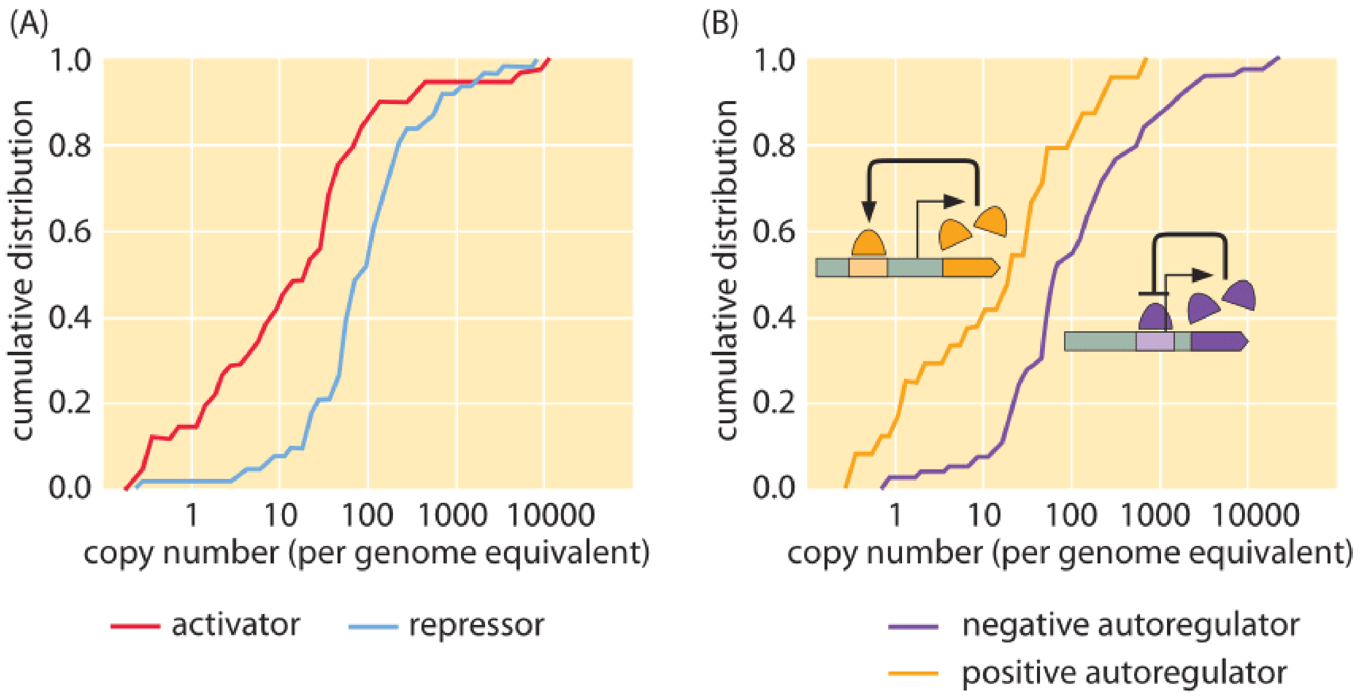
Gene regulation ¶

The expression of genes is tightly regulated in space and time.

- Transcription factors: proteins that bind to DNA and enhance or suppress expression
- Sigma factors (transcription initiation)
- Chromatin state: the genome can be packaged away and silenced
- DNA modifications: methylation, histone marks
- Protein modifications, dimerizations, etc

The molecular biology of these process is very complicated and we will focus here on general aspects of the problem.

How abundant are transcription factors?



Abundance of transcription factors in E.coli. From bionumbers.org with original data from [Li et al., 2014]. The data are shown as a cumulative distribution, that is the y-axis show the fraction of transcription factors that have an abundance below the value on the x-axis. Cumulative distributions might be a little unfamiliar to read, but have many advantages over classical histograms since they don't require a choice of binning.

- Activators: rare, often under 10 copies
- Repressors: more common typically 100 copies

Uniqueness of binding site

In a random string of ACGT of length one million, every 10-mer of bases will occur once on average. 10 bases is also the length of one turn of the DNA double helix and a transcription factor can meaningfully interact with about that many base pairs. Hence transcription factor binding in bacteria is approximately unique. This is reflected in the typical architecture where one, two, or sometimes three transcription factors regulate a gene or operon.

In eurkaryotes with 1000-fold larger genomes, TF binding is rarely unique. Instead, regulation is combinatorial with many layers contributing.

In [1]:

```
# Bacterium
print("\n\nBacterium\n")
L = 5e6
for k in range(2,13):
    print(f"number specific of {k}-mers in a genome of length {L}: {L/4**k:1.2f}
")

# human
print("\n\nHuman\n")
L = 3e9
for k in range(2,24,2):
    print(f"number specific of {k}-mers in a genome of length {L}: {L/4**k:1.2f}
")
```

```
Bacterium

number specific of 2-mers in a genome of length 5000000.0: 312500.0
0
number specific of 3-mers in a genome of length 5000000.0: 78125.00
number specific of 4-mers in a genome of length 5000000.0: 19531.25
number specific of 5-mers in a genome of length 5000000.0: 4882.81
number specific of 6-mers in a genome of length 5000000.0: 1220.70
number specific of 7-mers in a genome of length 5000000.0: 305.18
number specific of 8-mers in a genome of length 5000000.0: 76.29
number specific of 9-mers in a genome of length 5000000.0: 19.07
number specific of 10-mers in a genome of length 5000000.0: 4.77
number specific of 11-mers in a genome of length 5000000.0: 1.19
number specific of 12-mers in a genome of length 5000000.0: 0.30


Human

number specific of 2-mers in a genome of length 3000000000.0: 18750
0000.00
number specific of 4-mers in a genome of length 3000000000.0: 11718
750.00
number specific of 6-mers in a genome of length 3000000000.0: 73242
1.88
number specific of 8-mers in a genome of length 3000000000.0: 4577
6.37
number specific of 10-mers in a genome of length 3000000000.0: 286
1.02
number specific of 12-mers in a genome of length 3000000000.0: 178.
81
number specific of 14-mers in a genome of length 3000000000.0: 11.1
8
number specific of 16-mers in a genome of length 3000000000.0: 0.70
number specific of 18-mers in a genome of length 3000000000.0: 0.04
number specific of 20-mers in a genome of length 3000000000.0: 0.00
number specific of 22-mers in a genome of length 3000000000.0: 0.00
```

Transcription factor search

- need to find a binding site (typically one or a few) in a genome of millions of bases
- only a small number of TFs diffuse around in the cell.
- yet binding is very rapid and faster than expectation from the diffusion limit.

Diffusion limit

- $D_{TF} = 100 \mu m^2/s$
- $D_{DNA} \approx 0 \mu m^2/s$ (large molecule)
- length scale of TF/DNA interaction: $r = 0.3 \mu m$ (base pair height determines the accuracy on which TF and DNA have to interact)

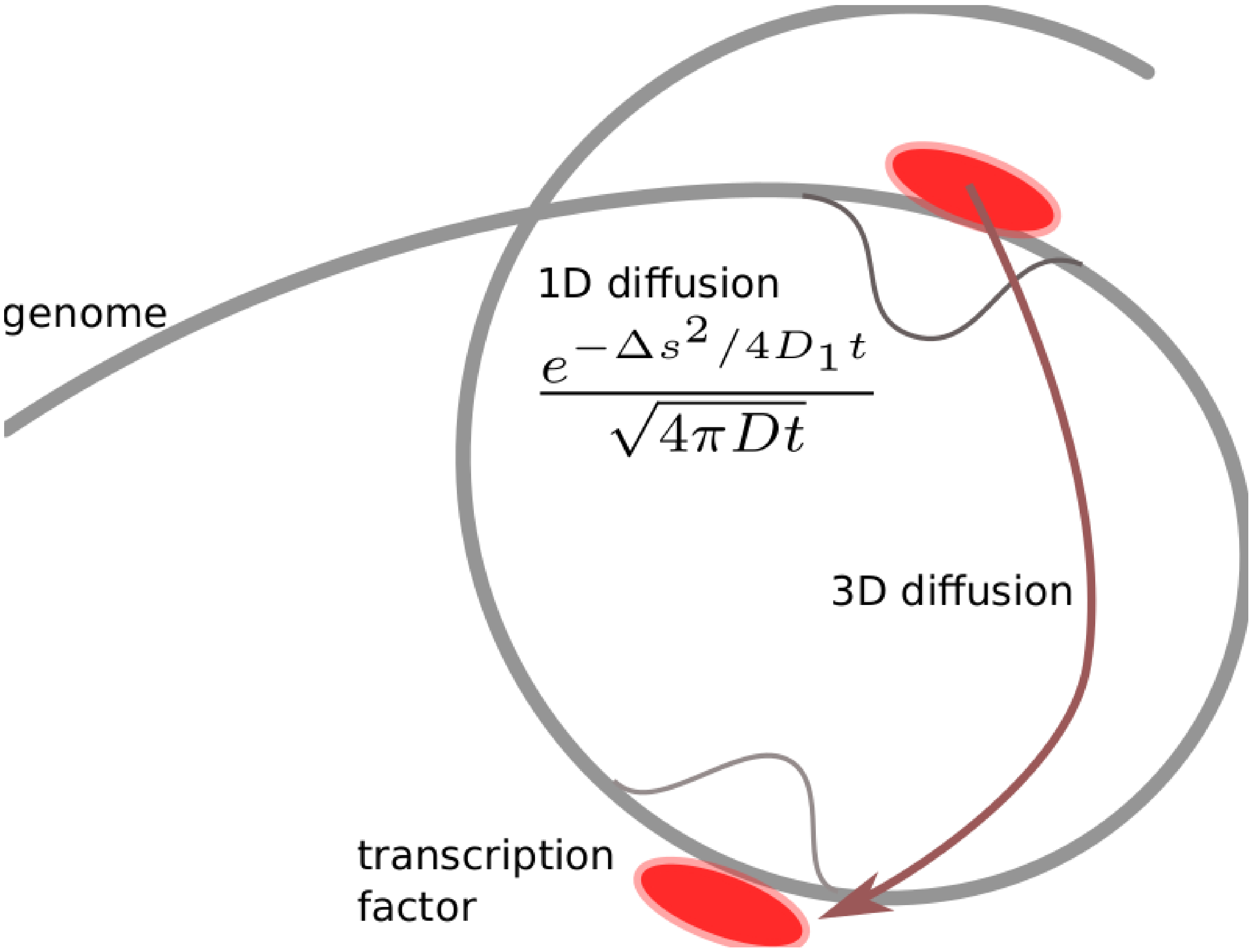
On-rate:

$$\kappa_D = 4\pi(D_{TF} + D_{DNA})r = 4\pi \times 100 \times 3 \times 10^{-4} \mu m^3/s \approx 0.4 \mu m^3/s \approx 2 \times 10^8 M^{-1} s^{-1}$$

Experimentally measured rate is 100 times larger!

Combined 1D and 3D search

A potential solution this conundrum was proposed by Hippel and Berg (http://www.jbc.org/content/264/2/675) and elaborated by Mirny et al (http://stacks.iop.org/1751-8121/42/i=43/a=434013): Combined 1D search along the genome and 3D search in the cytosol:



- 1D: don't waste time floating around placed where there is no DNA, but looking at the same place many times
- 3D: less redundant.

If combined 1D/3D diffusion was the mechanism by which TFs find their target, how should they be dividing their time between 3D and 1D search?

The total time until the target is found can be expressed as the sum over multiple rounds of 3D/1D search:

$$t_s = \sum_{i=1}^{K} (\tau_{1D,i} + \tau_{3D,i})$$

If l bases are searched in a round of 1D search, $\bar{K} = L/l$ rounds are necessary (L genome length).

Since $l \sim \sqrt{2D_{1D}\tau_{1D}}$, we obtain for the average search time

$$t_s = \frac{L}{\sqrt{2D_{1D}\tau_{1D}}}(\tau_{1D} + \tau_{3D})$$

The search time is minimal when

$$\frac{dt_s}{d\tau_{1D}} = \frac{L}{2\sqrt{2D_{1D}}}(\tau_{1D}^{-1/2} - \tau_{3D}\tau_{1D}^{-3/2}) = 0$$

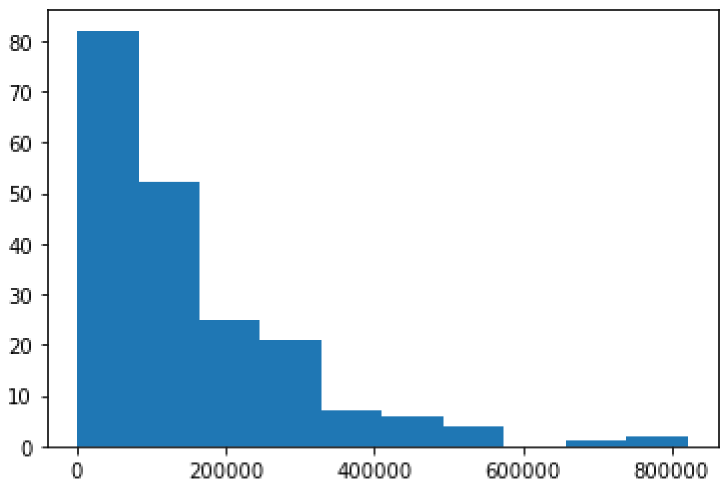
which requires $\tau_{1D} = \tau_{3D}$, i.e., the TF should spend equal times on the DNA and in solution. The mean search time is therefore

$$t_s = L\sqrt{\frac{2\tau_{1D}}{D_{1D}}}$$

In [2]:

```
import numpy as np
import matplotlib.pyplot as plt

# define the time spend on the DNA or in the cytosol
tau1D = 100
tau3D = 100
hop_off = 1/tau1D

L = 1e4
nmax = 200
# assume binding site is at position 0
target = 0

data = []
for i in range(nmax):
    # pick a random position
    x0 = np.random.randint(0,L)
    total_time = 0
    cycles = 0
    while x0!=target:
        p = np.random.random()
        if p<hop_off:
            # detach and reattach after time tau3D
            x0 = np.random.randint(0,L)
            total_time += tau3D
            cycles += 1
        else:
            # determine left or right hop
            if p>hop_off + (1-hop_off)*0.5:
                x0 += 1
            else:
                x0 -= 1
            # make sure the chromosome is a circle L+x -> x
            x0 = x0%L
            total_time += 1
    data.append((cycles, total_time))

data = np.array(data)
plt.hist(data[:,1])
mean_cycles, mean_time = data.mean(axis=0)
print(f"Average number of cycles: {mean_cycles:1.2f}, expected {L/np.sqrt(2*tau1
D):1.2f}")
print(f"Average time: {mean_time:1.2f}, expected {L/np.sqrt(2*tau1D)*(tau1D + ta
u3D):1.2f}")
```

```
Average number of cycles: 759.90, expected 707.11
Average time: 151221.62, expected 141421.36
```

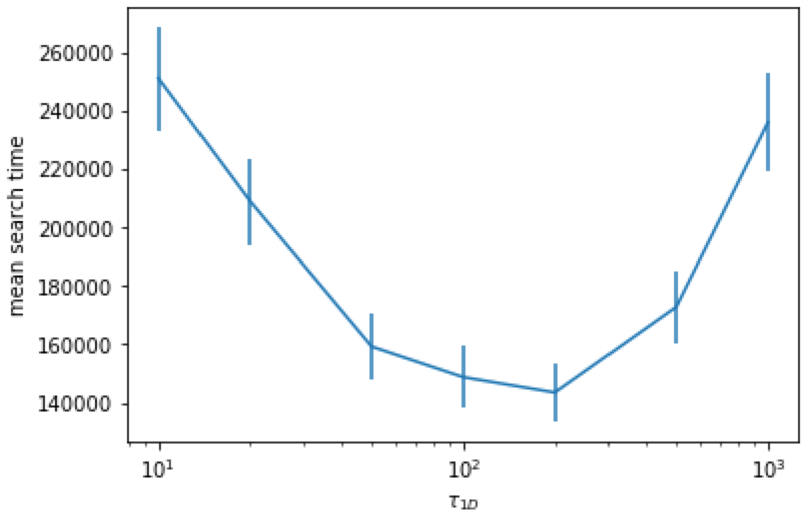


In [3]:

```
L = 1e4
nmax = 200
target = 0

tau3D = 100
mean_values = []
for tau1D in [10, 20,50, 100, 200, 500, 1000]:
    hop_off = 1/tau1D

    data = []
    for i in range(nmax):
        x0 = np.random.randint(0,L)
        total_time = 0
        cycles = 0
        while x0!=target:
            p = np.random.random()
            if p<hop_off:
                # detach and reattach after time tau3D
                x0 = np.random.randint(0,L)
                total_time += tau3D
                cycles += 1
            else:
                # determine left or right hop
                if p>hop_off + (1-hop_off)*0.5:
                    x0 += 1
                else:
                    x0 -= 1
                # make sure the chromosome is a circle L+x -> x
                x0 = x0%L
                total_time += 1
        data.append((cycles, total_time))

    data = np.array(data)
    mean_cycles, mean_time = data.mean(axis=0)
    stderr_cycles, stderr_time = data.mean(axis=0)/np.sqrt(nmax-1)
    mean_values.append([tau1D, mean_cycles, mean_time, stderr_cycles, stderr_tim
e])

mean_values = np.array(mean_values)
```

In [4]:

```
plt.errorbar(mean_values[:,0], mean_values[:,2], mean_values[:,4])
plt.xlabel(r'$\tau_{1D}$')
plt.ylabel('mean search time')
plt.xscale('log')
```



- a combination of one dimensional and three dimensional diffusion can accellerate the search process.
- predicts an optimum of 50/50 1D and 3D search that minimizes local oversampling and time "lost" in the cytosol
- reality is more complicated and additional process likely help binding site search.

Summary

- In bacterial gene regulation, a small number of molecules needs to find target sites -- a noisy process
- Activators tend to be at lower numbers than repressors
- Binding site search is surprisingly fast -- achieved by combined 1D and 3D diffusion