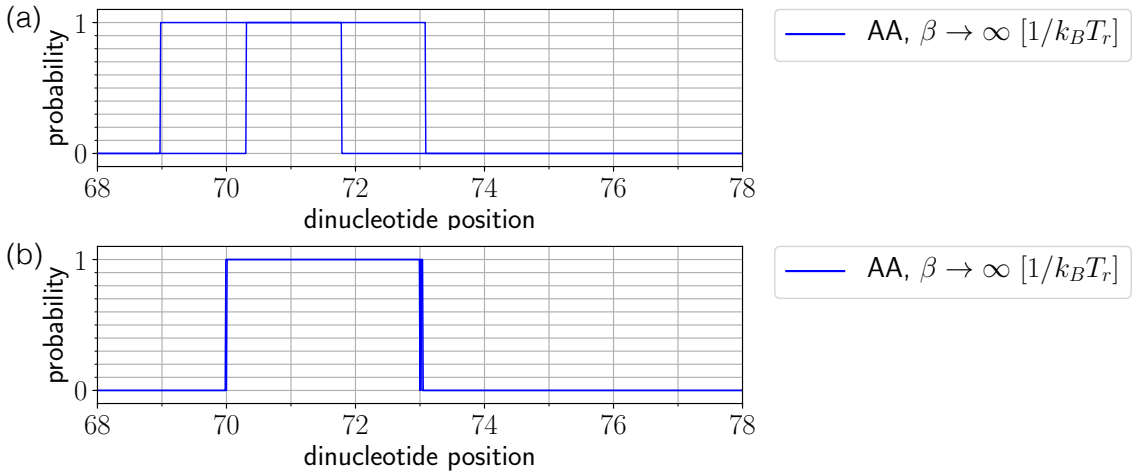


Figure A.7: (a) First-order and (b) second-order bounds on the probability to find dinucleotide AA at several dinucleotide positions on a nucleosome, in the limit of zero temperature. Higher-order bounds get increasingly sharper. At zero temperature the only possible DNA sequences are ground state sequences, hence the bounds provide us statistics on the ground states of our system. When the lower and upper bounds are 0 and 1, AA is part of an unknown number of ground states at this position. If AA is part of all possible ground states the bounds are 1 and 1, and if it is not a part of the ground state they are 0 and 0.

We finally take the limit $\beta \to \infty$, see Fig. A.7. This figure shows the second-order and third-order bounds on the probability to encounter AA at zero temperature. The only possible sequences are now ground state sequences (due to the high level of symmetry in our model we expect many different ground states). This explains why the probability of AA can take the values 0 and 1: at several positions AA is not part of any ground state sequence (probability is zero), while at other positions AA is part of all possible ground state sequences (probability is 1). At some positions the method cannot determine the percentage of ground state sequences AA is part of, resulting in bounds of 0 *and* 1.

The method of obtaining upper and lower bounds remains effective at all possible temperatures for our model, and even provides insight into the possible ground states. Going to higher-order bounds (i.e., taking neighbours that are further away into account as well) or using the exact probability should eliminate the discrepancy between the upper and lower value. However, the method employed in Chapter 3 (which uses a graph representation of all possible sequences in combination with a shortest path algorithm) is much more efficient in obtaining ground states.

Appendix B

Shortest paths through synonymous codons

B.1 Definition of the energy

In Chapter 3, we aim to find sequences with 'special' energies, e.g. the sequences with the lowest and highest possible energies. To calculate the energy of a sequence, we use the probabilistic trinucleotide model by Tompitak et al. [10] which is based on the sequence preferences of a coarse grained nucleosome model, parametrized by experimental parameters derived from protein-DNA crystals [32]. Because it is a trinucleotide model, we are able to represent the total energy of a sequence as a sum of 'conditional' trinucleotide energies, which function as the (main ingredients of the) weights in our graphs. Here we will formally define these energies.

Let $\mathcal{B}$ be the set of all nucleotides, $\mathcal{B} = \{A, T, C, G\}$. For the trinucleotide model, it is assumed that the probability of a nucleotide depends only on the previous two. Defining S as a sequence of length L, consisting of nucleotides $S_i \in \mathcal{B}$ with i from 1 to 147, this gives a probability for the full sequence:

$$P(S) = \frac{\prod_{n=1}^{L-2} P_n(S_{n+2} \cap S_{n+1} \cap S_n)}{\prod_{n=1}^{L-3} P_n(S_{n+2} \cap S_{n+1})} \tag{B.1}$$

where $P_n(S_{n+2} \cap S_{n+1} \cap S_n)$ is the joint (trinucleotide) probability to obtain S_{n+2}, S_{n+1}, and S_n at position n, and $P(S_{n+2} \cap S_{n+1})$ the joint (dinucleotide) probability to obtain S_{n+2}, S_{n+1} at position n. However, since the original trinucleotide model by Tompitak et al. does not enforce the symmetry of the coding and noncoding strand, we introduce symmetrized probabilities:

$$\begin{aligned} P'_n(S_n \cap S_{n-1} \cap S_{n-2}) = &\frac{1}{2}\left[P_n(S_n \cap S_{n-1} \cap S_{n-2})\right] \\ &+ \frac{1}{2}\left[P_n(S'_{n-2} \cap S'_{n-1} \cap S'_n)\right] \end{aligned} \tag{B.2}$$

and

$$P'_n(S_n \cap S_{n-1}) = \frac{1}{2}\left[P_n(S_n \cap S_{n-1}) + P_n(S'_{n-1} \cap S'_n)\right] \tag{B.3}$$

where

$$S'_n \equiv \begin{cases} A_{148-n} \text{ if } S_n = T \\ T_{148-n} \text{ if } S_n = A \\ C_{148-n} \text{ if } S_n = G \\ G_{148-n} \text{ if } S_n = C \end{cases} \tag{B.4}$$

such that

$$P'_n(S) = \frac{\prod_{n=1}^{L-2} P'_n(S_{n+2} \cap S_{n+1} \cap S_n)}{\prod_{n=1}^{L-3} P'_n(S_{n+2} \cap S_{n+1})}. \tag{B.5}$$

Following Tompitak et al., we use the probability to calculate a free energy, using $E(S) = -k_B T_r \ln[P(S)] + const.$ We rewrite the energy as:

$$E(S) = \sum_{n=1}^{L-2} E_n(S_{n+2}, S_{n+1}, S_n) + const. \tag{B.6}$$

where

$$E_n(S_n, S_{n+1}, S_{n+2}) = \begin{cases} -k_B T_r \ln[P'(S_{n+2} \cap S_{n+1} \cap S_n)] \text{ if } n = 1 \\ -k_B T_r \ln\left[\frac{P'(S_{n+2} \cap S_{n+1} \cap S_n)}{P'(S_{n+1} \cap S_n)}\right] \text{ if } 1 < n < 146 \\ 0 \text{ else.} \end{cases} \tag{B.7}$$

We define *const.* such that the energy E is zero if S is the ground state.

For $n = 1$, E_n is the energy cost related to the first three bases of a sequence S, for $1 < n < 146$, it is a 'conditional' energy, and it is zero elsewhere. We use these terms as weights of our graph, while keeping in mind that the sum of these weights will provide the well-defined total energy E.

B.2 Definition of the depth of a minimum

In the main text of Chapter 3, we use the depth of a minimum $\mathcal{D}$ as a measure for how well the nucleosome is positioned at this minimum. Here we will formally define $\mathcal{D}$.

Let $\mathcal{S}$ be some sequence of length greater than $L + 10$ (with $L = 147$). Let S^p be a subsequence of $\mathcal{S}$ of length L starting at position p.

We call a nucleosome positioned at p if the energy $E(S^p)$ is lower than the energies at positions $p - 5, p - 4, ..., p + 5$ (excluding p). We denote the energy corresponding to a nucleosome containing the sequence S^p by $\mathscr{E}_p \equiv E(S^p)$. For a minimum at $p_{\min}$ of sequence S we are interested in its depth, $\mathcal{D}(S^{p_{\min}})$. Now we can formally define the depth as

$$\mathcal{D}(S^{p_{\min}}) \equiv \min\left[\mathscr{E}^{\max}_{\text{left}}(S^{p_{\min}}), \mathscr{E}^{\max}_{\text{right}}(S^{p_{\min}})\right] \tag{B.8}$$

where

$$\mathscr{E}^{\max}_{\text{left}}(S^{p_{\min}}) \equiv \max\left[\mathscr{E}_{p_{\min}-i}(S) \text{ for } i \in \{1, 2, ..., 5\}\right] - \mathscr{E}_{p_{\min}}(S), \tag{B.9}$$

$$\mathscr{E}^{\max}_{\text{right}}(S^{p_{\min}}) \equiv \max\left[\mathscr{E}_{p_{\min}+i}(S) \text{ for } i \in \{1, 2, .., 5\}\right] - \mathscr{E}_{p_{\min}}(S). \tag{B.10}$$

B.3 The deepest possible minimum

Here we show how to obtain the deepest possible minimum, with only a tiny possible error, by taking the shortest paths through the graphs $\mathcal{G}^+_{h,j}$ defined in the main text.

A nucleosome is best positioned at a minimum $p_{\min}$ if $\mathcal{D}(S^{p_{\min}})$ is maximal. We assume that the deepest possible minimum $\mathcal{D}(S_{\text{deepest}})$ is found for a sequence S_{deepest}. Furthermore, we assume that

$$\mathscr{E}^{\max}_{\text{left}}(S_{\text{deepest}}) = \mathscr{E}_{p_{\min}+h}(S_{\text{deepest}}) - \mathscr{E}_{p_{\min}}(S_{\text{deepest}}) \tag{B.11}$$

and

$$\mathscr{E}^{\max}_{\text{right}}(S_{\text{deepest}}) = \mathscr{E}_{p_{\min}+j}(S_{\text{deepest}}) - \mathscr{E}_{p_{\min}}(S_{\text{deepest}}) \tag{B.12}$$

for $h \in \{-5,-4,..,-1\}$, $j \in \{1,2...,5\}$.

Let us denote the shortest path through $\mathcal{G}^+_{h,j}$ by $S_{h,j}$ with the minimum at $p_{\min}$. A shortest path through $\mathcal{G}^+_{h,j}$ will minimize the quantity $2\mathscr{E}_{p_{\min}} - \mathscr{E}_{p_{\min}+h} - \mathscr{E}_{p_{\min}+j}$. Because of this, we have

$$\mathscr{E}^{\max}_{\text{left}}(S^{p_{\min}}_{h,j}) + \mathscr{E}^{\max}_{\text{right}}(S^{p_{\min}}_{h,j}) \geq \mathscr{E}^{\max}_{\text{left}}(S^{p_{\min}}_{\text{deepest}}) + \mathscr{E}^{\max}_{\text{right}}(S^{p_{\min}}_{\text{deepest}}). \tag{B.13}$$

Since $S^{p_{\min}}_{\text{deepest}}$ is the sequence with the greatest depth, we have

$$\begin{aligned} &\min\left[\mathscr{E}^{\max}_{\text{left}}(S^{p_{\min}}_{h,j}), \mathscr{E}^{\max}_{\text{right}}(S^{p_{\min}}_{h,j})\right] \leq \\ &\quad \min\left[\mathscr{E}^{\max}_{\text{left}}(S^{p_{\min}}_{\text{deepest}}), \mathscr{E}^{\max}_{\text{right}}(S^{p_{\min}}_{\text{deepest}})\right]. \end{aligned} \tag{B.14}$$

Combining Eq. B.13 and B.14 leads to bounds on the depth of the deepest possible minimum:

$$\begin{aligned} &\min\left[\mathscr{E}^{\max}_{\text{left}}(S^{p_{\min}}_{h,j}), \mathscr{E}^{\max}_{\text{right}}(S^{p_{\min}}_{h,j})\right] \leq \\ &\quad \mathscr{E}^{\max}_{\text{left}}(S^{p_{\min}}_{\text{deepest}}) && \text{(B.15)} \\ &\quad \leq \frac{1}{2}\left[\mathscr{E}^{\max}_{\text{left}}(S^{p_{\min}}_{h,j}) + \mathscr{E}^{\max}_{\text{right}}(S^{p_{\min}}_{h,j})\right]. && \text{(B.16)} \end{aligned}$$

We took the shortest path through all graphs $\mathcal{G}^+_{h,j}$ for all $h \in \{-5,-4,..,-1\}$, $j \in \{1,2...,5\}$. Of all the graphs, $\mathcal{G}^+_{-5,5}$ provided the deepest minimum. Using the above equation, we obtained $83.47 \pm 0.03\ k_B T_r$ as the deepest possible minimum.

B.4 Graphs

We have defined the graphs $\mathcal{G}^+_{h,j}$, extensions of $\mathcal{G}$ with differently assigned weights, for $h,j \in \{-5,-4,..,-1,1,2...,5\}$. A visual depiction is shown by Fig. B.1. The graph $\mathcal{G}^+_{\text{gene}}$, an extended version of $\mathcal{G}_{\text{gene}}$, is depicted by Fig. B.2.

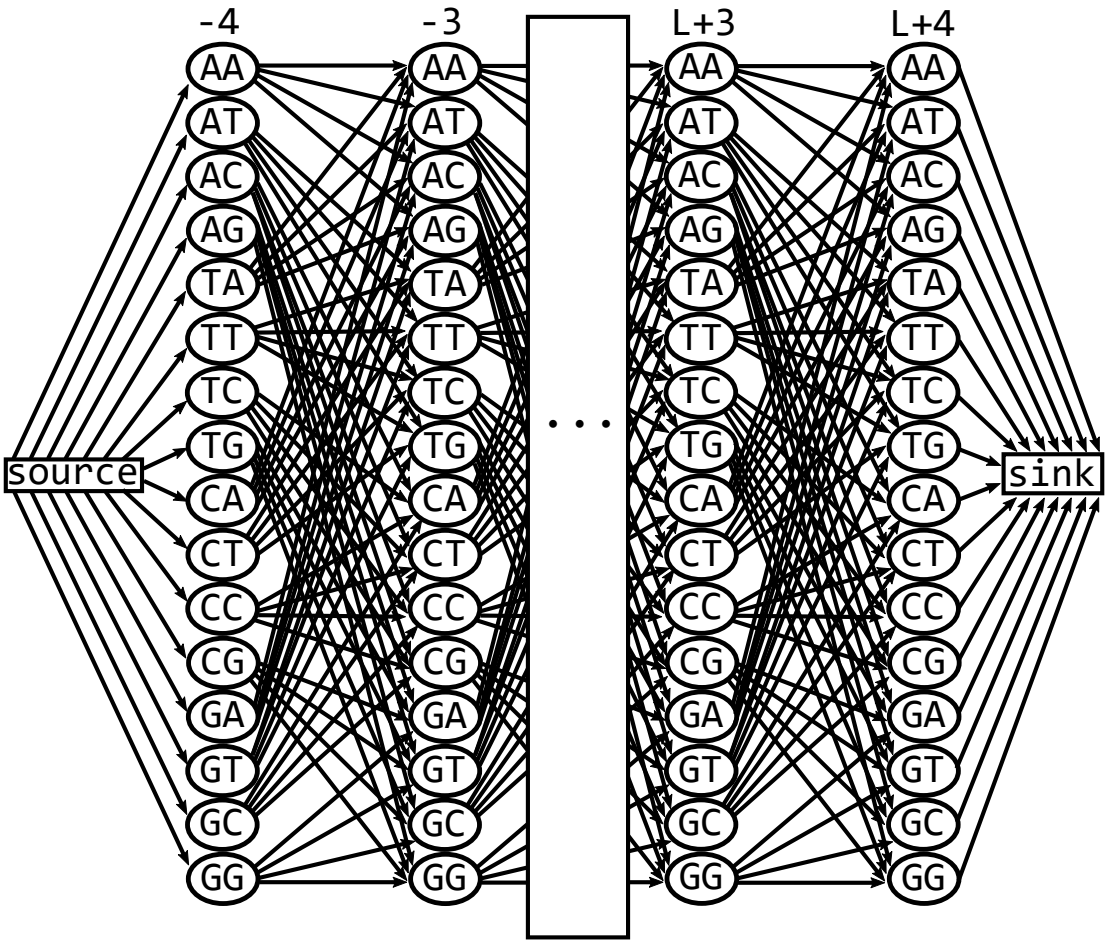


Figure B.1: Visualisation of a graph $\mathcal{G}^{+}_{h,j}$

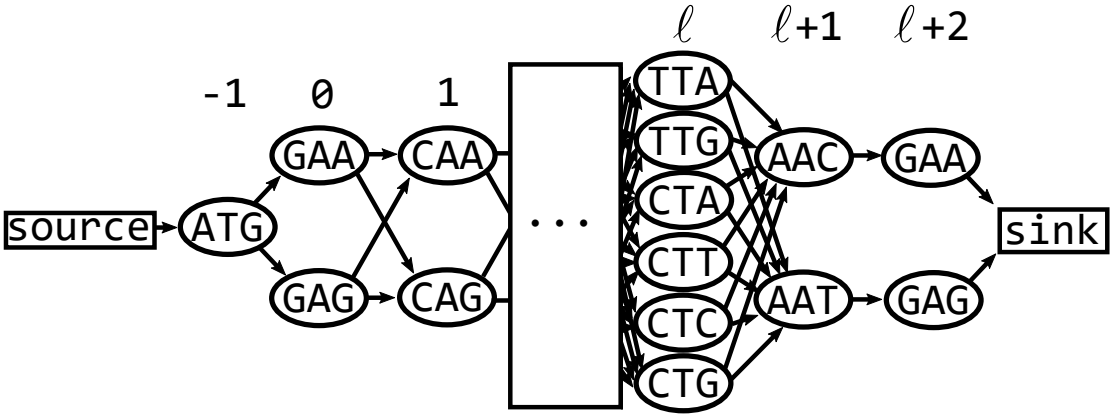


Figure B.2: Visualisation of a graph $\mathcal{G}_{\text{gene}}$. This graph corresponds to creating a minimum at the 7^{th} nucleosome position on the gene YAL002W of yeast.

B.5 Create local minima on top of genes

To create local minima at a position on a gene, we came up with a specifically tailored method where we alter the values of the constants c_i with each iteration.

iteration	**starting conditions**	**action**
all iterations	$c_0 = 1$ $c_i = 0$ for $i \notin \{-5, 0, 5\}$	minimum and depth check: $\mathcal{D} \geq 10\ k_B T_r$
1-20	$c_{-5} = c_5 = -0.3$	regular decrement
21-40	$c_{-5} = c_5 = -0.3$	neighbor decrement
41-60	$c_{-5} = c_5 = -0.2$	regular decrement
61-80	$c_{-5} = c_5 = -0.2$	neighbor decrement
81-100	$c_{-5} = c_5 = -0.1$	regular decrement
101-120	$c_{-5} = c_5 = -0.1$	neighbor decrement
121-140	$c_{-5} = c_5 = 0$	regular decrement
141-160	$c_{-5} = c_5 = 0$	regular decrement
if all fail	-	take best solution

Table B.2: Schematic form of specifically tailored method to create deep local minima at a position on a gene. The method works by altering the weights w_i' of graph $\mathcal{G}^{\text{gene}}$ by changing the constants c_i, see Eq. 4 of the main text.

This will result in a differently weighted graph each iteration and different shortest paths. The algorithm uses at most 160 iterations per position. The iterations are grouped in eight parts, with differing starting conditions and different increment rules. See Table B.2 for an overview of this method.

All iterations start with $c_0 = 1$, $c_i = 0$ for $i \notin \{-5, 0, 5\}$. Iterations 1-20 start with $c_{-5} = c_5 = -0.3$. At the start of iteration 21-40, all constants are reset and we again begin with $c_{-5} = c_5 = -0.3$. Iterations 41-60 and 61-80 have $c_{-5} = c_5 = -0.2$, 81-100 and 100-120 have $c_{-5} = c_5 = -0.1$, and 121-140 and 141-160 have $c_{-5} = c_5 = 0$. The different starting conditions are intended to first try to create deep minima through a larger incentive to have high walls, but if this fails, settle for lower minima.

At the beginning of each and every iteration a check is performed. The energy landscape corresponding to the shortest path is evaluated to find whether a local minimum has been created at the right position. If there is such a local minimum, we evaluate how deep it is. If it is deeper than $10\ k_B T_r$, we accept the corresponding sequence. If the local minimum is not deep enough, we evaluate which side of the energy well has the lowest wall. If the left or right wall is lowest, we set $c_{-5} \to c_{-5} - 0.1$ or $c_5 \to c_5 - 0.1$, respectively, and move to the next iteration. If there is no local minimum, we perform one of the two distinct schemes: 'regular decrement' and 'neighbor decrement', introduced below. We perform a 'regular decrement' at iterations 1-20, 41-60, etc., and a 'neighbor decrement' at all other iterations.

The regular decrement is defined as follows: if the position with the lowest energy is $p_{\min} + i$ instead of the intended position $p_{\min}$, we perform $c_i \to c_i - 0.1$. Differently stated, we give our algorithm an incentive to raise the energy at positions where the energy is lower than at $p_{\min}$. The main problem of the regular decrement is that the lowest energy position often alternates between $p_{\min} + 1$ and $p_{\min} - 1$. Making the decrements smaller turned out to be ineffective in solving this problem, so instead we define the 'neighbour decrement'.

The neighbour decrement is the same as the regular decrement, with one dif-

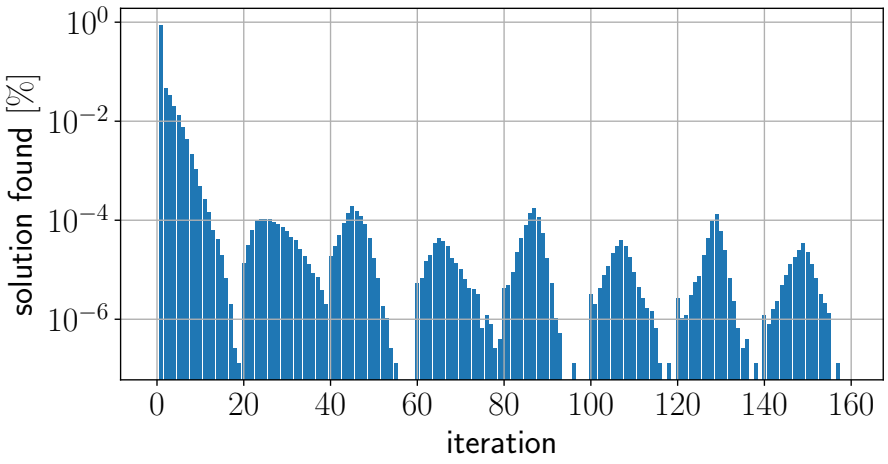


Figure B.3: For each possible iteration, the percentage of positions solved (i.e. with a deep enough minimum found) is depicted. All positions on genes from yeast *S. cerevisiae* (ignoring genes with introns) where evaluated. The bulk of the positions were completed at the first iteration.

ference: if the position with the lowest energy is $p_{\min} \pm 1$ instead of the intended position $p_{\min}$, we perform $c_{i\pm2} \to c_{i\pm2} - 0.1$.

It is possible that, after 160 iterations, no deep enough minimum is found. Then we take the deepest minimum we encountered (if any exists) as our result. The percentage of positions resolved at which iteration is depicted by Fig. B.3. It shows that the bulk of the positions were completed at the first iteration.

Appendix C

Multiplexing mechanical and translational cues on genes

C.1 Graph to obtain highest and lowest possible nucleosome energy

In Chapter 4 we use a graph representation of all possible sequences that code for the same protein. To understand the new method we use in this chapter, we first shortly summarize the method we used in Chapter 3[1], where we were able to obtain the highest and lowest possible nucleosome energies on all positions of a gene. To obtain these energies we use a graph containing all synonymous codons of the gene section corresponding to one nucleosome position.

The DNA on a nucleosome consists of 147 base pairs, which corresponds to either 49 or 50 codons. Suppose we have a sequence of 50 codons. These codons encode a sequence of amino acids $p_0, p_1, p_2, ..., p_{49}$. The number of different codons coding for the same amino acid is 6 at most. Therefore, the most general representation of all possible ways to code for the same protein at one nucleosome position is given by figure C.1 (we use the most general representation to make it easier to understand the graphs related to three layers of information).

In this figure, under each amino acid p_n, six numbers are shown representing the (at most) six possible codons, which we will refer to in the following as $p_n(1), p_n(2), ..., p_n(6)$. The actual base pairs of the codons depend on the amino acid in question. To obtain this graph we draw the following weighted edges: from start to $p_0(i)$ with weight zero for any i, from $p_{49}(i)$ to end with weight $w_{\text{end}}(p_{49}(i))$ for any i, and from $p_n(i)$ to $p_{n+1}(j)$ with weight $w_n(p_n(i), p_{n+1}(j))$ for any i, j and $n = 0, 1, ..., 48$. The weight w_i is given by

$$w_i(C, D) = E_{3i-2}(C_1, C_2, C_3) + E_{3i-1}(C_2, C_3, D_1) + E_{3i}(C_3, D_1, D_2) \tag{C.1}$$

[1]There are two main advantages to summarizing this method again, as opposed to simply referring to the previous chapter/appendix. It makes Chapter 4, in combination with its appendix, readable (and hopefully comprehensable) as a single unit. Secondly: the notation we use here is quite different, such that we can more easily incorporate translation speed in the graph (see appendix C.2).

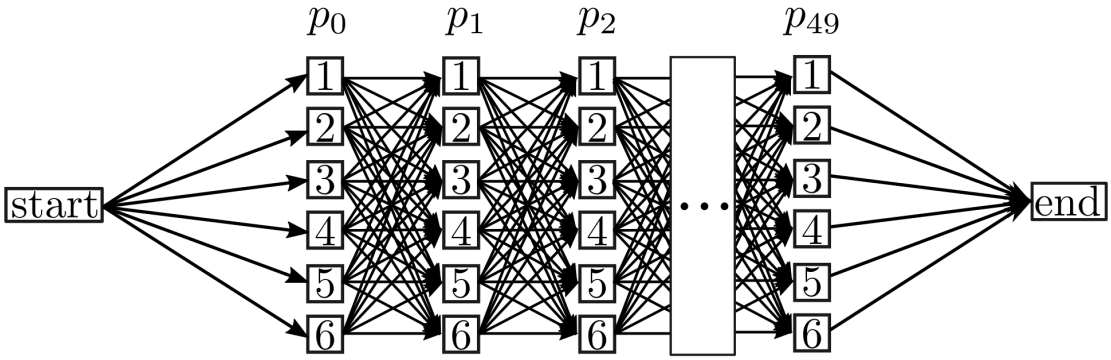


Figure C.1: Graph $\mathcal{G}_{\mathrm{E}}$ shows all synonymous ways to encode a given amino acid sequence $p_0, p_1, ..., p_{49}$ in the most general case. For each amino acid six options are shown, representing the at most six possible ways to code for the same amino acid. The actual bases depend on the amino acid in question. When there are less than six options, one can simply leave out the surplus of nodes. Weights are assigned such that each path from *start* to *end* has a length equal to the total energy of the corresponding codon sequence.

and the weight w_{end} by

$$w_{\mathrm{end}}(D) = E_{145}(D_1, D_2, D_3) \tag{C.2}$$

where C_k and D_k denote the kth base of codons C and D. Now the length of a path from *start* to *end* in the graph equals the energy of a corresponding sequence. The lowest and highest energy can be found using a shortest path algorithm.

C.2 Obtaining the highest and lowest possible nucleosome energy, incorporating translation speed

Here we describe the method used to obtain the highest and lowest possible nucleosome energy, incorporating a restriction on the translation speed. The method uses a graph $\mathcal{G}_{\mathrm{T\&E}}$, which is similar to graph $\mathcal{G}_{\mathrm{E}}$. Since we study in chapter 4 five-codon averages of the translation speed, $\mathcal{G}_{\mathrm{T\&E}}$ incorporates translation speed by using nodes consisting of five codons, see figure C.2. These nodes are connected such that any node

$$p_n(x_n)p_{n+1}(x_{n+1})p_{n+2}(x_{n+2})p_{n+3}(x_{n+3})p_{n+4}(x_{n+4})$$

can only be connected to nodes

$$p_{n+1}(x_{n+1})p_{n+2}(x_{n+2})p_{n+3}(x_{n+3})p_{n+4}(x_{n+4})p_{n+5}(x_{n+5})$$

for any $x_i \in \{1, 2, ..., 6\}$, with weight $w_{n+4}(p_{n+4}(x_{n+4}), p_{n+5}(x_{n+5}))$. All other edges have zero weight. Now, to ensure that one does not alter the translation speed landscape too much when changing the nucleosome energy, one can, for each node, calculate the difference between the translation speed of that node and the original speed. When the difference exceeds a certain threshold, the node needs to be pruned, such that each path through the graph corresponds to a sequence that does not change the underlying amino acid sequence and the translation speed landscape remains the same up to the threshold. Again, the lowest and highest energy can be found using a shortest path algorithm.

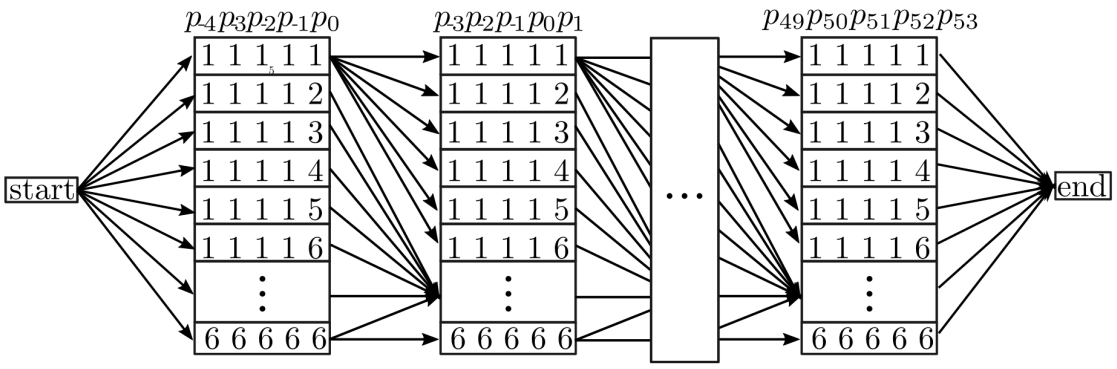


Figure C.2: Graph $\mathcal{G}_{\mathrm{T\&E}}$ is similar to graph $\mathcal{G}_{\mathrm{E}}$ from figure C.1. It incorporates translation speed by using nodes consisting of five codons. These nodes are connected such that any node $p_n(x_n)p_{n+1}(x_{n+1})p_{n+2}(x_{n+2})p_{n+3}(x_{n+3})p_{n+4}(x_{n+4})$ can only be connected to nodes $p_{n+1}(x_{n+1})p_{n+2}(x_{n+2})p_{n+3}(x_{n+3})p_{n+4}(x_{n+4})p_{n+5}(x_{n+5})$ for any $x_i \in \{1, 2, ..., 6\}$. When the translation speed of five codons (a node) is too different from the original speed, it is pruned. The weights of the graph are again chosen such that any path length corresponds to the nucleosome energy of the corresponding sequence.

C.3 Recovering the original nucleosome energy and translation speed landscapes in host organisms

To create the closest possible translation speed landscape in a different organism, we modify graph $\mathcal{G}_{\mathrm{T\&E}}$ to become gene-wide and obtain graph $\mathcal{G}_{\mathrm{gene}}$ see figure C.3.

We also change the weights. We denote the weights corresponding to the closest possible translation speed landscape by w^{T}. Again these nodes are connected such that any node $p_n(x_n)p_{n+1}(x_{n+1})p_{n+2}(x_{n+2})p_{n+3}(x_{n+3})p_{n+4}(x_{n+4})$ can only be connected to nodes $p_{n+1}(x_{n+1})p_{n+2}(x_{n+2})p_{n+3}(x_{n+3})p_{n+4}(x_{n+4})p_{n+5}(x_{n+5})$ for any $x_i \in \{1, 2, ..., 6\}$, but now with weight w^{T} given by

$$w^{\mathrm{T}} = \left| \sum_{i=1}^{i=5} T_{\mathrm{original}}(p_{n+i}(s_{n+i})) - T_{\mathrm{host}}(p_{n+i}(x_{n+i})) \right| \tag{C.3}$$

where s_i denote the original codon choices in the original organism. Now this weight denotes the linear difference between five original codon choices in the organism human, and five (possibly different) choices in host organism yeast. Note that the translation speed functions T now explicitely denote for which organism they are calculated, the original or host.

To find the sequence G'' where both the translation speed landscape and the nucleosome energy landscape in a host organism are close to their original counterparts, we only need to change the weights of $\mathcal{G}_{\mathrm{gene}}$. The weight $w^{\mathrm{T\&E}}$ of edges between

$$p_n(x_n)p_{n+1}(x_{n+1})p_{n+2}(x_{n+2})p_{n+3}(x_{n+3})p_{n+4}(x_{n+4})$$

and

$$p_{n+1}(x_{n+1})p_{n+2}(x_{n+2})p_{n+3}(x_{n+3})p_{n+4}(x_{n+4})p_{n+5}(x_{n+5})$$

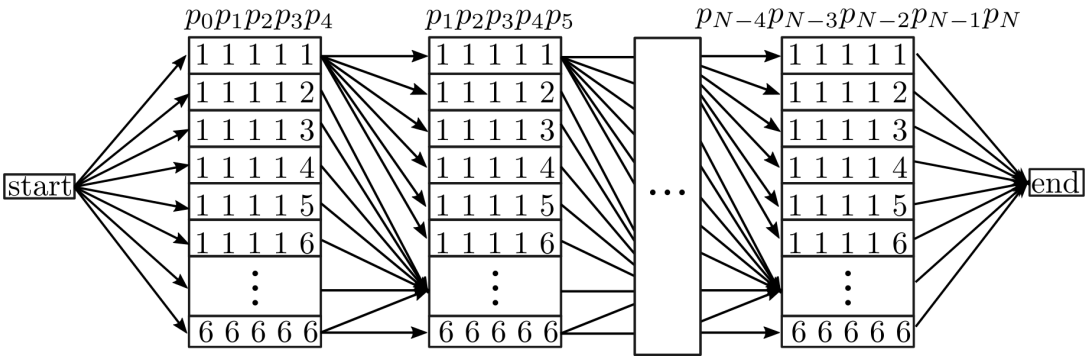


Figure C.3: Graph $\mathcal{G}_{\text{gene}}$ is similar to graph $\mathcal{G}_{\text{E}}$, see figure C.2. The graph includes the entire gene with N the number of codons on the gene. The weights of the graph depend on its purpose: the weights can be defined such that the closest possible translation speed landscape is found for a gene in a host organism, or a combination of the closest translation speed and nucleosome energy landscapes.

for any $x_i \in \{1, 2, ..., 6\}$ are now given by

$$w^{\text{T\&E}} = c_T w^{\text{T}} + c_E w^{\text{E}} \tag{C.4}$$

with

$$w^{\text{E}} = \sum_{j=-7}^{147+7-2} \left| \sum_{i=-7}^{i=7-2} E_{i+j}(S_{p+2+i}, S_{p+1+i}, S_{p+i}) - E_{i+j}(X_{p+2+i}, X_{p+1+i}, X_{p+i}) \right| \tag{C.5}$$

where X is a sequence of 15 base pairs, the sequence corresponding to

$$p_{n+1}(x_{n+1})p_{n+2}(x_{n+2})p_{n+3}(x_{n+3})p_{n+4}(x_{n+4})p_{n+5}(x_{n+5})$$

and S denotes the 15 base pairs in the original organism corresponding to

$$p_{n+1}(s_{n+1})p_{n+2}(s_{n+2})p_{n+3}(s_{n+3})p_{n+4}(s_{n+4})p_{n+5}(s_{n+5}).$$

C.4 Genetically modified organisms: many genes

In section 4.5, we introduced a method to, when one puts a gene in a different organism, this all three layers of information on the gene would be close to the original. Fig. 4.5 showed the results for one exon of the gene TNF. To remove possible bias from our results, we use the same method on a variety of human genes, randomly selected with a few non-biasing features: the exons of each transcript are fully translated and the first exon has a length ≥ 500 (the latter condition ensures a nucleosome landscape of significant size). The results are depicted in Figs. C.4-C.14.

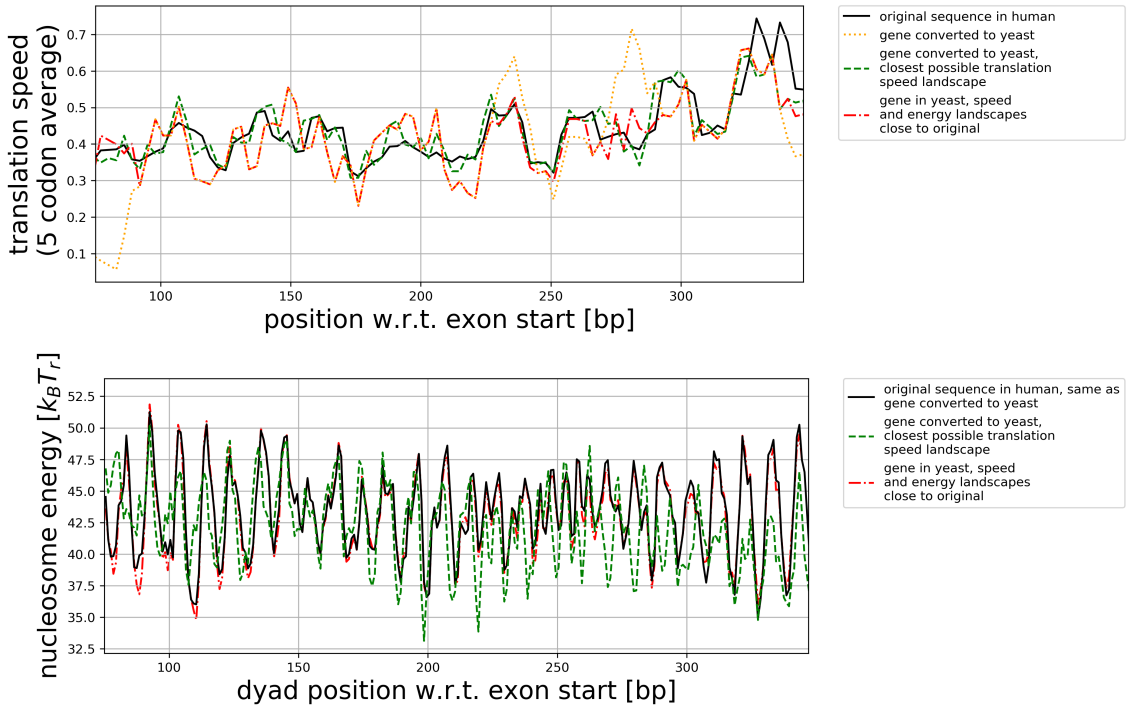


Figure C.4: Same as Fig. 4.5 but for the first exon of transcript OR6P1-001 of gene OR6P1 from human. This transcript was randomly selected with a few non-biasing features: the exons of each transcript are fully translated and the first exon has a length ≥ 500. As in Fig. 4.5, (a) depicts the translation speed landscape of this exon in three organisms: the original (human) and two possible host organisms: yeast and rice. Again, (b) shows the original landscape as well as the highest and lowest possible translation speed values in the hosts.

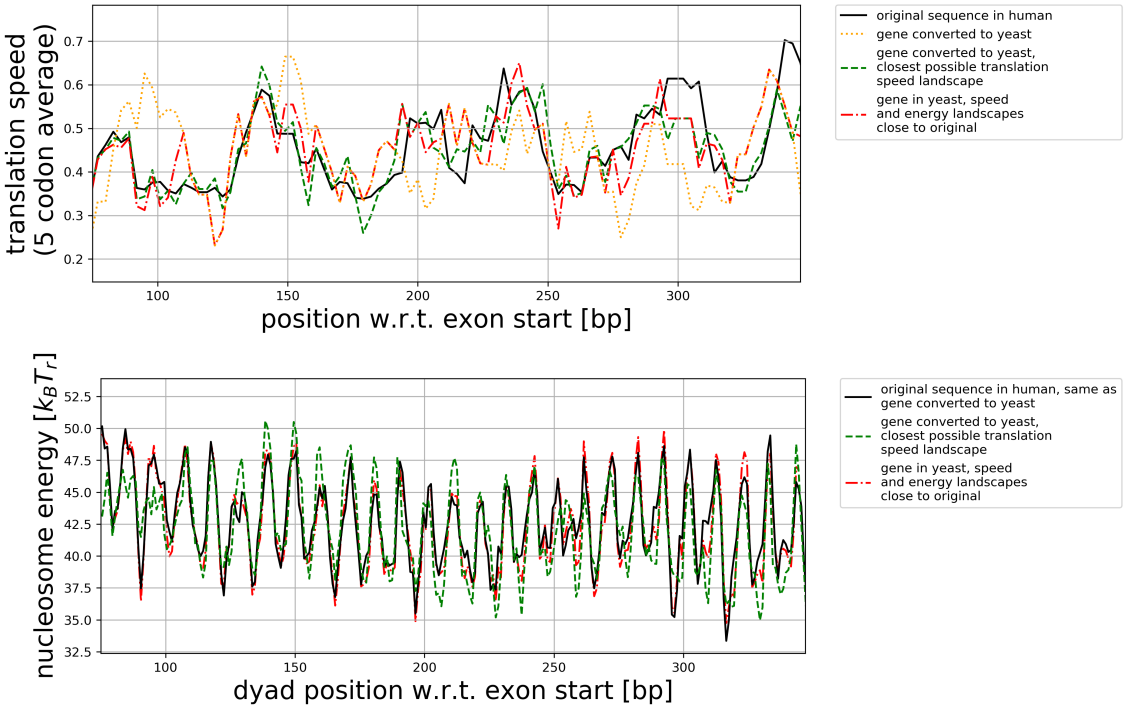


Figure C.5: Same as Fig. C.4 but for the first exon of transcript OR10J3-201 of gene OR10J3 from human.

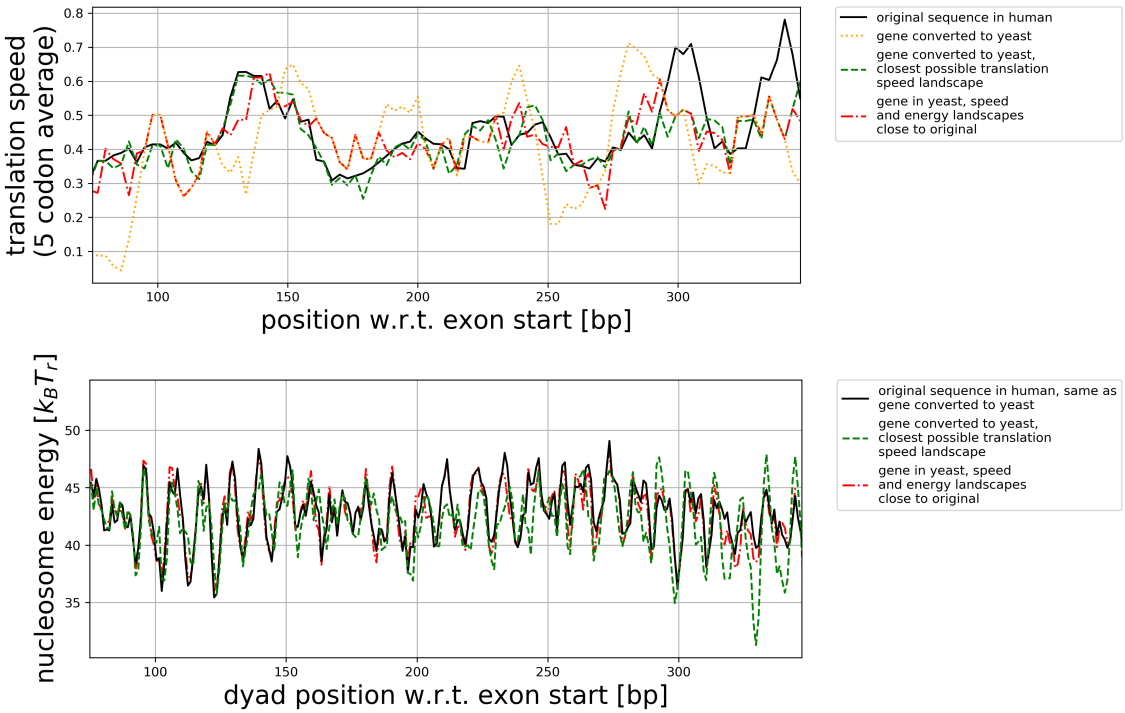


Figure C.6: Same as Fig. C.4 but for the first exon of transcript OR10T2-201 of gene OR10T2 from human.

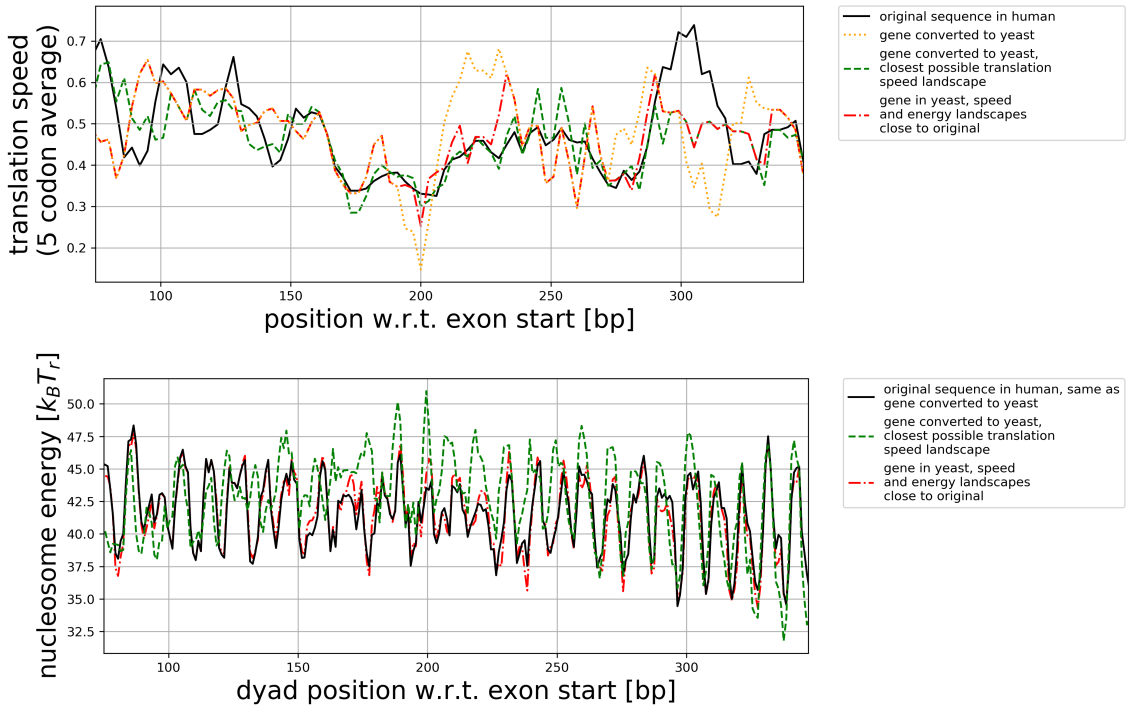


Figure C.7: Same as Fig. C.4 but for the first exon of transcript OR2T6-201 of gene OR2T6 from human.

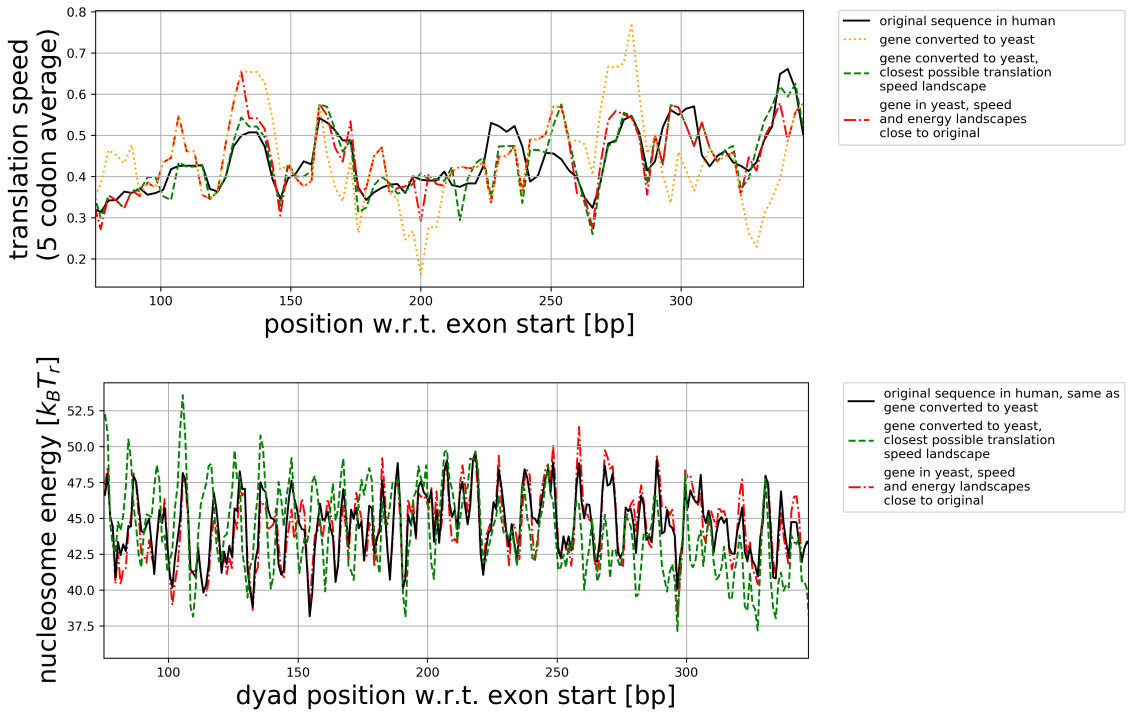


Figure C.8: Same as Fig. C.4 but for the first exon of transcript OR2M4-201 of gene OR2M4 from human.

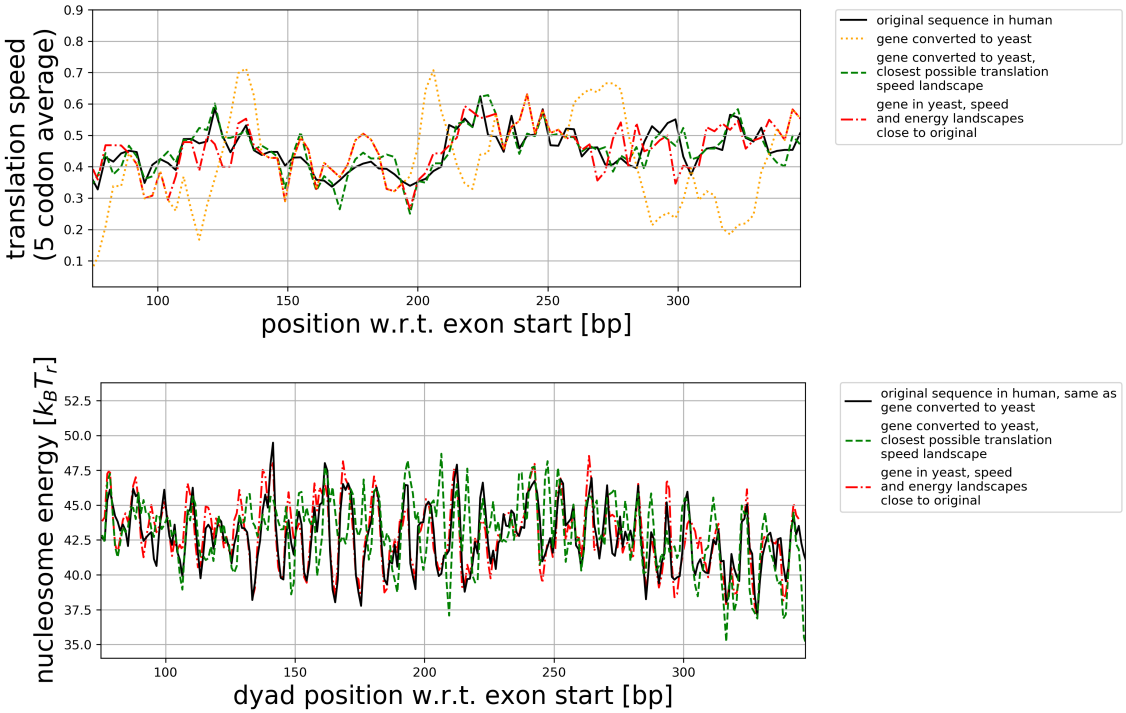


Figure C.9: Same as Fig. C.4 but for the first exon of transcript OR14K1-201 of gene OR14K1 from human.

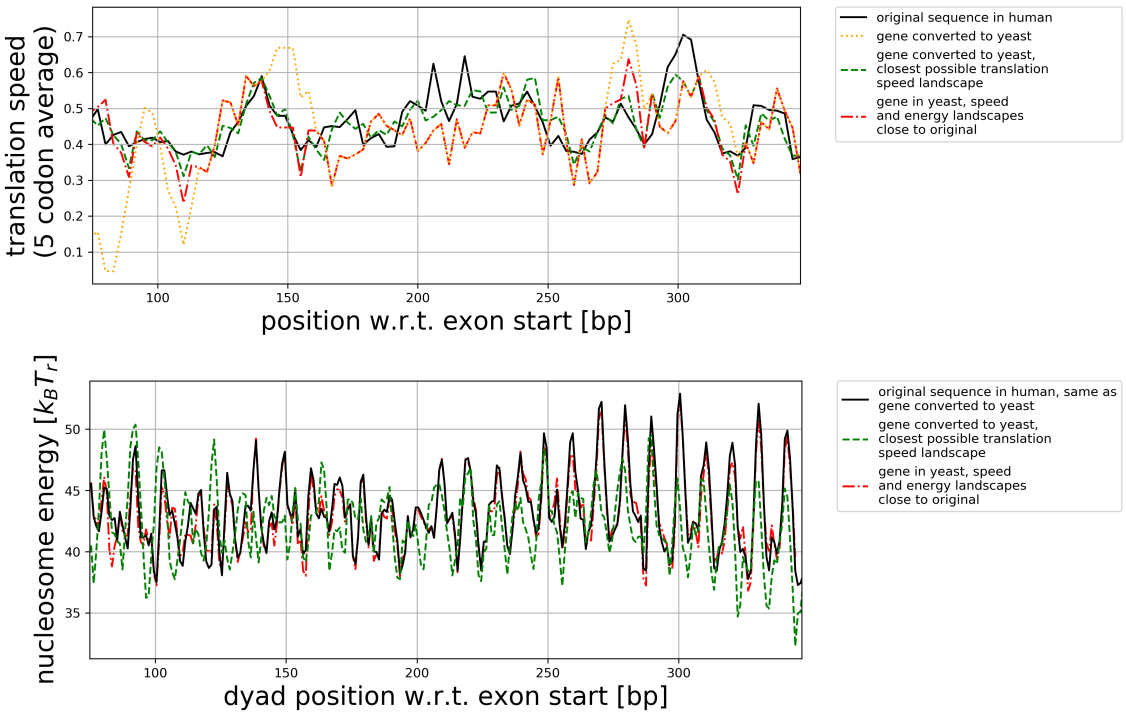


Figure C.10: Same as Fig. C.4 but for the first exon of transcript OR10K2 of gene OR10K2-201 from human.

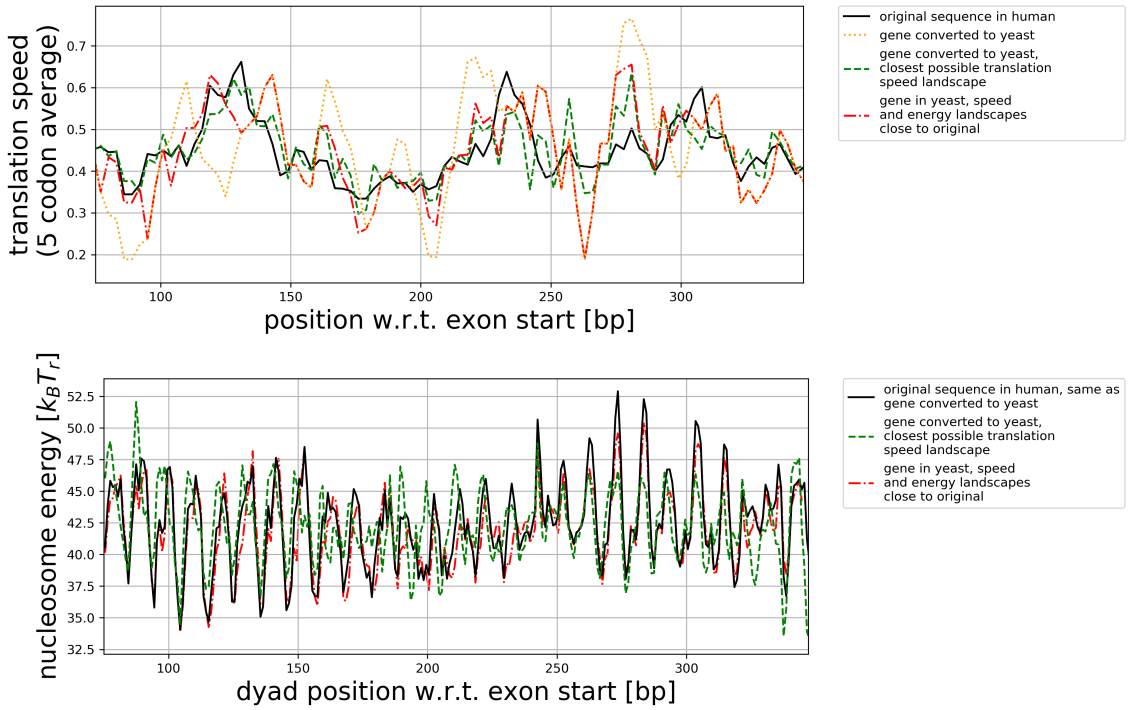
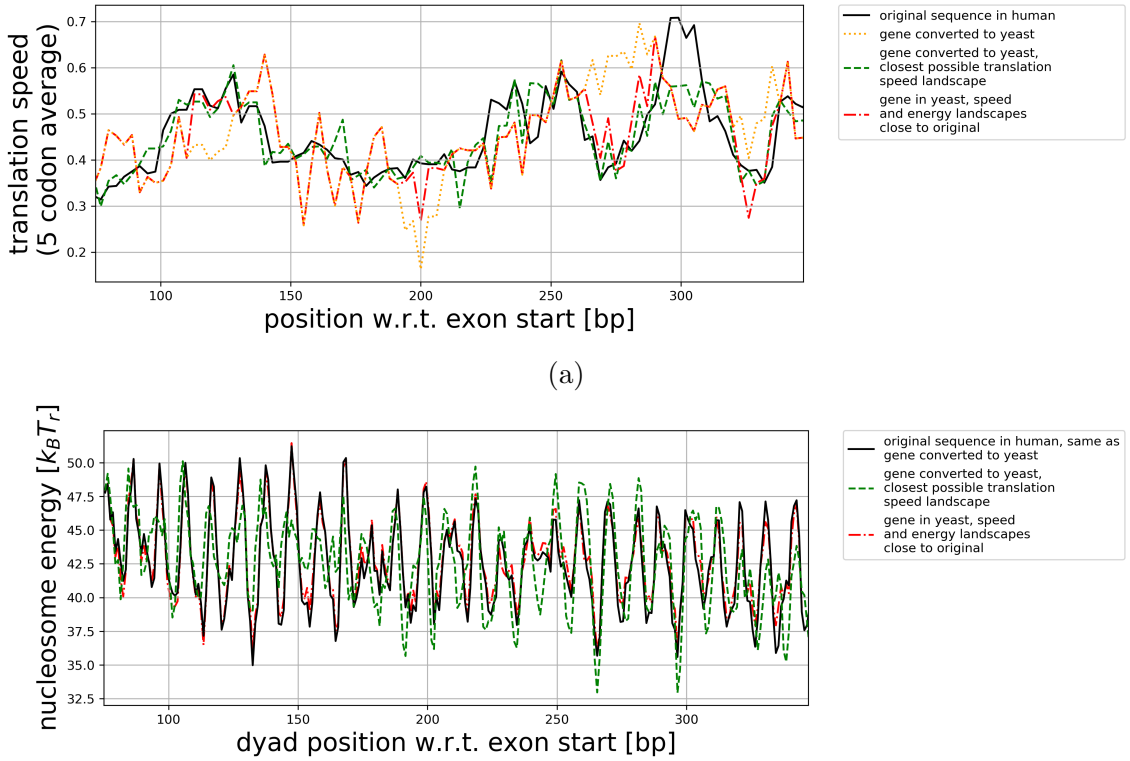


Figure C.11: Same as Fig. C.4 but for the first exon of transcript OR2T35-201 of gene OR2T35 from human.



(a)

Figure C.12: Same as Fig. C.4 but for the first exon of transcript OR2M7-201 of gene OR2M7 from human.

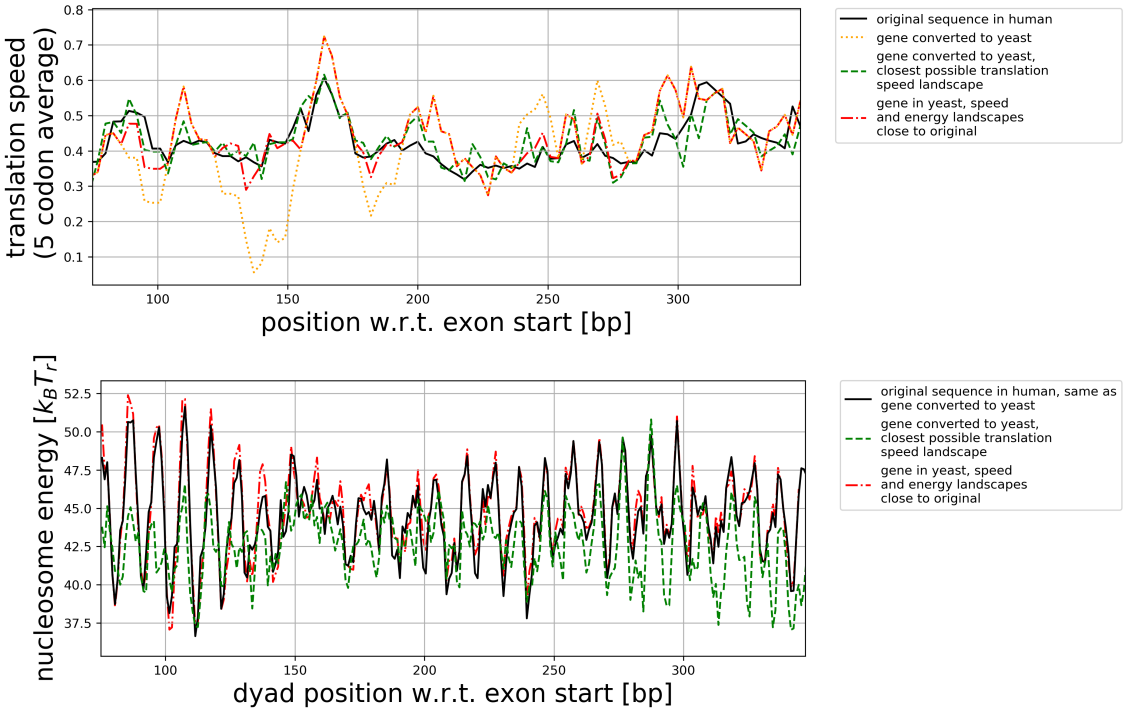


Figure C.13: Same as Fig. C.4 but for the first exon of transcript OR6Y1 of gene OR6Y1-201 from human.

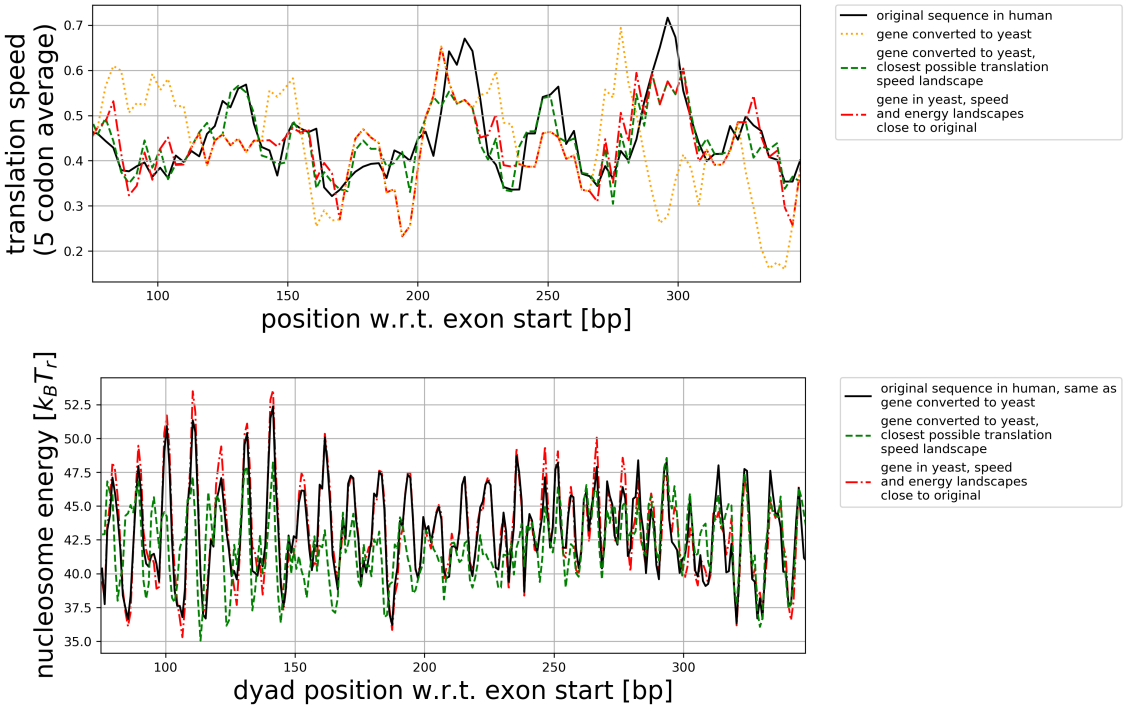


Figure C.14: Same as Fig. C.4 but for the first exon of transcript OR14C36 of gene OR14C36-201 from human.

Appendix D

How mechanical information is multiplexed on the transcribed regions of protein-coding genes

D.1 Data acquisition using Biomart

Here we provide a manual of sorts to obtain genome data the same way we did. We have acquired the genome data from the Ensembl Project website (www.ensembl.org), using their web-based tool Biomart. From Biomart, we always used the database Ensembl Genes 101. From the database we could pick an organism (such as Human genes). Under Filters, we expand the Gene menu and set the transcript type to protein-coding. Under Attributes, we chose Sequences, and under the SEQUENCES menu we chose Unspliced (Transcript), set Upstream flank to 1000 and Downstream flank to 1000. Under Attributes, in the HEADER INFORMATION menu, we chose, in this exact order: Transcript stable ID, Strand, Transcription start site (TSS), Genomic coding start, Genomic coding end, Exon region start (bp), Exon region end (bp). Using this header information we could determine the positions of the exons, introns and UTRs on any gene. By pressing the results button, and subsequently the Go button, one obtains the data in a plain text file.

Alternatively, one can download the data more efficiently. By pressing the XML button one can show the query in XML Web Service Format, which can then be used to download the data using a software package such as *wget*. We downloaded the required genomes by simply replacing the name of the organism in the XML string by the name of any available organism. An example of a command line to download the data for human is

```
wget -O human.txt 'http://www.ensembl.org/biomart/martservice?query=<?xml
    version="1.0" encoding="UTF-8"?><!DOCTYPE Query><Query
    virtualSchemaName = "default" formatter = "FASTA" header = "0"
    uniqueRows = "0" count = "" datasetConfigVersion = "0.6" ><Dataset
    name = "hsapiens_gene_ensembl" interface = "default" ><Filter name =
    "downstream_flank" value = "1000"/><Filter name = "upstream_flank"
    value = "1000"/><Filter name = "transcript_biotype" value =
    "protein_coding"/><Attribute name = "ensembl_transcript_id"
    /><Attribute name = "transcript_exon_intron" /><Attribute name =
    "strand" /><Attribute name = "transcription_start_site" /><Attribute
    name = "genomic_coding_start" /><Attribute name =
    "genomic_coding_end" /><Attribute name = "exon_chrom_start"
    /><Attribute name = "exon_chrom_end" /></Dataset></Query>'
```

For plants, e.g. for *Oryza sativa*, the command line can be given by

```
wget -O osativa.txt
    'http://plants.ensembl.org/biomart/martservice?query=<?xml
    version="1.0" encoding="UTF-8"?><!DOCTYPE Query><Query
    virtualSchemaName = "plants_mart" formatter = "FASTA" header = "0"
    uniqueRows = "0" count = "" datasetConfigVersion = "0.6" ><Dataset
    name = "osativa_eg_gene" interface = "default" ><Filter name =
    "downstream_flank" value = "1000"/><Filter name = "upstream_flank"
    value = "1000"/><Filter name = "transcript_biotype" value =
    "protein_coding"/><Attribute name = "ensembl_transcript_id"
    /><Attribute name = "transcript_exon_intron" /><Attribute name =
    "strand" /><Attribute name = "transcription_start_site" /><Attribute
    name = "genomic_coding_start" /><Attribute name =
    "genomic_coding_end" /><Attribute name = "exon_chrom_start"
    /><Attribute name = "exon_chrom_end" /></Dataset></Query>'
```

The Python code we use to turn the raw data into usable data is depicted below. An early version of this code was provided by Rhys Bird.

```
#this function requires a data file from the Biomart webtool. It provides
    three lists: cds2 is a list containing header information such as the
    name of the transcript, seq2 is a list containing 2000 bases
    corresponding to any transcipt, starting 1000 bp before the TSS. The
    list codingseq2 contains the same, but some of the base pairs have
    been replaced: all intronic bp are replaced by ";", 5'UTRs are
    replaced by "<", 3'UTR by ">".

def GetGenesShort(inputfile_string,upstream=1000, downstream=1000):
cds = []
seq = []
tempseq = ''
x=0
#here seq will be a list containing ALL bases corresponding to any
    transcipt
with open(inputfile_string) as inputfile:
```

```
for line in inputfile:
if line[0] == '>':
x+=1
if line.strip() != '>':
cds.append(line.replace('>','').split('|'))
if tempseq != '':
seq.append(tempseq)
tempseq = ''
else:
tempseq += line.replace('\n','')
for i in range(len(cds)):
cds[i][3] = cds[i][3].split(';')
cds[i][4] = cds[i][4].split(';')
cds[i][5] = cds[i][5].split(';')
cds[i][6] = cds[i][6].split(';')
codingseq = ['']*len(seq)
for i in range(len(seq)):
Xseq = [';']*len(seq[i])
#cds[i][1] tells us whether the raw data is 5' to 3', or 3' to 5'. In the
    latter scenario, the data is flipped such that everything is 5' to 3'.
#all UTRs are first replaced by ">", later we substitute it by "<" for
    5'UTRs.
if int(cds[i][1]) == 1:
for j in range(len(cds[i][3])):
start = int(cds[i][3][j]) - int(cds[i][2])+upstream
end = int(cds[i][4][j]) - (int(cds[i][2])-1)+upstream
Xseq[start:end] = [">" for _ in range(abs(end-start))]
list(seq[i][start:end])
for j in range(len(cds[i][5])):
start = int(cds[i][5][j]) - int(cds[i][2])+upstream
end = int(cds[i][6][j]) - (int(cds[i][2])-1)+upstream
Xseq[start:end] = list(seq[i][start:end])
elif int(cds[i][1]) == -1:
for j in range(len(cds[i][3])):
start = int(cds[i][2]) - int(cds[i][4][j])+upstream
end = int(cds[i][2]) - (int(cds[i][3][j])-1)+upstream
Xseq[start:end] = [">" for _ in range(abs(end-start))]
for j in range(len(cds[i][5])):
start = int(cds[i][2]) - int(cds[i][6][j])+upstream
end = int(cds[i][2]) - (int(cds[i][5][j])-1)+upstream
Xseq[start:end] = list(seq[i][start:end])
codingseq[i] += ''.join(Xseq)
for i in range(len(codingseq)):
codingseq[i]=upstream*","+codingseq[i][upstream:len(codingseq[i])-downstream]
    + downstream*","
seq2 = []
codingseq2 = []
cds2 = []
x=0
#here ">" is substituted by "<" for 5'UTRs. Also, all sequences are cut
```

```
    off after length 2000.
for i in range(len(seq)):
if len(seq[i])==len(codingseq[i]):
seq2.append(seq[i][0:2000])
cds2.append(cds[i][0:2000])
codingseq2.append([])
codingseq2[x]=codingseq[i][0:1000]
for j in range(1000,2000):
if codingseq[i][j] in ["A","T","C","G"]:
codingseq2[x]+=codingseq[i][j:2000]
break
if codingseq[i][j]==">":
codingseq2[x]+="<"
else:
codingseq2[x]+=codingseq[i][j]
x+=1
return cds2,seq2,codingseq2
```

D.2 List of animals used to obtain data

This section serves to provide a table with the list of animals of which data was obtained from biomart.

1	dmelanogaster	Drosophila melanogaster	A fruitfly
2	celegans	Caenorhabditis elegans	A nematode
3	cintestinales	Ciona intestinalis	Vase tunicate
4	csavignyi	Ciona savignyi	Solitary sea squirt
5	pmarinus	Petromyzon marinus	Sea lamprey
6	loculatus	Lepisosteus oculatus	Spotted gar
7	amexicanus	Astyanax mexicanus	Mexican tetra
8	trubripes	Takifugu rubripes	Japanese puffer
9	tnigroviridis	Tetraodon nigroviridis	Green spotted puffer
10	oniloticus	Oreochromis niloticus	Nile tilapia
11	gaculeatus	Gasterosteus aculeatus	Three-spined stickleback
12	olhni	Oryzias latipes	Japanese rice fish
13	pformosa	Poecilia formosa	Amazon molly
14	xmaculatus	Xiphophorus maculatus	Southern platyfish
15	lchalumnae	Latimeria chalumnae	West Indian Ocean coelacanth
16	xtropicalis	Xenopus tropicalis	Western clawed frog
17	acarolinensis	Anolis carolinensis	Green anole
18	psinensis	Pelodiscus sinensis	Chinese softshell turtle
19	falbicollis	Ficedula albicollis	Collared flycatcher
20	aplatyrhynchos	Anas platyrhynchos	Mallard
21	ggallus	Gallus gallus	Red junglefowl
22	mgallopavo	Meleagris gallopavo	Wild turkey
23	oanatinus	Ornithorhynchus anatinus	Platypus
24	mdomestica	Monodelphis domestica	Gray short-tailed opossum
25	sharrisii	Sarcophilus harrisii	Tasmanian devil
26	sscrofa	Sus scrofa	Wild boar
27	btaurus	Bos taurus	Cow
28	oaries	Ovis aries	Sheep
29	mlucifugus	Myotis lucifugus	Little brown bat
30	ecallabus	Equus caballus	Horse
31	fcatus	Felis catus	Cat
32	mpfuro	Mustela putorius furo	Ferret
33	ocuniculus	Oryctolagus cuniculus	European rabbit
34	cporcellus	Cavia porcellus	Guinea pig
35	itridecemlineatus	Ictidomys tridecemlineatus	Thirteen-lined ground squirrel
36	dordii	Dipodomys ordii	Ord's kangaroo rat
37	mmusculus	Mus musculus	House mouse
38	rnorvegicus	Rattus norvegicus	Brown rat
39	mmurinus	Microcebus murinus	Gray mouse lemur
41	ogarnettii	Otolemur garnettii	Northern greater galago
42	csyrichta	Carlito syrichta	Philippine tarsier
43	cjacchus	Callithrix jacchus	Common marmoset
44	csabaeus	Chlorocebus sabaeus	Green monkey
45	panubis	Papio anubis	Olive baboon

continues on the next page

46	mmulatta	Macaca mulatta	Rhesus macaque
47	nleucogenys	Nomascus leucogenys	Northern white-cheeked gibbon
48	pabelii	Pongo abelii	Sumatran orangutan
49	ggorilla	Gorilla gorilla gorilla	Western lowland gorilla
50	hsapiens	Homo sapiens	Human
51	ptroglodytes	Pan troglodytes	Chimpanzee

Table D.1: This table depicts the list of animals used to obtain data. It depicts the names of the organisms as they appear in Biomart, as well as their Latin names and a short description of the animal.

Bibliography

[1] R. Dahm, "Discovering DNA: Friedrich Miescher and the early years of nucleic acid research," *Human Genetics*, vol. 122, no. 6, pp. 565–581, 2008.

[2] H. Lodish, A. Berk, C. A. Kaiser, M. Krieger, A. Bretscher, H. Ploegh, A. Amon, and M. P. Scott, *Molecular Cell Biology, Seventh Edition*. W. H. Freeman and Company, 2013.

[3] G. A. Maston, S. K. Evans, and M. R. Green, "Transcriptional regulatory elements in the human genome," *Annual Review of Genomics and Human Genetics*, vol. 7, no. 1, pp. 29–59, 2006.

[4] S. C. Satchwell, H. R. Drew, and A. A. Travers, "Sequence periodicities in chicken nucleosome core DNA," *Journal of Molecular Biology*, vol. 191, pp. 659–675, 1986.

[5] E. Segal, Y. Fondufe-Mittendorf, L. Chen, A. Thaström, I. K. M. Y. Field, J. Z. Wang, and J. Widom, "A genomic code for nucleosome positioning," *Nature*, vol. 442, pp. 772–778, 2006.

[6] N. Kaplan, I. K. Moore, Y. Fondufe-Mittendorf, A. J. Gossett, D. Tillo, Y. Field, E. M. LeProust, T. R. Hughes, J. D. Lieb, J. Widom, and E. Segal, "The DNA-encoded nucleosome organization of a eukaryotic genome," *Nature*, vol. 458, pp. 362–366, 2009.

[7] P. T. Lowary and J. Widom, "Nucleosome packaging and nucleosome positioning of genomic DNA," *Proceedings of the National Academy of Sciences*, vol. 94, no. 4, pp. 1183–1188, 1997.

[8] G. Drillon, F. A. B. Audit, and A. Arneodo, "Evidence of selection for an accessible nucleosomal array in human," *BMC Genomics*, vol. 17, p. 526, 2016.

[9] M. Tompitak, C. Vaillant, and H. Schiessel, "Genomes of multicellular organisms have evolved to attract nucleosomes to promoter regions," *Biophysical Journal*, vol. 112, pp. 505–511, 2017.

[10] M. Tompitak, G. T. Barkema, and H. Schiessel, "Benchmarking and refining probability-based models for nucleosome-DNA interaction," *BMC Bioinformatics*, vol. 18, p. 157, 2017.

[11] B. Eslami-Mossallam, H. Schiessel, and J. van Noort, "Nucleosome dynamics: Sequence matters," *Advances in Colloid and Interface Science*, vol. 232, pp. 101–113, 2016.

[12] C. R. Calladine and H. R. Drew, "A base-centred explanation of the B-to-A transition in DNA," *Journal of Molecular Biology*, vol. 178, pp. 773–782, 1984.

[13] B. D. Coleman, W. K. Olson, and D. Swigdon, "Theory of sequence-dependent DNA elasticity," *Journal of Chemical Physics*, vol. 118, pp. 7127–7140, 2003.

[14] B. Eslami-Mossallam, R. D. Schram, M. Tompitak, J. van Noort, and H. Schiessel, "Multiplexing genetic and nucleosome positioning codes: A computational approach," *PLoS ONE*, vol. 11, p. e0156905, 2016.

[15] M. Y. Tolstorukov, A. V. Colasanti, D. M. McCandlish, W. K. Olson, and V. B. Zhurkin, "A novel role-and-slide mechanism for DNA folding in chromatin: implications for nucleosome positioning," *Journal of Molecular Biology*, vol. 371, pp. 725–738, 2007.

[16] T. Drsata, N. Spackova, P. Jurecka, M. Zgarbova, S. Sponer, and F. Lankas, "Mechanical properties of symmetric and asymmetric DNA A-tracts: implications for looping and nucleosome positioning," *Nucl. Acids Res.*, vol. 42, pp. 7383–7394, 2014.

[17] D. Norouzi and F. Mohammad-Rafiee, "DNA conformation and energy in nucleosome core: a theoretical approach," *J. Biomol. Struct. Dyn.*, vol. 32, pp. 104–114, 2014.

[18] A. Fathizadeh, A. B. Besya, M. R. Ejtehadi, and H. Schiessel, "Rigid-body molecular dynamics of DNA inside a nucleosome," *European Physical Journal E*, vol. 36, p. 21, 2013.

[19] L. de Bruin, M. Tompitak, B. Eslami-Mossallam, and H. Schiessel, "Why do nucleosomes unwrap asymmetrically?," *J. Phys. Chem. B.*, vol. 120, pp. 5855–5863, 2016.

[20] C. Anselmi, G. Bocchinfuso, P. D. Santis, M. Savino, and A. Scipioni, "A theoretical model for the prediction of sequence-dependent nucleosome thermodynamic stability," *Biophysical Journal*, vol. 79, pp. 601–613, 2000.

[21] C. Vaillant, B. Audit, and A. Arneodo, "Experiments confirm the influence of genome long-range correlations on nucleosome positioning," *Physical Review Letters*, vol. 99, p. 218103, 2007.

[22] S. Balasubramanian, F. Xu, and W. K. Olson, "DNA sequence-directed organization of chromatin: structure-based computational analysis of nucleosome-binding sequences," *Biophysical Journal*, vol. 96.

[23] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *J. ACM*, vol. 34, no. 3, p. 596–615, 1987.

[24] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[25] A. Bitran, W. M. Jacobs, X. Zhai, and E. Shakhnovich, "Cotranslational folding allows misfolding-prone proteins to circumvent deep kinetic traps," *Proceedings of the National Academy of Sciences*, vol. 117, no. 3, pp. 1485–1495, 2020.

[26] H. Gingold, D. Tehler, N. Christoffersen, M. Nielsen, F. Asmar, S. Kooistra, N. Christophersen, L. L. Christensen, M. Borre, D. Karina, L. Dyrskjøt, C. Andersen, E. Hulleman, T. Wurdinger, E. Ralfkiær, K. Helin, K. Grønbaek, T. Orntoft, S. Waszak, and Y. Pilpel, "A dual program for translation regulation in cellular proliferation and differentiation," *Cell*, vol. 158, pp. 1281–92, 2014.

[27] K. C. Stein and J. Frydman, "The stop-and-go traffic regulating protein biogenesis: how translation kinetics controls proteostasis," *Journal of Biological Chemistry*, vol. 294, no. 6, pp. 2076–2084, 2019.

[28] K. Stein and J. Frydman, "The stop-and-go traffic regulating protein biogenesis: How translation kinetics controls proteostasis," *Journal of Biological Chemistry*, vol. 294, p. jbc.REV118.002814, 2018.

[29] S. Rudorf, M. Thommen, M. V. Rodnina, and R. Lipowsky, "Deducing the kinetics of protein synthesis in vivo from the transition rates measured in vitro," *PLoS Comput Biol*, vol. 10, no. 10, p. e1003909, 2014.

[30] E. Dolgin, "The most popular genes in the human genome," *Nature*, vol. 551, no. 7681, pp. 427–431, 2017.

[31] M. Zuiddam, R. Everaers, and H. Schiessel, "Physics behind the mechanical nucleosome positioning code," *Physical Review E*, vol. 96, p. 052412, 2017.

[32] W. K. Olson, A. A. Gorin, X. J. Lu, L. M. Hock, and V. B. Zhurkin, "DNA sequence-dependent deformability deduced from protein-DNA crystal complexes," *Proc. Natl. Acad. Sci. USA*, vol. 95, pp. 11163–11168, 1998.

[33] K. Luger, A. W. Mäder, R. K. Richmond, D. F. Sargent, and T. J. Richmond, "Crystal structure of the nucleosome core particle at 2.8 åresolution," *Nature*, vol. 389, pp. 251–260, 1997.

[34] H. Schiessel, "The physics of chromatin," *J. Phys.: Condens. Matter*, vol. 15, pp. R699–R774, 2003.

[35] T. Vavouri and B. Lehner, "Chromatin organization in sperm may be the major functional consequence of base composition variation in the human genome," *PLoS Genet.*, vol. 7, p. e1002036, 2011.

[36] A. V. Morozov, K. Fortney, D. A. Gaykalova, V. M. Studitsky, J. Widom, and E. D. Siggia, "Using DNA mechanics to predict in vitro nucleosome positions and formation energies," *Nucl. Acids Res*, vol. 37, pp. 4707–4722, 2009.

[37] N. B. Becker and R. Everaers, "DNA nanomechanics in the nucleosome," *Structure*, vol. 17, pp. 579–589, 2009.

[38] N. B. Becker, L. Wolff, and R. Everaers, "Indirect readout: detection of optimized subsequences and calculation of relative binding affinities using different DNA elastic potentials," *Nucl Acids Res.*, vol. 34, pp. 5638–5649, 2006.

[39] R. Lavery, M. Moakher, J. H. Maddocks, D. Petkeviciute, and K. Zakrzewska, "Conformational analysis of nucleic acids revisited: Curves+," *Nucleic Acids Res.*, vol. 37, pp. 5917–5929, 2009.

[40] V. B. Teif, "General transfer matrix formalism to calculate DNA-protein-drug binding in gene regulation: application to o_r operator of phage λ," *Nucleic Acids Res.*, vol. 35, p. e80, 2007.

[41] G. Chevereau, A. Arneodo, and C. Vaillant, "Influence of the genomic sequence on the primary structure of chromatin," *Frontiers in Life Science*, vol. 5, pp. 29–68, 2011.

[42] G. S. Freeman, J. P. Lequieu, D. M. Hinckley, J. K. Whitmer, and J. de Pablo, "DNA shape dominates sequence affinity in nucleosome formation," *Physical Review Letters*, vol. 113, p. 168101, 2014.

[43] M. Zuiddam and H. Schiessel, "Shortest paths through synonymous genomes," *Physical Review E*, vol. 99, p. 012422, 2019.

[44] G. Meersseman, S. Pennings, and E. Bradbury, "Mobile nucleosomes–a general behavior.," *The EMBO Journal*, vol. 11, no. 8, pp. 2951–2959, 1992.

[45] I. M. Kulic and H. Schiessel, "Chromatin dynamics: Nucleosomes go mobile through twist defects," *Physical Review Letters*, vol. 91, p. 148103, 2003.

[46] G. B. Brandani, T. Niina, C. Tan, and S. Takada, "DNA sliding in nucleosomes via twist defect propagation revealed by molecular simulations," *Nucleic Acids Research*, vol. 46, no. 6, pp. 2788–2801, 2018.

[47] H. Schiessel, J. Widom, R. F. Bruinsma, and W. M. Gelbart, "Polymer reptation and nucleosome repositioning," *Physical Review Letters*, vol. 86, pp. 4414–4417, 2001.

[48] I. M. Kulic and H. Schiessel, "Nucleosome Repositioning via Loop Formation," *Biophysical Journal*, vol. 84, no. 5, pp. 3197–3211, 2003.

[49] J. Lequieu, D. C. Schwartz, and J. J. de Pablo, "In silico evidence for sequence-dependent nucleosome sliding," *Proceedings of the National Academy of Sciences*, vol. 114, no. 44, pp. E9197–E9205, 2017.

[50] T. Niina, G. Brandani, C. Tan, and S. Takada, "Sequence-dependent nucleosome sliding in rotation-coupled and uncoupled modes revealed by molecular simulations," *PLOS Computational Biology*, vol. 13, p. e1005880, 2017.

[51] J. Winger, I. Nodelman, R. Levendosky, and G. Bowman, "A twist defect mechanism for ATP-dependent translocation of nucleosomal DNA," *eLife*, vol. 7, 2018.

[52] G. B. Brandani, T. Niina, C. Tan, and S. Takada, "DNA sliding in nucleosomes via twist defect propagation revealed by molecular simulations," *Nucleic Acids Research*, vol. 46, no. 6, pp. 2788–2801, 2018.

[53] E. Segal, Y. Fondufe-Mittendorf, L. Chen, A. Thaström, Y. Field, I. Moore, J. Wang, and J. Widom, "A genomic code for nucleosome positioning," *Nature*, vol. 442, pp. 772–8, 2006.

[54] K. Struhl and E. Segal, "Determinants of nucleosome positioning," *Nature structural & molecular biology*, vol. 20, pp. 267–73, 2013.

[55] R. D. Kornberg and L. Stryer, "Statistical distributions of nucleosomes: non-random locations by a stochastic mechanism," *Nucleic Acids Research*, vol. 16, no. 14, pp. 6677–6690, 1988.

[56] G. Chevereau, L. Palmeira, C. Thermes, A. Arneodo, and C. Vaillant, "Thermodynamics of intragenic nucleosome ordering," *Physical Review Letters*, vol. 103, p. 188103, 2009.

[57] F. Brunet, B. Audit, G. Drillon, F. Argoul, J. Volff, and A. Arneodo, "Evidence for DNA sequence encoding of an accessible nucleosomal array across vertebrates," *Biophysical Journal*, vol. 114, 2018.

[58] M. Tompitak, L. de Bruin, B. Eslami-Mossallam, and H. Schiessel, "Designing nucleosomal force sensors," *Physical Review E*, vol. 95, p. 052402, 2017.

[59] T. E. Shrader and D. M. Crothers, "Artificial nucleosome positioning sequences," *Proc. Natl. Acad. Sci. USA*, vol. 86, pp. 7418–7422, 1989.

[60] T. E. Shrader and D. M. Crothers, "Effect of DNA sequence and histone-histone interactions on nucleosome placement," *Journal of Molecular Biology*, vol. 216, pp. 69–84, 1990.

[61] J. A. Wondergem, H. Schiessel, and M. Tompitak, "Performing selex experiments in silico.," *The Journal of chemical physics*, vol. 147 17, p. 174101, 2017.

[62] A. Valouev, S. M. Johnson, S. D. Boyd, C. L. Smith, A. Z. Fire, and A. Sidow, "Determinants of nucleosome organization in primary human cells," *Nature*, vol. 474, pp. 516–520, 2011.

[63] S. Ercan, S. Lubling, E. Segal, and J. D. Lieb, "High nucleosome occupancy is encoded at x-linked gene promoters in c. elegans," *Genome Research*, vol. 21, p. 237–244, 2011.

[64] J. Culkin, L. de Bruin, M. Tompitak, R. Phillips, and H. Schiessel, "The role of DNA sequence in nucleosome breathing," *European Physical Journal E*, vol. 40, p. 106, 2017.

[65] K. J. Polach and J. Widom, "Mechanism of protein access to specific DNA sequences in chromatin: a dynamic equilibrium model for gene regulation," *Journal of Molecular Biology*, vol. 254, pp. 130–149, 1995.

[66] J. D. Anderson and J. Widom, "Sequence and position-dependence of the equilibrium accessibility of nucleosomal DNA target sites," *Journal of Molecular Biology*, vol. 296, pp. 979–987, 2000.

[67] T. T. M. Ngo, Q. Zhang, R. Zhou, J. G. Yodh, and T. Ha, "Asymmetric unwrapping of nucleosomes under tension directed by DNA local flexibility," *Cell*, vol. 160, pp. 1135–1144, 2015.

[68] E. Segal and J. Widom, "Poly(dA:dT) tracts: major determinants of nucleosome organization," *Current Opinion in Structural Biology*, vol. 19, pp. 65–71, 2009.

[69] P. T. Lowary and J. Widom, "New DNA sequence rules for high affinity binding to histone octamer and sequence-directed nucleosome positioning," *Journal of Molecular Biology*, vol. 276, pp. 19–42, 1998.

[70] S. Todolli, P. Perez, N. Clauvelin, and W. Olson, "Contributions of sequence to the higher-order structures of DNA," *Biophysical Journal*, vol. 112, 2016.

[71] G. Rosanio, J. Widom, and O. C. Uhlenbeck, "In vitro selection of DNAs with an increased propensity to form small circles," *Biopolymers*, vol. 103, no. 6, pp. 303–320, 2015.

[72] H. Meng, K. Andresen, and J. van Noort, "Quantitative analysis of single-molecule force spectroscopy on folded chromatin fibers," *Nucleic Acids Research*, vol. 43, no. 7, pp. 3578–3590, 2015.

[73] B. E. de Jong, T. B. Brouwer, A. Kaczmarczyk, B. Visscher, and J. van Noort, "Rigid basepair monte carlo simulations of one-start and two-start chromatin fiber unfolding by force," *Biophysical Journal*, vol. 115, no. 10, pp. 1848–1859, 2018.

[74] E. N. Trifonov, "The multiple codes of nucleotide sequences," *Bulletin of Mathematical Biology*, vol. 51, no. 4, pp. 417–432, 1989.

[75] E. N. Trifonov and J. L. Sussman, "The pitch of chromatin DNA is reflected in its nucleotide sequence," *Proceedings of the National Academy of Sciences*, vol. 77, no. 7, pp. 3816–3820, 1980.

[76] D. Tillo and T. Hughes, "G+C content dominates intrinsic nucleosome occupancy," *BMC Bioinformatics*, vol. 10, p. 442, 2009.

[77] J. Neipel, G. Brandani, and H. Schiessel, "Translational nucleosome positioning: A computational study," *Physical Review E*, vol. 101, p. 022405, 2020.

[78] F. Mohammad-Rafiee, I. M. Kulić, and H. Schiessel, "Theory of nucleosome corkscrew sliding in the presence of synthetic DNA ligands," *Journal of Molecular Biology*, vol. 344, no. 1, pp. 47–58, 2004.

[79] A. Z. Guo, J. Lequieu, and J. J. de Pablo, "Extracting collective motions underlying nucleosome dynamics via nonlinear manifold learning," *The Journal of Chemical Physics*, vol. 150, no. 5, p. 054902, 2019.

[80] S. Rudnizky, H. Khamis, O. Malik, P. Melamed, and A. Kaplan, "The base pair-scale diffusion of nucleosomes modulates binding of transcription factors," *Proceedings of the National Academy of Sciences*, vol. 116, no. 25, pp. 12161–12166, 2019.

[81] M. Li, X. Xia, Y. Tian, Q. Jia, X. Liu, Y. Lu, M. Li, X. Li, and Z. Chen, "Mechanism of DNA translocation underlying chromatin remodelling by snf2," *Nature*, vol. 567, p. 409–413, 2019.

[82] A. Sabantsev, R. Levendosky, X. Zhuang, G. Bowman, and S. Deindl, "Direct observation of coordinated DNA movements on the nucleosome during chromatin remodelling," *Nature Communications*, vol. 10, p. 1720, 2019.

[83] H. Schiessel and R. Blossey, "Pioneer transcription factors in chromatin remodeling: The kinetic proofreading view," *Physical Review E*, vol. 101, p. 040401, 2020.

[84] H. Dong, L. Nilsson, and C. G. Kurland, "Co-variation of tRNA abundance and codon usage in Escherichia *coli* at different growth rates," *Journal of molecular biology*, vol. 260, no. 5, pp. 649–663, 1996.

[85] M. A. Collart and B. Weiss, "Ribosome pausing, a dangerous necessity for co-translational events," *Nucleic acids research*, vol. 48, no. 3, pp. 1043–1055, 2020.

[86] D. López and F. Pazos, "Protein functional features are reflected in the patterns of mrna translation speed," *BMC genomics*, vol. 16, no. 1, pp. 1–13, 2015.

[87] S. Pechmann and J. Frydman, "Evolutionary conservation of codon optimality reveals hidden signatures of cotranslational folding," *Nature structural & molecular biology*, vol. 20, no. 2, p. 237, 2013.

[88] M. Zhou, J. Guo, J. Cha, M. Chae, S. Chen, J. M. Barral, M. S. Sachs, and Y. Liu, "Non-optimal codon usage affects expression, structure and function of clock protein frq," *Nature*, vol. 495, no. 7439, pp. 111–115, 2013.

[89] E. P. O'Brien, P. Ciryam, M. Vendruscolo, and C. M. Dobson, "Understanding the influence of codon translation rates on cotranslational protein folding," *Accounts of chemical research*, vol. 47, no. 5, pp. 1536–1544, 2014.

[90] M. Liutkute, E. Samatova, and M. V. Rodnina, "Cotranslational folding of proteins on the ribosome," *Biomolecules*, vol. 10, no. 1, p. 97, 2020.

[91] W. Chu, "Tumor necrosis factor," *Cancer letters*, vol. 328, no. 2, pp. 222–225, 2013.

[92] J. Frank and R. L. Gonzalez Jr, "Structure and dynamics of a processive brownian motor: the translating ribosome," *Annual review of biochemistry*, vol. 79, pp. 381–412, 2010.

[93] I. Wohlgemuth, C. Pohl, J. Mittelstaet, A. L. Konevega, and M. V. Rodnina, "Evolutionary optimization of speed and accuracy of decoding on the ribosome," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 366, no. 1580, pp. 2979–2986, 2011.

[94] D. N. Wilson and J. H. D. C. Cate, "The structure and function of the eukaryotic ribosome," *Cold Spring Harbor Perspectives in Biology*, vol. 4, no. 5, p. a011536, 2012.

[95] V. Haberle and A. Stark, "Eukaryotic core promoters and the functional basis of transcription initiation," *Nature reviews. Molecular cell biology*, vol. 19, no. 10, pp. 621–637, 2018.

[96] R. I. Vishwanath, "Nucleosome positioning: bringing order to the eukaryotic genome," *Trends in Cell Biology*, vol. 22, no. 5, pp. 250–256, 2012.

[97] M. Han and M. Grunstein, "Nucleosome loss activates yeast downstream promoters in vivo," *Cell*, vol. 55, no. 6, p. 1137—1145, 1988.

[98] R. Dreos, G. Ambrosini, and P. Bucher, "Influence of rotational nucleosome positioning on transcription start site selection in animal promoters," *PLoS computational biology*, vol. 12, no. 10, pp. e1005144–e1005144, 2016.

[99] D. Tillo, N. Kaplan, I. K. Moore, Y. Fondufe-Mittendorf, A. J. Gossett, Y. Field, J. D. Lieb, J. Widom, E. Segal, and T. R. Hughes, "High nucleosome occupancy is encoded at human regulatory sequences," *PloS one*, vol. 5, no. 2, pp. e9129–e9129, 2010.

[100] M. Chorev and L. Carmel, "The function of introns," *Frontiers in genetics*, vol. 3, pp. 55–55, 2012.

[101] F. Mignone, C. Gissi, S. Liuni, and G. Pesole, "Untranslated regions of mrnas," *Genome biology*, vol. 3, no. 3, 2002.

[102] C. team of Ensembl, "Species tree." `http://Feb2021.archive.ensembl.org/info/about/speciestree.html`. Accessed: 1-3-2021.

[103] T. Tuller, A. Carmi, K. Vestsigian, S. Navon, Y. Dorfan, J. Zaborske, T. Pan, O. Dahan, I. Furman, and Y. Pilpel, "An evolutionarily conserved mechanism for controlling the efficiency of protein translation," *Cell*, vol. 141, no. 2, p. 344—354, 2010.