1 Kinetics of protein–DNA interaction

1.1 Reaction Kinetics

[1] The rate of change with time of the concentration of a protein–DNA complex is the sum of two terms. A positive contribution due to complex formation between a previously free specific site on a DNA molecule and a previously free protein, and a negative contribution due to complex break-up. At sufficiently low concentrations, the first term must be proportional to the probability of finding the a specific sites on the DNA molecule and a free protein molecule at the same site, and the second term must be proportional to the concentration of the complex:

$$\frac{d}{dt}[P|DNA] = k_a[P][DNA] - k_d[P|DNA]. \tag{1}$$

Let us apply this equation to *E. coli* bacteria and lac repressors introduced in previous lectures. In vitro experiments on repressor–DNA solutions (containing the operator target sequence) report that on-rate is $k_a \sim 10^{10}\mathrm{M}^{-1}\mathrm{s}^{-1}$ under standard conditions.

Suppose that at times $t < 0$ there are no repressor–DNA complexes because the concentration of lactose is high and repressors are in inactive state. At time $t = 0$, the lactose concentration drops to zero. How long will it take the activated lac repressors to locate the specific operator sequence and switch-off gene expression? There are only a few operator sequences per *E. coli*. Assuming a volume of $1\mu\mathrm{m}^3$, the (initial) concentration of unoccupied operator sequences $[DNA]$ is of order $1/\mu\mathrm{m}^3$ or about 10^{-9}M (concentrations are usually presented in molar units $\mathrm{M} = 6 \times 10^{26}\mathrm{m}^{-3}$). According to Eq. (1), for early times t, the concentration of occupied operator sequences will grow linearly in time as

$$\frac{d}{dt}[P|DNA] \approx (k_a[DNA])[P] = [P]/\tau, \tag{2}$$

where we have identified $\tau = 1/(k_a[P|DNA])$ as the characteristic time scale for a free repressor to locate the operator sequence. For the measured value of k_a, this search time is of order $\tau \sim 0.1$s.

1.2 Debye-Smoluchowski theory

In this section we will compute the on–rate k_a. The classical theory of the on–rate of diffusion–limited chemical reactions is due to Debye and Smoluchowski. Assume a spherical container (the cell) of radius R and place the specific target sequence at the center of the

[1] First two sections closely follow discussion in R. F. Bruinsma, Physica A **313**, 211-237 (2002)

container. Let $C(\vec{r}, t)$ be the concentration of free repressors. The concentration field obeys the diffusion equation

$$\frac{\partial C}{\partial t} = D_3 \nabla^2 C \tag{3}$$

with D_3 the diffusion constant of the protein in cytoplasm. We now want to know when the target sequence is occupied for the first time by a protein. Assume that this will happen when a diffusing protein enters for the first time a small sphere, of radius $b \ll R$, at the origin. Protein must find the exact target sequence, thus $b \approx 0.34$nm.

We will solve an easier problem by assuming that the small sphere at the origin acts as an absorber. Whenever a diffusing particle hits the small sphere, it disappears (free protein particle becomes bound protein-DNA complex). This approximation is valid, when specific sites on DNA are unoccupied. At the outer radius R we are constantly providing particles to keep concentration at a fixed value C_R. This is an easier problem because under these conditions, a time-independent steady-state current I is established of protein molecules diffusing from the outer to the inner sphere. To obtain this current, we must solve Laplaces law:

$$\nabla^2 C = 0 \tag{4}$$

with the boundary conditions $C(R) = C_R$ and $C(b) = 0$ (because diffusing particles disappear at $r = b$). Spherical symmetric solution is

$$C(r) = C_0 \left[1 - \frac{b}{r}\right], \tag{5}$$

where $C_0 = C_R/(1 - b/R) \approx C_R$ for $b \ll R$. The diffusion current density of proteins along the radial inward direction is $\vec{J} = -D_3 \nabla C = -D_3 b C_0/r^2 \ \hat{e_r}$, so the diffusion current I equals:

$$I = J(r) 4\pi r^2 = -4\pi D_3 b C_0 \tag{6}$$

Now compare this result with Eq. (2). The left-hand side of Eq. (2) is the number of complexes forming per second and must equal (minus) the incoming current I of free proteins. On the right-hand side we can identify C_0 with the free proteins concentration $[P]$ far from the operator. This leads to

$$k_a = 4\pi D_3 b \tag{7}$$

known as the Debye–Smoluchowski rate. If we use for the target radius base-pair distance $b \approx 0.34$nm and in vitro measured diffusion rate for lac repressors in water $D_3 \approx 3 \times 10^{-11} \mathrm{m^2 s^{-1}}$ we find that on-rate is of order $k_a \approx 10^8 M^{-1} s^{-1}$. We expect that actual on-rates are even smaller, because it takes certain time for protein to line up with the target and also diffusion constant D_3 is smaller in cytoplasm. Recall that for lac repressor measured on-rate $k_a \approx 10^{10} M^{-1} s^{-1}$ is two orders of magnitudes larger than the highest possible rate obtained by diffusion. This implies that proteins use some other mechanism to find specific site quickly.

1.3 Berg – von Hippel theory

In 1980s Berg and von Hippel proposed that proteins use combination of 1D (sliding) and 3D (jumps) diffusion to quickly find the target site on the DNA (Figure 1). Proteins are able

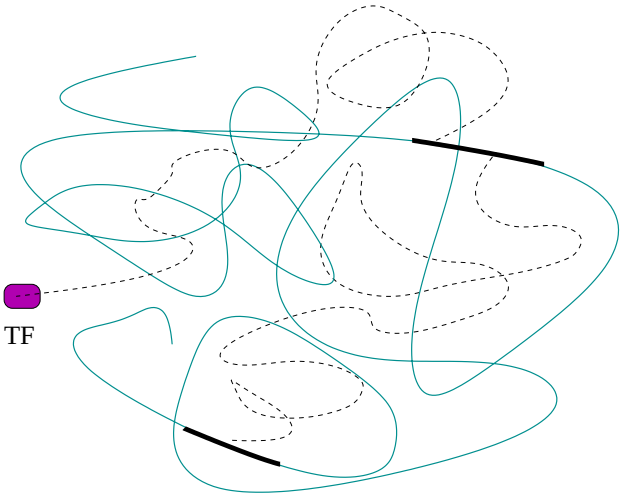

Figure 1: Schematics of 1D/3D search for target site on the DNA. Dashed lines represent 3D diffusion trajectories and thick lines are 1D sliding footprints.

to bind to any site on the DNA and then diffuse along the DNA (sliding). Once proteins detach from the DNA, they diffuse around until they attach to another site on the DNA (jump). Proteins keep sliding and jumping until they find the specific target site, where they get stuck because binding to the specific target site $E_s \sim 20 - 25 k_B T$ is a lot stronger than binding to a non-specific site $E_{\text{ns}} \sim 5 - 10 k_B T$.

During jumps it is reasonable to assume that protein can attach to any site on the DNA with equal probability, because DNA is in very compact form and even if two DNA segments are close in real space, they could be very far apart in the DNA sequence. Sliding events are thus independent and they start at uniformly distributed random positions along the DNA sequence. Probability that protein finds the target location in one sliding event is $q = n/M$, where n is number of visited sited during each sliding event and M is total number of sites on the DNA.

First we consider unrealistic case where every sliding event takes a fixed amount of time τ_1 and fixed number of sites n are visited by protein. The probability that single protein will find the target in exactly N_R rounds of sliding and jumping is $p(N_R) = q(1-q)^{N_R - 1}$, where the $1 - q$ factor reflects the probability that protein didn't find the target in first $N_R - 1$ rounds and factor q reflects the probability that protein has found the target in the last round. The average number of rounds needed for protein to find the target is:

$$\overline{N_R} = \sum_{N_R=1}^{\infty} N_R q (1-q)^{N_R - 1} = \frac{1}{q} = \frac{M}{n} \tag{8}$$

The average search time for protein to find the target site is:

$$\overline{t_s} = \overline{N_R}(\tau_1 + \tau_3) = \frac{M}{n}\ (\tau_1 + \tau_3), \tag{9}$$

where τ_3 is average time of 3D jump.

In reality sliding events don't take fixed amount of time. Protein detach from the DNA with rate $k_d^{(\mathrm{ns})} = 1/\overline{\tau_1}$ and time of each sliding event is taken from exponential distribution $\rho(\tau_1) = \exp(-\tau_1/\overline{\tau_1})/\overline{\tau_1}$. During sliding event every visited site is important, because protein is immediately trapped in specific site. Therefore we need to take the average distance between the leftmost and rightmost site to estimate the number of visited sites $n(\tau_1) = \sqrt{16 D_1 \tau_1/\pi b^2}$, where b is basepair distance, and not just the distance between the start and end site of the sliding, which would give $\sqrt{2 D_1 \tau_1/b^2}$. The average number of rounds needed for protein to find the target in this case is:

$$\langle \overline{N_R} \rangle = \sum_{N_R=1}^{\infty} N_R \left\langle q(\tau_{1,N_R}) \prod_{i=1}^{N_R-1} [1 - q(\tau_{1,i})] \right\rangle, \tag{10}$$

where $\tau_{1,i}$ is time protein spent during ith sliding event and bracket denotes averaging over all possible sliding times $\tau_{1,i}$. Sliding events are independent and this equation can be simplified:

$$\langle \overline{N_R} \rangle = \sum_{N_R=1}^{\infty} N_R \langle q \rangle [1 - \langle q \rangle]^{N_R - 1} = \frac{1}{\langle q \rangle} = \frac{Mb}{2\sqrt{D_1 \overline{\tau_1}}}, \tag{11}$$

where we used $\langle n \rangle = \int_0^\infty n(\tau_1)\rho(\tau_1) d\tau_1 = 2\sqrt{D_1 \overline{\tau_1}/b^2}$. Calculating average search time is a bit more complicated:

$$\begin{aligned}
\langle \overline{t_s} \rangle &= \sum_{N_R=1}^{\infty} N_R \left\langle \left(\sum_{i=1}^{N_R} [\tau_{1,i} + \tau_3] \right) q(\tau_{1,N_R}) \prod_{i=1}^{N_R-1} [1 - q(\tau_{1,i})] \right\rangle \\
\langle \overline{t_s} \rangle &= \langle N_R \rangle \tau_3 + \sum_{N_R=1}^{\infty} \left\{ (N_R - 1)\langle q \rangle (1 - \langle q \rangle)^{N_R - 2} [\langle \tau_1 \rangle - \langle q\tau_1 \rangle] + \langle q\tau_1 \rangle (1 - \langle q \rangle)^{N_R - 1} \right\} \\
\langle \overline{t_s} \rangle &= \langle N_R \rangle \tau_3 + \left\{ \langle N_R \rangle [\langle \tau_1 \rangle - \langle q\tau_1 \rangle] + \langle N_R \rangle \langle q\tau_1 \rangle \right\} \\
\langle \overline{t_s} \rangle &= \langle N_R \rangle \big(\langle \tau_1 \rangle + \tau_3 \big) = \frac{Mb}{2\sqrt{D_1 \overline{\tau_1}}} \big(\overline{\tau_1} + \tau_3 \big)
\end{aligned} \tag{12}$$

It is interesting to evaluate the optimal sliding time $\overline{\tau_1}^{(\mathrm{opt})}$ that minimizes the average search time.

$$0 = \frac{\partial \langle \overline{t_s} \rangle}{\partial \overline{\tau_1}} = \frac{Mb}{2\sqrt{D_1 \overline{\tau_1}}} \left(\frac{1}{2} - \frac{\tau_3}{2\overline{\tau_1}} \right) \qquad \Longrightarrow \qquad \overline{\tau_1}^{(\mathrm{opt})} = \tau_3 \tag{13}$$

For $\overline{\tau_1} > \overline{\tau_1}^{(\mathrm{opt})}$, protein spends too much time sliding. In the other case $\overline{\tau_1} < \overline{\tau_1}^{(\mathrm{opt})}$ is jumping a lot and spends too much time with 3D diffusion. Protein dissociation rate from the DNA

$k_d^{(\mathrm{ns})} = 1/\overline{\tau_1}$ strongly depends on the binding strength E_{ns} (homework), which depends on the salt concentration in the cytoplasm. Increasing the salt concentration reduces the non-specific binding energy E_{ns} since this interaction is predominantly electrostatic. One would expect that at standard physiological conditions $\overline{\tau_1}$ is close to the optimal value τ_3, but it turns out that protein spends more time sliding than jumping $\overline{\tau_1} > \tau_3$. Typical measured sliding time for lac repressor is $\overline{\tau_1} \sim 10^{-3}$s, while we can estimate the typical jumping time $\tau_3 \sim V/D_3 L \sim 10^{-4}$s, where $V \sim 1\mu\mathrm{m}^3$ is volume of the *E. coli*, $L = Mb \sim 1mm$ is DNA length and $D_3 \approx 3 \times 10^{-11}\mathrm{m}^2\mathrm{s}^{-1}$ measured diffusion constant in the cytoplasm. Using these results and measured diffusion constant $D_1 \approx 5 \times 10^{-14}\mathrm{m}^2\mathrm{s}^{-1}$ we can estimate the average search time:

$$\langle \overline{t_s} \rangle = \frac{L}{2\sqrt{D_1 \overline{\tau_1}}} \left(\overline{\tau_1} + \tau_3 \right) \sim 10 - 100\mathrm{s} \tag{14}$$

When n_p proteins are searching for target site simultaneously it is important to know the search time of the fastest of the n_p proteins, because once the first protein binds the target site, gene expression is shut down. Search time for a single protein to find the target site is exponentially distributed: $\rho_1(t_s) = \exp(-t_s/\langle t_s \rangle)/\langle t_s \rangle$. Distribution of search times for the fastest of the n_p proteins is obtained by standard extreme value distribution.

$$\rho_{n_p}(t_s) = n_p \rho_1(t_s) \left(1 - \int_0^{t_s} \rho_1(t) dt \right)^{n_p - 1} = \frac{n_p}{\langle t_s \rangle} \exp(-t_s n_p / \langle t_s \rangle) \tag{15}$$

The mean search time of the fastest of the n_p proteins to find the target location goes as $\langle t_s \rangle / n_p$. There could be of the order of $n_p \sim 100$ copies of proteins searching for target at the same time, which greatly speeds up the search time of proteins for the target site.