Walking, Hopping and Jumping: A Model of Transcription Factor Dynamics on DNA

David J. Barnes[1] and Dominique Chu[1]

[1]School of Computing, University of Kent, CT2 7NF, Canterbury, UK
{d.j.barnes, d.f.chu}@kent.ac.uk

Abstract

We present a model of how transcription factors scan DNA to find their specific binding sites. Following the classical work of Winter et al. (1981), our model assumes two modes of transcription factor dynamics. Adjacent moves, where the proteins make a single step movement to one side, or short walks where the transcription factors slide along the DNA several binding sites at a time. The purpose of this article is twofold. Firstly, we discuss how such a system can be efficiently modelled computationally. Secondly, we analyse how the mean first binding times of transcription factors to their specific time depends on key parameters of the system.

Introduction

Regulation of gene activity can be understood as a computational process, in the sense that the cell reacts to changes in the environment by changing its internal states. There are several mechanisms the cell can use to make such internal changes. One important such mechanisms is the regulation of genes. In bacteria, gene regulation often involves the binding of regulatory proteins, so called *transcription factors* (TFs), to particular binding sites on the DNA.

One aspect that has commanded significant attention from bioscientists, physicists and systems biologists is the time required for regulatory proteins to find their target binding site on the genome. The problem is as follows: In order to turn a gene on (or indeed repress it) the TF needs to locate a specific binding site. The problem is that TFs are "sticky" to all parts of the DNA. When binding to the DNA a TF actually binds to an l-long sequence of nucleotides. The binding strength depends on the match between the bound sequence and an optimal pattern which represents the sequence of the specific binding site. The closer the match, the higher the affinity. While the binding affinity to specific sites is much higher than to most non-specific sites, the contribution of the latter is still significant enough to potentially "distract" a TF from locating its specific site. Furthermore, there are millions of non-specific sites and only few of the specific and active sites for each particular TF. Therefore, even though a TF spends very little time being bound to each of the non-specific sites, it may take a significant time to sample all of them before the specific site is eventually found. The process of a TF finding its specific binding site necessarily limits the speed of a biological computation.

This problem, which has been known about for a long time, was first addressed by Winter et al. (1981), who proposed a random walk model of facilitated diffusion. The idea of this model is that the TF performs a mixed 1D and 3D random walk. The 1D random walk explores a small adjacent neighborhood of DNA, while the 3D random walk allows the TF to explore far-away, unconnected parts of the genome. It has been suggested by Wunderlich and Mirny (2008); Slutsky et al. (2004) and Murugan (2009) that the most efficient exploration of the genome, in the sense that it offers the fastest location of the specific binding site, is achieved when the 3D and 1D components are weighted approximately equally.

Most of the above work has been analytical. There are also a number of other results available. In this article we will describe an approach to building an efficient computer simulation model of TFs finding their specific binding sites (Barnes and Chu (2010)). This new approach will allow realistically sized simulations, thus significantly expanding the scope of previous models. The essence of the efficiency of the model is a careful management of memory to make the problem scalable, regardless of genome occupancy.

The Model

The movement dynamics of TFs involves a search across a discrete (but very high) number of spatially organised binding sites. This suggests the potential for an individual agent-based modelling approach. The environment of the TF agents is a non-metric space; that is, there is no measure of distance between the agents. Embedded in this space is the DNA itself, which is represented as a string of the symbols `a,c,g,t` with periodic boundary condition. For all simulations reported here we used the genome of *E.coli* K12 (The University of Wisconsin (2009)). At any given time, every agent is either bound to one of the binding sites of the genome, or suspended in the non-metric space. We think of the space as a 'reservoir' of currently unbound TFs.

We define two types of agents, namely focal and non-focal TFs. We are primarily interested in the former, yet the latter are important in that their presence on the DNA could interfere with the search dynamics of the focal TFs. The number of non-focal TFs is kept constant during a specific simulation run (for reasons of computational efficiency), whereas the focal TFs are created and degraded with user-defined rates, hence particle numbers within the cell fluctuate over time.

Focal TFs have a definite binding *motif* $\mathbf{m}$ that is used to determine their binding energy and, hence, their mean binding time at every DNA binding site in the model. If the length of the binding motif $\mathbf{m}$ is l then the binding free energy to a particular sequence is calculated as follows:

$$F_s = \sum_{i=1}^{l} \omega_i \delta_{m_i, s_i} \tag{1}$$

Here, m_i is the i-th entry of the motif $\mathbf{m}$, s_i the corresponding base of the actual binding sequence $\mathbf{s}$ and ω_i the empirically determined weighting factor of the binding motif. In contrast, non-focal TFs do not have specific binding sites; rather, they share low, position-dependent affinities to all sites on the DNA. Rather than calculating the binding energies dynamically, the affinity values for both types of TF are pre-calculated for every position on the DNA and stored in arrays of the same length as the DNA, making binding-time calculation very efficient.

The model update algorithm is event based, with three main classes of event available at each step:

- Create a focal TF.
- Bind a TF of either type to the DNA.
- Unbind a bound TF from the DNA.

Unbind events can result in complete unbinding into the reservoir or short, local 1D movements. Essential for the reliability of the model is to design the update algorithm so that the behaviour of the model is correct with respect to the choice of parameters (in the sense that it reproduces the statistics implied by the various binding and unbinding rates). To achieve this, we have adapted the Gillespie algorithm (Gillespie (1977)) to schedule events.

On every event, regardless of its class, only a single TF is updated. Breaking down the event classes in more details: an update consists of one of the following actions:

- A new focal TF is created and might attempt to bind.
- A TF binds from the reservoir to the DNA.
- A bound TF unbinds from the DNA into the reservoir.
- A bound TF moves to an adjacent binding site on the DNA.
- A bound TF makes a short move, i.e., binds with a uniform probability to an available binding site in the vicinity of its current site. The range of what counts as "vicinity" is user determined.
- A bound TF is destroyed.

Scheduling of events

At model initialization all non-focal TFs are created and seeded onto random locations on the DNA via bind events at time zero. If there is insufficient space then the excess ends up in the reservoir. Then the creation times of all focal TFs are determined according to a user-defined rate, and creation events scheduled accordingly. Their lifetime is also determined at creation with a random number drawn from an exponential distribution with a mean of 1 over the deletion rate.

When its creation event occurs, a focal TF will immediately attempt to bind to a site on the genome with a user-defined probability; any such attempt is successful with a probability $p = N_{\text{free}}/N_{\text{range}}$ where N_{range} is the total number of binding sites in range and N_{free} is the number of unoccupied sites in that range. We specify N_{range} because the initial bind attempt for a focal TF takes place within a limited, user-defined birth range on the DNA. This models the effect that (in bacterial cells) transcription and translation are performed in one step and hence proteins are produced close to their gene.

If the newly-created TF does not bind, then it is placed in the reservoir and may have the opportunity to attempt a general bind (i.e., one over the full range of the DNA) at a later time. The range restriction only applies to the initial binding attempt of a focal TF.

Binding events

General binding is used both to seed initial occupancy of the DNA with non-focal TFs, and to support binding of both types of TF from the reservoir. A random available binding site is chosen from the full length of the DNA.

At the completion of every event, there is a probability that an unbound TF might attempt to bind from the reservoir. The time to the bind event is drawn from an exponential distribution with a mean of 1 over a value that depends upon the number of unbound TFs T_u, the number of available binding sites N_{free} along the full range of the DNA, and a constant factor k:

$$P(\text{bind}) = (kN_{\text{free}}T_u) \tag{2}$$

A new binding event will only be scheduled if it would occur before the next already scheduled event. This is because the binding probability depends on the current availability of binding sites which generally changes over time.

Unbinding events

The duration time of a DNA-protein bond depends on the affinity of the type of TF for its binding site; specifically, for focal TFs this affinity is determined from equation 1. It is drawn from a Poisson distribution with mean μ.

$$\mu = \exp(-\frac{F_s}{kT})$$

Here k is the Boltzmann constant and T the absolute temperature. Binding from the reservoir onto the DNA is determined stochastically with a given user-determined rate.

At every unbind event, the next state of the TF is determined stochastically. Assuming that the TF has not reached the end of its life (in which case it would be destroyed), with a user-defined probability one of the following options will apply to it:

- the TF will attempt to make a one place move left or right (an immediately scheduled bind event);
- the TF will attempt a short move within a user-defined range either side of the previous binding site (an immediately scheduled bind event);
- the TF goes into the reservoir.

Either move could fail, due to roadblocks, and lead to the TF going into the reservoir. It should be clear from the above description that, on each iteration, the heart of the event loop is primarily concerned with: placing a TF on the DNA; removing a TF from the DNA; or both. Therefore, identifying free sections on the DNA is a potential performance bottleneck that could prevent scaling of the method to realistic sizes of both DNA and numbers of TFs.

The memory model

The key to efficient implementation of binding and movement is the fast identification of available binding sites — i.e., not just empty bases but runs of bases that are at least as long as the binding motif (see eq 1) and can thus support binding of a TF. A naive representation of the DNA might be an array of Boolean values, one for each possible site, recording whether a site is currently occupied by a bound TF or not. In this implementation, an attempt to bind would involve the generation of a random number within the desired location range and a check as to whether that location is free or not. If it is not free then options might be: abandoning the attempt immediately; searching from that location in one or other direction until a free site is found; or identifying a fresh random location and repeating the process until a free site is found. While simple to implement, the weakness of this approach is immediately clear as the time to find a free location is dependent upon the occupancy of the DNA. Indeed, even when there are plenty of free individual bases, there are no guarantees that a long enough consecutive run will exist to allow a TF to bind, and the approach outlined above must ensure that a search in vain will ultimately terminate.

Using this scheme the time to locate a free binding site depends on the occupancy of the DNA, and scales poorly with the size of the genome. In this model we therefore use a different approach that can find binding sites within a time independent of the occupancy. Rather than an unstructured array of Boolean status values we maintain a data structure that records all the remaining bindable sections of the DNA, as $(position, length)$ pairs. The DNA is modeled as a 1D wrap-around structure. Note that because binding and unbinding occur at irregular intervals, sections of binding sites are occupied and freed according to no particular regular pattern. The resulting space management problem is akin to *dynamic storage allocation* in program runtime environments (Knuth (1997)), as opposed to *stack* (last-in, first-out) memory management, for instance. A significant difference, however, is that traditional allocation algorithms, such as *first fit* and *best fit*, are inapplicable in this context, because the memory manager must always allocate a particular section of free space that has been selected by the bind event, rather than having a free choice. In common with dynamic memory allocation, available space quickly becomes "fragmented". For instance, consider a run of $l+n$ unoccupied sites, where l is the length of a TF to be bound and $n >= l$ (Figure 2a). This sequences offers $n+1$ potential binding sites before a bind but anywhere between 0 and $n-l+1$ sites after the bind, depending on where the bind takes place within the run and the size of n in comparison to l. If the TF were to bind across the middle of the section then the two fragments either side may well be too short to support another TF (Figure 2b). As a result, the data structure recording bindable sections must be supplemented by a similar data structure recording unbindable fragments. For both we use the `set` associative container from the C++ STL (Meyers (2001)), which provides efficient access via its key which, in our case, is the binding position. Note that a fragment resulting from the bind of one TF may become usable before that particular TF unbinds — as a result of the earlier bound TF occupying the adjacent section at the other end of the fragment becoming unbound (Figure 2c). Indeed, most of the complexity of the memory management occurs during the bind-unbind cycle, at the point where a TF unbinds and the section it occupies becomes available again. Before being returned to the set of available sections it must be reunited with any fragments at either end. In addition, the newly freed section may now be contiguous with another already available section, in which case the two must be coalesced into one.

Methods

All simulations in this article were performed by starting with an empty wraparound DNA of length 4639675 at time

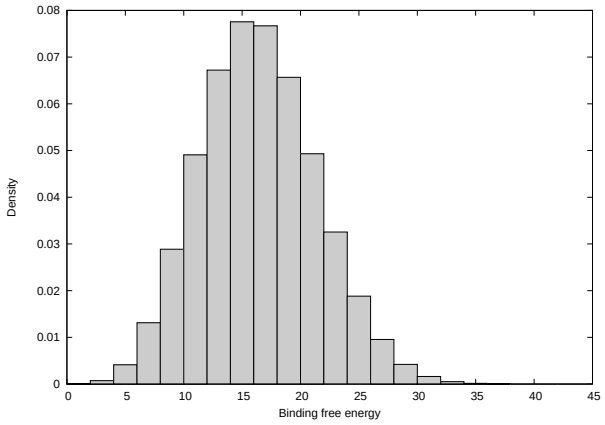

Figure 1: A histogram of the binding free energies as calculated from eq. 1. The energies are Gaussian distributed.

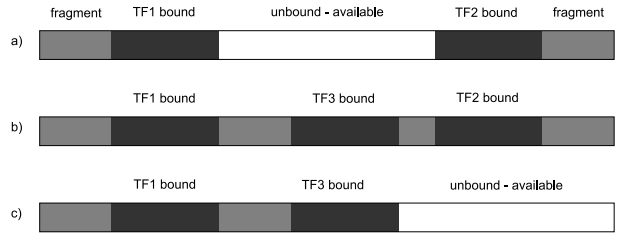

Figure 2: DNA section illustrating fragmentation and defragmentation during TF binding and unbinding: a) Two bound TFs, fragments and an available section; b) A third TF binds, resulting in two new fragments; c) A TF unbinds, fragments become available again.

0. Upon starting, the simulation protein was created with a rate of 0.01. The degradation rate of protein was 0.0009. Each simulation was run for a maximum of 10^9 time units, but was halted as soon as a TF was bound to the specific binding site at position 4540692 on the DNA. The halting time was taken as the mean first binding time (MFBT) referred to below. For each set of parameters the MFBT was calculated from 10 independent simulations (unless specified otherwise). In the graphs below, each point indicates the MFBT where the mean has been taken over the set of simulations that had been performed. Error-bars and standard deviation are not indicated in the graphs to preserve legibility. In nearly all experiments we performed, the standard deviation is comparable to the mean, indicating that typical binding times deviate significantly from the mean.

The source code of the program used here is available for free download.[1]

[1]via anonymous FTP from `ftp.cs.kent.ac.uk` as `pub/djb/exp/exp-distrib.tgz`

Results

One of the main variables to consider is the time the TF requires to reach its specific site. For a single random walker it is expected that MFBT scales with the square root of the distance. In the case of an ensemble walking this may be different. We decided to check this. To this end we performed a number of experiments with the following setup: We chose a synthesis site at which the TFs were produced. This has the effect that the TFs would attach at random to the binding site within a specified window. This introduces a stochastic element into the simulation, in the sense that not all TFs start from the same site. Some will start closer to the specific site, some from farther away. This choice has another effect. It limits the number of TFs that can attach to the DNA per time unit. The reason is that, upon binding to the DNA, TFs either occupy the binding sites within the initial binding window or they are released into the cytoplasm (represented by the "reservoir" in our model). If all sites within this window are occupied, no further TFs can bind and newly synthesised TF will always be released into the cytoplasm. We set the parameters such that no binding from the cytoplasm to the DNA is possible; hence, for the purpose of our simulation, once a TF unbinds from the DNA it is, in effect, lost forever. We found that the initial binding window is a strong restriction on the number of bound TFs.

We first performed a number of simulations with the initial binding window equal in size to the DNA. The effect of this is that newly created TFs will bind anywhere on the DNA. We allow the TFs to perform short moves of length up to 50 binding sites at a time; adjacent moves happen with a probability of 0. In this case we would predict that the MFBT is independent of the location of the synthesis site, but we would expect that the MFBT decreases as the TFs can travel faster, that is a higher short move length should lead to lower MFBTs. We varied both the probability of short moves and the site where TFs are synthesized. Figure 3 summarises the results of these simulations and confirms that the synthesis site is irrelevant, as expected. The graph shows the MFBT when all movements are only adjacent neighbor moves ($P = 1$), they are all short move events ($P = 0$) and an in-between case ($P = 0.8$). For other values of P we found that the MFBT always increases with increasing P. As can be seen from figure 3 the difference between the MFBTs for extreme cases of P are at the order of a magnitude.

In bacteria translation and transcription are closely interlinked. This means that protein tends to be made in close spatial proximity to the gene that codes for the particular protein. Following gene synthesis there is thus an increased chance that a TF binds to a particular local region of the DNA. We investigate the effect of this on the MFBT by varying the location of the initial binding window. Figure 4 shows a number of simulations with a window size of 40 (20 on each site of an assumed protein synthesis site). Such a

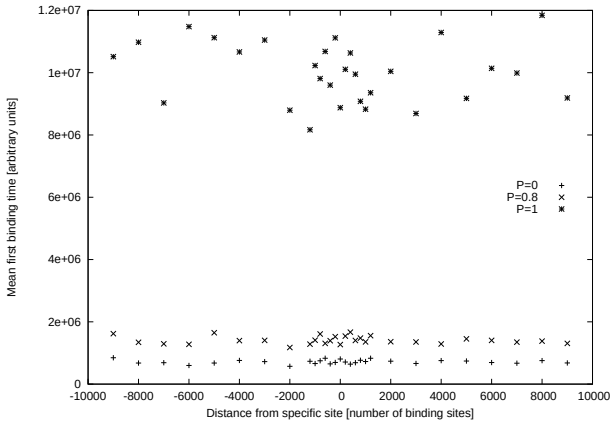

Figure 3: The mean first binding time to reach a particular specific site as a function of the short move distance. The window size equals the size entire genome. The short move length was set to 50 in these simulations.

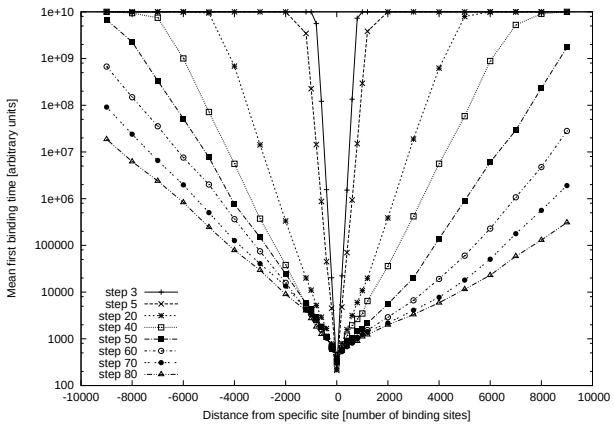

Figure 4: The mean first binding time to reach a particular specific site as a function of the short move distance. The short move probability was set to 0.9.

small preferred binding window is, admittedly, biologically unrealistic. However, it was chosen for practical considerations relating to the simulation speed. We found a strong dependence of the MFBT on the protein synthesis location as summarised in figure 4.

From these experiments it seems that a higher short move probability speeds up the search process. However, we would expect that the importance of this effect depends on the proximity of the synthesis site to the specific binding site. If the binding site is very close to the synthesis site, then one would conjecture that large step sizes will tend to "overshoot," that is they will simply miss the specific site during the movement. With larger initial distances this overshoot will happen as well, but TFs will move faster into the proximity of the specific site, hence counteracting this effect.

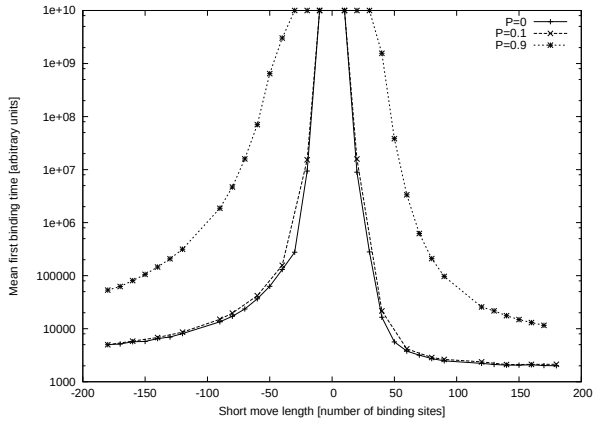

Figure 5: The mean first binding time to reach a particular specific site as a function of the short move distance. The short move probability was set to 0.9.

We performed a variant of the above experiments to understand this in more detail. The graph in figure 5 shows simulations where we kept the initial binding site fixed at an offset of ± 3000 binding sites from the specific site. The x-axis shows the short move length in the simulation and the sign of the x-axis indicates the centre of the initial binding site. So, for example the point marked at $x = -100$ represents a simulation with an offset of the initial binding site of -3000 from the specific binding site, and a short move length of 100. In these simulations each point represents the average MFBT over 1000 simulation experiments. The graph shows values for 3 different adjacent move probabilities, corresponding to all movements are short-moves, 90% of all events are short-move events and 10% of all move events are short move events.

The graph is somewhat complex to interpret, but shows that the MFBT falls faster than exponential with the short move length. For $P = 0$ and $P = 0.9$ the MFBT decreases by several orders of magnitude as the short move length increases from 20 to 100. When the short move length is smaller than 20, then irrespective of the value of P in the simulations shown here the MFBT is larger than the maximum simulation time of 10^9 time units.

A closer look at the simulation results, particularly at figure 5 reveals that the MFBT is asymmetric around the specific binding site. When the binding site is to the right of the specific site (i.e., higher id-numbers in the coordinate system used here), then the MFBT tends to be lower than when the TF is synthesised to the left. This effect is clearly illustrated in figure 5. Particularly for high short move length values there is a clear difference between the two synthesis sites. For example, when the short move length is 180, then for the parameters used in the figure the difference between the MFBTs amounts to nearly a factor of 2.

The underlying cause of the difference appears to be the

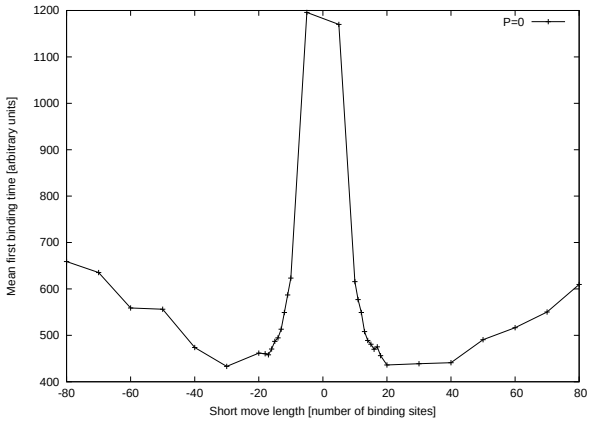

Figure 6: Same as figure 5, but with $P = 0$ and with an offset of ± 100.

presence of 3 further specific binding sites to the right of the focal site we are interested in. These three additional specific binding sites are in very close spatial proximity to the focal site with offsets of 18, 303 and 321 binding sites respectively. One can think of the dynamics as follows: When a TF binds to one of the 4 binding sites, then it acts as a reflecting boundary for random walkers in the area, confining random walkers within the area of the specific binding sites. This has the net effect of reducing the MFBT for the random walkers.

In figure 5 it appears that the longer the sliding distance, the shorter the MFBT. This is somewhat counter-intuitive. We would expect that there is an optimum sliding distance, which allows fast approach of the specific binding site, while balancing this with the problem of over-shooting the specific site. Within the short move distances considered in figure 5 such an optimum is not apparent. However, we would expect that such an optimum short move distance depends on the distance of the synthesis site from the specific site; the closer the synthesis site, the shorter the optimal short move distance. To check this we performed another set of experiments varying the short move distance, but with synthesis sites located at an offset of ± 100. Figure 6 shows the results. It is apparent that there is a clear minumum MFBT for both offsets, as expected.

Discussion and Conclusion

In this contribution we have presented a model that supports the efficient simulation of the process of TFs finding their specific binding sites. One of the problems that we had identified was that realistic simulations are computationally extremely demanding. For this reason, modeling of specific binding site localisation has been restricted to mathematically tractable but unrealistic models. Here we have made the first steps towards a computationally feasible implementation. One of the bottlenecks we have identified is the localisation of free binding sites on the DNA. By adapting approaches from dynamic memory allocation we were able to achieve speedups with respect to a naive algorithm of many orders of magnitude.

Apart from finding an efficient simulation implementation, we found that the MFBT depends in a complicated way on the short move distance, the synthesis site, but also the local configuration of the binding sites. Our simulations are a significant extension (although in simulation) to the analytical results developed by both Murugan and Mirny *et al*. The picture emerging from these simulations is that the situation is significantly more involved than suggested by these previous articles. For example: One of the conclusions by Murugan was that there is an optimal division between adjacent moves and short moves. We could not reproduce this in our setup. Instead we found that, up to the range we investigated, short moves are generally faster and more efficient than adjacent moves. We do not mean to imply that the their conclusions are wrong. However, it is clear that the conclusions of various models are not robust with respect to variations of underlying assumptions. This is normally a worrying sign in modelling.

This suggests that a more thorough investigation of this system is necessary, in order to come to a clear understanding of how previous mathematical results relate to the simulation results obtained here.

References

Barnes, D. and Chu, D. (2010). An efficient model for investigating specific site binding of transcription factors. In *Proceedings of the 4th International Conference on Bioinformatics and Biomedical Engineering, June 18-20, Chengdu, China, 2010*, page 4. IEE Xplore.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81(25):2340–2361.

Knuth, D. E. (1997). *Art of Computer Programming, Volume 1: Fundamental Algorithms (3rd Edition)*. Addison-Wesley Professional.

Meyers, S. (2001). *Effective STL: 50 Specific Ways to Improve the Use of the Standard Template Library*. Addison-Wesley Professional Computing Series. Addison Wesley.

Murugan, R. (2009). Packaging effects on site-specific dna-protein interactions. *Physical Review E*, 79(6 Pt 1):061920.

Slutsky, M., Kardar, M., and Mirny, L. (2004). Diffusion in correlated random potentials, with applications to DNA. *Physical Review E*, 69(6):061903.

The University of Wisconsin, M. (2009). *E. coli* genome project.

Winter, R., Berg, O., and von Hippel, P. (1981). Diffusion-driven mechanisms of protein translocation on nucleic acids. 3. the *escherichia coli lac* repressor–operator interaction: kinetic measurements and conclusions. *Biochemistry*, 20(24):6961–6977.

Wunderlich, Z. and Mirny, L. (2008). Spatial effects on the speed and reliability of protein-DNA search. *Nucleic Acids Research*, 36(11):3570–3578.