EXPLORING SINGLE-MOLECULE HETEROGENEITY AND THE PRICE OF CELL SIGNALING

by

TENGLONG WANG

Submitted in partial fulfillment of the requirements

For the degree of Doctor of Philosophy

Thesis Adviser: Prof. Michael Hinczewski

Department of Physics

CASE WESTERN RESERVE UNIVERSITY

January, 2022

Case Western Reserve University
Case School of Graduate Studies

We hereby approve the thesis[1] of

TENGLONG WANG

for the degree of

Doctor of Philosophy

Prof. Michael Hinczewski

Committee Chair, Advisor — Date
Department of Physics

Prof. Lydia Kisley

Committee Member — Date
Department of Physics

Prof. Harsh Mathur

Committee Member — Date
Department of Physics

Prof. Peter Thomas

Committee Member — Date
Department of Mathematics, Applied Mathematics, and Statistics

Date of Defense

October 6, 2020

[1]We certify that written approval has been obtained for any proprietary material contained therein.

Dedicated to my beloved grandparents:
Jin-Xiu & Gui-Zhen
Thank you for everything

Table of Contents

List of Tables

List of Figures

Acknowledgements

Without support from many people, this thesis would not have a chance to exist. I am particularly grateful to my advisor, Michael Hinczewski, for being not only a considerate mentor but also an excellent friend, for leading me into the field of theoretical biophysics, for tremendous freedom I have enjoyed during my time at Case, and for his guidance, patience, and inspiration, in research as well as in life. He is supportive, easygoing, and open-minded. So many times, when I have doubts about my work and myself, he is the one who gives positive support. His talks with me, academic or not, constantly shed insights on how to realize the world. His enthusiasm for science and life encourage me to broaden my horizons through studying in unfamiliar areas. No doubt that Mike showed me what a scientist should look like, how we should treat people equally and friendly, as well as how science should be studied objectively without bias. If I have a chance to become a faculty in the future, he would definitely be my role model.

I would like to express my sincere gratitude to Prof. GuangRi Jin for his guidance, training, and mentorship during my master years at BJTU. His faith in me, a graduate with bachelor degree of management, and his offer—a chance to work with him—initialized my academic adventure in physics. I would also like to thank Prof. Wen Yang for his constructive advice and support during my research in master years.

For everyone I have interacted with in the Hinczewski lab, I am grateful for their time, expertise and kindness. Through countless discussions with Shishir, I learned a lot about the history of Nepal, culture, politics, camping, tech gadgets, and the frontier of science. His proposal of a self-study machine learning reading course with me led to our research project described in Chapter 3. Ben and I worked together quite a while on

the project described in Chapter 2, his enthusiasm and joy in science impressed me. I also learned a lot about different countries' culture, history, politics, food and so on from Casey, Efe, Shamreen, Brandon, Joshua, Niksa and Prof. Alkan Kabakcioglu.

I would also like to thank Profs. Lydia Kisley, Harsh Mathur, and Peter Thomas for being my thesis committee.

I have enjoyed spending time with friends. First year at Case, the friendship from Marcio, Michael, Klaountia, and Saurabh helped me to settle in rapidly. I have been fortunate to have Chujun, Marcio, Haixiang (aka Feixiang) as roommates and close friends. I wish to show my gratitude to Xiaobin for her support, countless talks and sending me a machine learning textbook that contributed a lot to my thesis. I would like to thank my homies Chong-Yang, Tantai, Jay (aka JMP), Jun (aka Heart-protecting Jun), Rui-Jiao (aka A Jiao), Ke (aka Si-Mai-Jie), Feng-Long (aka Boss Jia) for being inclusive, communicative, trustworthy, loyal, patient and consistent for me.

I would like to pay my special regards to Einstein, Hawking, Feynman, Dirac, Lu Xun, Franz Kafka, Isaac Asimov, and Stephen Chow whose works helped me to set milestones and taught me how to think, write, fantasize and laugh as who I am today. The countless authors (e.g. Vernor Vinge, Robert J. Sawyer, Arthur C. Clark, Robert A. Heinlein, George R. R. Martin, Cixin Liu, Cuttlefish That Loves Diving, The Fat Loss Expert), directors (e.g. Wen Jiang, Johnnie To, Wong Kar-Wai, Park Chan-wook, Takeshi Kitano, Stanley Kubrick, David Lynch, Martin Scorsese, Christopher Nolan), TV/Anime/Manga/Game maker, and musician whose works keep me in peace, encouraged and struggling in the adventure of life. Because of them, life is less boring.

Last and foremost, I had the good fortune to be raised in a loving and supportive family full of laughter. I would like to thank my family, for meticulous care and understanding, for creating an open-minded atmosphere that respects independent will of every member, and for providing fertile ground for fantasy, curiosity, and freedom of thought. Because of them, I have the motivation, purpose and power to complete my studies and chase my dreams.

Abstract

Exploring Single-molecule Heterogeneity and the Price of Cell Signaling

Abstract

by

TENGLONG WANG

In the last two decades, advances in experimental techniques have opened up new vistas for understanding bio-molecules and their complex networks of interactions in the cell. In this thesis, we use theoretical modeling and machine learning to explore two surprising aspects that have been revealed by recent experiments: (i) the discovery that many different types of cellular signaling networks, in both prokaryotes and eukaryotes, can transmit at most 1 to 3 bits of information; (ii) the observation that single bio-molecules can exhibit multiple, stable conformational states with extremely heterogeneous functional properties.

The first part of the thesis investigates how the energetic costs of signaling in biological networks constrain the amount of information that can be transferred through them. The focus is specifically on the kinase-phosphatase enzymatic network, one of the basic elements of cellular signaling pathways. We find a remarkably simple analytical relationship for the minimum rate of ATP consumption necessary to achieve a certain signal fidelity across a range of frequencies. This defines a fundamental performance limit for such enzymatic systems, and we find evidence that a component of the yeast osmotic shock pathway may be close to this optimality line. By quantifying

the evolutionary pressures that operate on these networks, we argue that this is not a coincidence: natural selection is capable of pushing signaling systems toward optimality, particularly in unicellular organisms. Our theoretical framework is directly verifiable using existing experimental techniques, and predicts that many more examples of such optimality should exist in nature.

In the second part of the thesis, we develop two machine learning methods to analyze data from single-molecule AFM pulling experiments: a supervised (deep learning) and an unsupervised (non-parametric Bayesian) algorithm. These experiments involve applying an increasing force on a bio-molecule or bio-molecular complex until it unfolds or ruptures. The distribution of times it takes for this unfolding/rupture to occur, collected from many repetitions of the experiment, contains signatures of heterogeneity: information about the number and properties of the different conformational states that exist in a given system. We show that both machine learning techniques can effectively tease out this information, though each has its own strengths and weaknesses. The algorithms are validated on a large set of synthetic data, generated to mimic the wide range of biological parameters and experimental settings one would encounter in real-world applications.

1 Introduction

This thesis concerns two topics—cell signaling and single bio-molecule heterogeneity—that are both considered crucial for the biological function of living cells. Cell signaling is essential for collecting information about the environment, while heterogeneity enables this information to be reflected in different functional conformations of a bio-molecule: for example environmental conditions can "anneal" a bio-molecule, allowing it to switch rapidly between conformations [1], or alternatively favor long-lived states.

In Chapter 2, we will explore the price of information transfer in living cells by analyzing a canonical signaling circuit: the enzymatic push-pull loop. In Chapter 3, we will explore single-molecule heterogeneity using two different machine learning techniques. Before going into the details of these two topics, let us first introduce some general ideas, quantities, and useful techniques from biological information theory and machine learning.

1.1 Cellular signaling

In focusing on how information is transferred in living cells, we will first introduce several important properties through the theoretical framework of a simple input-output signaling network. This network is a coarse-grained version of the full enzymatic model

we introduce in Chapter 2, and hence less suitable for direct experimental comparisons. But it is simple enough to provide a convenient introduction to the ideas we will explore in more depth later on.

1.1.1 Modeling a simple input-output signaling network

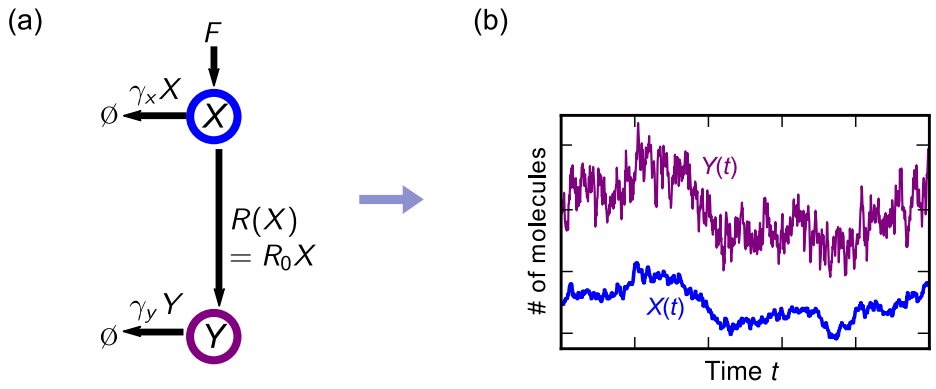

Figure 1.1. (a) A simple signaling circuit, involving an input species $X(t)$ and output species $Y(t)$, related by the production rate R_0. (b) Both input $X(t)$ and output $Y(t)$ vary as function of time. The simple signaling circuit can be treated as an amplifier, transducing an input signal into an amplified output signal.

The complexity of cell signaling can be daunting: for instance, there are over 500 protein kinases operating in many interconnected pathways just inside humans [2]. To obtain a better understanding, we start with a simple signaling network that illustrates the general theoretical approach. This sets up a foundation for studying more realistic biological signaling pathways. Consider the simple signaling pathway shown in Fig. 1.1, including only two chemical species: the input species $X(t)$ and the output species $Y(t)$. For example, the input and output species can be interpreted as the active and phosphorylated forms of two protein kinases. As shown in the figure, the upstream part of the pathway is described by an effective production rate F for input species X. The output species Y is produced by given input X with a production rate R_0. The input and output

have decay rate γ_x and γ_y respectively. In general, all the rates F, R_0, γ_x and γ_y can be time dependent (influenced by a fluctuating extracellular and intracellular conditions), but we assume they are constant here for simplicity. This signaling network can function as an amplifier, transducing an input signal into an amplified output signal.

1.1.2 Auto-correlation time and frequency of the input

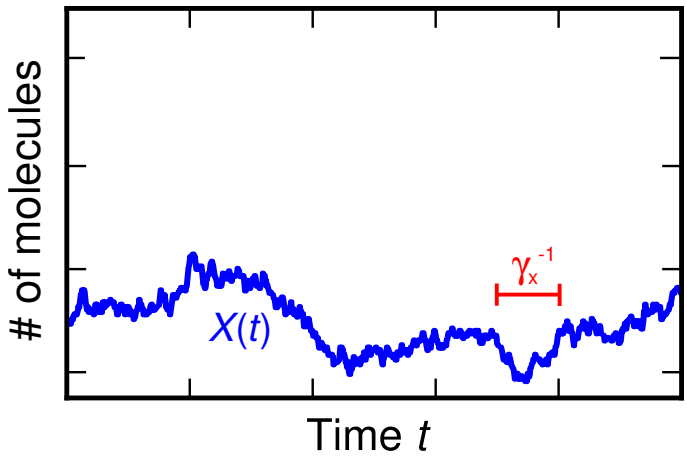

Figure 1.2. The autocorrelation time of the input, γ_x^{-1}, can be interpreted as the characteristic timescale of the input fluctuations, and we will denote its inverse, γ_x, as the effective "frequency" of the input.

To quantitatively measure the information transfer from $X(t)$ to $Y(t)$, several properties of the system are crucial to our analysis. The first is the auto-correlation time τ_a of the input, defined through the auto-correlation function

$$\overline{\delta X(t+\tau)\delta X(t)} = \overline{\delta X^2}\exp(-|\tau|/\tau_a), \tag{1.1}$$

where the bar denotes an average over an ensemble of trajectories in the stationary state and $\delta X(t) \equiv X(t) - \overline{X}$. For our simple model $\tau_a = \gamma_x^{-1}$, the inverse of the decay rate γ_x. Note, in the stationary state, that instantaneous averages like $\overline{X} \equiv \overline{X(t)}$ and $\overline{\delta X^2} \equiv \overline{\delta X^2(t)}$ are independent of t. Since γ_x^{-1} is the characteristic timescale of the input fluctuations,

we will denote its inverse, γ_x, as the effective "frequency" of the input. Similarly, the output decay rate sets the response time scale γ_y^{-1} over which $Y(t)$ can react to changes in the input.

1.1.3 Chemical potential

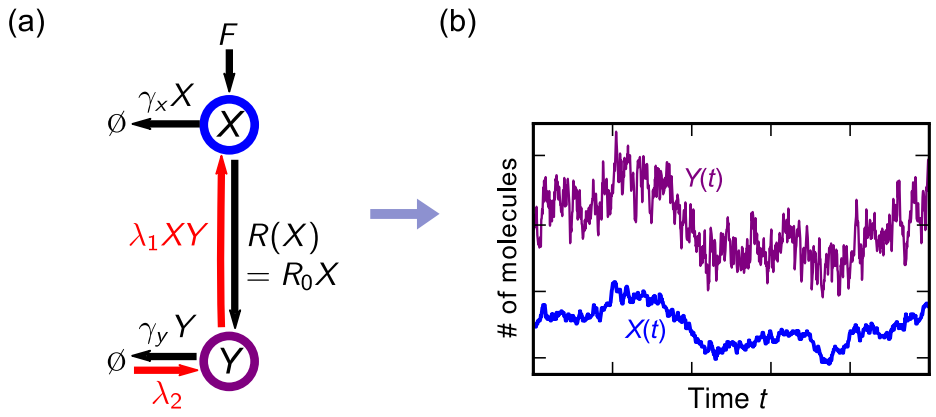

Figure 1.3. (a) A simple signaling circuit with reverse reactions (red arrows), involving an input species $X(t)$ and output species $Y(t)$. In the limit of "irreversible" enzymatic reactions, $\lambda_1 \to 0$, $\lambda_2 \to 0$, the system reduced to the case Fig. 1.1(a). (b) Both input $X(t)$ and output $Y(t)$ varies as function of time. The simple signaling circuit can be treated as an amplifier, transducing an input signal into an amplified output signal.

In reality, the input-output system described in Fig. 1.1 is incomplete, since the existence of reverse reactions are inevitable. Though the reverse rates are typically negligible under cellular conditions, they need to be considered to make our system thermodynamically consistent. Fig. 1.3 shows our model with reverse reactions. The rationale for these reactions can be understood by looking at a more detailed biochemical description: imagine that the underlying system was a kinase-phosphatase signaling system where X represents the kinase population. A substrate protein is phosphorylated to give an output Y (population of phosphorylated substrates) with a pseudo-first-order rate $R_0 X$ (the R_0 here depends on the substrate population). The reverse reaction in this

context would be the phosphorylated substrate rebinding to the kinase and the phosphate group being removed, with some rate $\lambda_1 XY$ that depends on both kinase X and phosphorylated substrate Y populations. Of course the main way dephosphorylation occurs is not through such unlikely reversal events but through a phosphatase enzyme binding to the phosphorylated substrate, with pseudo-first-order rate $\gamma_y Y$ (the γ_y depends on the phosphatase population). Reversal of the phosphatase-catalyzed reaction, which depends on the substrate rebinding to the phosphatase and getting back a phosphate group, is reflected in the rate λ_2 (with depends implicitly on both substrate and phosphatase populations, neither of which explicitly appears in the simple model). In the limit of "irreversible" enzymatic reactions, $\lambda_1 \to 0$, $\lambda_2 \to 0$, the system reduces to the case described by Fig. 1.1.

Generally a biological signaling network requires consumption of some metabolic "fuel", our simple input-output system is no exception. In order to make this signaling pathway transfer information effectively, the symmetry of the network should be broken, which means the forward rate direction (input to output $X \to Y$ direction) of the pathway should be preferred. So, what fuel ensures this preference? It is typically the chemical potential $\Delta\mu$ associated with ATP hydrolysis. ATP is constantly replenished from metabolic processing of nutrients to maintain sufficiently high chemical potential to drive the $X \to Y$ current forward. Mathematically, we can link the product of the ratios of the reverse rates relative to the forward ones and the chemical potential through a key thermodynamic relation arising from the principle of detailed balance [3, 4],

$$e^{-\beta\Delta\mu} = \frac{\lambda_1 \lambda_2}{\gamma_y R_0}, \tag{1.2}$$

where $\beta = (k_B T)^{-1}$, k_B is the Boltzmann constant, and T is the temperature. As long as $\Delta\mu$ is large and positive, the strong preference for the forward reaction direction is guaranteed. Moreover, every forward traversal ($X \to Y$) phosphorylates a substrate, in the process consuming a single ATP molecule through hydrolysis, releasing the products ADP and inorganic phosphate P_i back into the surroundings. $\Delta\mu$ depends on the concentrations [ATP], [ADP], and [P_i] through

$$\Delta\mu = \Delta\mu_0 + k_B T \ln \frac{[\mathrm{ATP}]\,(1\,\mathrm{M})}{[\mathrm{ADP}]\,[\mathrm{P_i}]}$$

where $\Delta\mu_0$ is the standard free energy of ATP hydrolysis ($\Delta\mu_0 \approx 12\ k_B T$ at room temperature [5]).

1.1.4 Gain parameter and ATP consumption

Another property we are interested in is the gain parameter: how much output is produced on average for each input molecule. For the simple signaling network here, it is obviously given by the production rate R_0. With the help of gain parameter R_0, we can quantify the average rate of ATP consumption. In stationary state, the average rate of ATP consumption is just the mean rate of the forward reaction step as $A = R_0\overline{X}$, if we assume that one ATP is consumed per reaction.

1.1.5 Mutual information

The last property of interest is a quantitative measure of the information transfer, given by the instantaneous stationary mutual information I between $X(t)$ and $Y(t)$. This is defined in terms of the joint probability $P(X,Y)$ of observing input value X and output value Y at the same moment of time, and the corresponding marginal probabilities $P(X)$

and $P(Y)$,

$$I = \sum_{X,Y} P(X,Y) \log_2 \frac{P(X,Y)}{P(X)P(Y)}. \tag{1.3}$$

The mutual information $I \geq 0$ in all cases, and is measured in bits, with larger values translating to a greater degree of correlation between input and output. In our model at larger population sizes, the marginal and joint distributions can be approximated as Gaussian:

$$P(X) = \frac{e^{\frac{-(X-\overline{X})^2}{2\sigma_x^2}}}{\sqrt{2\pi}\sigma_x}, \tag{1.4}$$

$$P(Y) = \frac{e^{\frac{-(Y-\overline{Y})^2}{2\sigma_y^2}}}{\sqrt{2\pi}\sigma_y}, \tag{1.5}$$

$$P(X,Y) = \frac{e^{-\frac{1}{2(1-\rho^2)}\left[\frac{(X-\overline{X})^2}{2\sigma_x^2} + \frac{(Y-\overline{Y})^2}{2\sigma_y^2} - \frac{2\rho(X-\overline{X})(Y-\overline{Y})}{\sigma_x\sigma_y}\right]}}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}}, \tag{1.6}$$

where $\overline{X}$ and $\overline{Y}$ are mean populations, $\sigma_x = \sqrt{\langle X^2\rangle - \langle X\rangle^2}$ and $\sigma_y = \sqrt{\langle Y^2\rangle - \langle Y\rangle^2}$ are standard deviations, and $\rho = (\langle XY\rangle - \langle X\rangle\langle Y\rangle)/\sigma_x\sigma_y$ is the Pearson correlation coefficient. This allows us to use a expression for I valid in this limit that is more convenient to evaluate [6]:

$$\begin{aligned} I &\approx -\frac{1}{2}\log_2 E, \\ \text{where} \quad E &\equiv 1 - \frac{\mathrm{cov}(XY)^2}{\mathrm{var}(X)\mathrm{var}(Y)} \\ &\equiv 1 - \frac{(\overline{XY} - \overline{X}\,\overline{Y})^2}{\left(\overline{X^2} - \overline{X}^2\right)\left(\overline{Y^2} - \overline{Y}^2\right)}. \end{aligned} \tag{1.7}$$

Here $E = 1 - \rho^2$ lies in the range $0 \leq E \leq 1$. For $E = 0$ (or equivalently $I = \infty$) we have perfect correlation between the input and output signal, while $E = 1$ ($I = 0$) corresponds to an output that is completely independent of the input.

1.1.6 Chemical Langevin solution

To derive analytical results for the above system, we need to mathematically describe the time evolution of the species in the system. A stochastic method that captures this time evolution at larger population sizes (where $X(t)$ and $Y(t)$ can be treated as continuous variables) is the linearized chemical Langevin approach [7]. For the above input-output system described in Fig. 1.3, the chemical Langevin equations are

$$\begin{aligned} \frac{dX}{dt} &= F - \gamma_x X + n_x, \\ \frac{dY}{dt} &= R_0 X + \lambda_2 - \lambda_1 XY - Y\gamma_y + n_y, \end{aligned} \tag{1.8}$$

where the noise term $n_i(t) = \sqrt{\Pi_i}\eta_i(t)$. The noise terms are associated with reactions in the system, and the corresponding prefactors represent the sum of the mean production(forward) and deactivation/unbinding (backward) contributions to each reaction, $\Pi_x = 2\overline{X}\gamma_x$ and $\Pi_y = 2(\overline{X}R_0 + \lambda_2)$. The Gaussian white noise functions η_i have correlations $\langle \eta_i(t)\eta_j(t')\rangle = \delta_{ij}(t-t')$. Considering the above equations, it is easy to obtain the stable-state solution as

$$\overline{X} = \langle X\rangle = \frac{F}{\gamma_x}, \qquad \overline{Y} = \langle Y\rangle = \frac{FR_0 + \gamma_x\lambda_2}{\gamma_x\gamma_y + F\lambda_1}. \tag{1.9}$$

Plugging in $X = \overline{X} + \delta X$ and $Y = \overline{Y} + \delta Y$, converting to Fourier space and linearizing, Eq. (1.8) becomes

$$\begin{aligned} \tilde{n}_x(\omega) + \delta\tilde{X}(\omega)(i\omega - \gamma_x) &= 0, \\ \tilde{n}_y(\omega) + \delta\tilde{X}(\omega)(R_0 - \overline{Y}\lambda_1) + \delta\tilde{Y}(\omega)(-\gamma_y - \overline{X}\lambda_1 + i\omega) &= 0, \end{aligned} \tag{1.10}$$

where the tilde indicates a Fourier-transformed function. The solutions of above equations (1.10) take the form of linear combination of the noise functions, $\sum_{i=x,y} a_i(\omega)\tilde{n}_i(\omega)$,

as follows:

$$\begin{aligned}
\delta\tilde{X}(\omega) &= \frac{1}{\gamma_x - i\omega}\tilde{n}_x, \\
\delta\tilde{Y}(\omega) &= \frac{i(R_0 - \overline{Y}\lambda_1)}{(\gamma_y + \overline{X}\lambda_1 - i\omega)(i\gamma_x + \omega)}\tilde{n}_x + \frac{(i\gamma_x + \omega)}{(\gamma_y + \overline{X}\lambda_1 - i\omega)(i\gamma_x + \omega)}\tilde{n}_y.
\end{aligned} \tag{1.11}$$

The corresponding input, output and cross power spectra—$P_X(\omega)$, $P_Y(\omega)$, and $P_{XY}(\omega)$ respectively—are defined through:

$$\begin{aligned}
&\overline{\delta\tilde{X}(\omega)\delta\tilde{X}(\omega')} = 2\pi P_X(\omega)\delta(\omega+\omega'), \quad \overline{\delta\tilde{Y}(\omega)\delta\tilde{Y}(\omega')} = 2\pi P_Y(\omega)\delta(\omega+\omega'), \\
&\overline{\delta\tilde{X}(\omega)\delta\tilde{Y}(\omega')} = 2\pi P_{XY}(\omega)\delta(\omega+\omega').
\end{aligned} \tag{1.12}$$

These can be easily obtained from Eq. (1.11) and the correlation properties of the noise terms:

$$\begin{aligned}
P_{XX}(\omega) &= \frac{1}{\gamma_x^2+\omega^2}\Pi_x = \frac{2F}{\gamma_x^2+\omega^2}, \\
P_{YY}(\omega) &= \frac{(R_0 - \overline{Y}\lambda_1)^2}{(\gamma_x^2+\omega^2)(\gamma_y^2 + 2\overline{X}\lambda_1\gamma_y + \overline{X}^2\lambda_1^2 + \omega^2)}\Pi_x \\
&\quad + \frac{1}{\gamma_y^2 + 2\overline{X}\lambda_1\gamma_y + \overline{X}^2\lambda_1^2 + \omega^2}\Pi_y \\
&= \frac{N_0 + N_1\omega^2}{D_0 + D_1\omega^2 + D_2\omega^4}, \\
\mathrm{Re}\, P_{XY}(\omega) &= \frac{2F(\gamma_y + \overline{X}\lambda_1)(R_0 - \overline{Y}\lambda_1)}{(\gamma_x^2+\omega^2)(\gamma_y^2 + 2\overline{X}\lambda_1\gamma_y + \overline{X}^2\lambda_1^2 + \omega^2)},
\end{aligned} \tag{1.13}$$

where the coefficients N_i (D_i) for the ω^{2i} terms in the numerator (denominator) of $P_{YY}(\omega)$ are

$$\begin{aligned}
N_0 &= 2\gamma_x^3\{FR_0[R_0\gamma_x\gamma_y^2+(\gamma_x\gamma_y+F\lambda_1)^2]+\gamma_x[\gamma_x^2\gamma_y^2+2F\gamma_x\gamma_y\lambda_1 \\
&+ F\lambda_1(F\lambda_1-2R_0\gamma_0)]\lambda_2+F\gamma_x\lambda_1^2\lambda_2^2\}, \\
N_1 &= 2\gamma_x(\gamma_x\gamma_y+F\lambda_1)^2(FR_0+\gamma_x\lambda_2), \\
D_0 &= \gamma_x^2(\gamma_x\gamma_y+F\lambda_1)^4, \\
D_1 &= (\gamma_x\gamma_y+F\lambda_1)^2[\gamma_x^4+(\gamma_x\gamma_y+F\lambda_1)^2], \\
D_2 &= \gamma_x^2(\gamma_x\gamma_y+F\lambda_1)^2. \qquad (1.14)
\end{aligned}$$

With the help of the above power spectra, we can obtain the variances and covariances as

$$\begin{aligned}
\mathrm{var}(X) &= \langle X^2\rangle-\langle X\rangle^2=\frac{1}{2\pi}\int_{-\infty}^{\infty}P_{XX}(\omega)d\omega, \\
\mathrm{var}(Y) &= \langle Y^2\rangle-\langle Y\rangle^2=\frac{1}{2\pi}\int_{-\infty}^{\infty}P_{YY}(\omega)d\omega, \\
\mathrm{cov}(XY) &= \langle XY\rangle-\langle X\rangle\langle Y\rangle=\frac{1}{2\pi}\int_{-\infty}^{\infty}P_{XY}(\omega)d\omega. \qquad (1.15)
\end{aligned}$$

Note that in the last line of Eq. (1.13), we only write down the real part of $P_{XY}(\omega)$, since the imaginary part will cancel out in above integral. After some tedious calculation, the

integrals in Eq. (1.15) can be explicitly evaluated to yield

$$\begin{aligned}
\mathrm{var}(X) &= \frac{F}{\gamma_x}, \\
\mathrm{var}(Y) &= \frac{FR_0\left[(F\lambda_1+\gamma_x\gamma_y)^2\left(F\lambda_1+\gamma_x(\gamma_x+\gamma_y)\right)+R_0\gamma_x^3\gamma_y^2\right]+F\lambda_1^2\lambda_2^2\gamma_x^3}{(F\lambda_1+\gamma_x\gamma_y)^3\left(F\lambda_1+\gamma_x(\gamma_x+\gamma_y)\right)} \\
&+ \frac{\lambda_2\gamma_x\left(F^3\lambda_1^3+F^2\lambda_1^2\gamma_x(\gamma_x+3\gamma_y)+F\lambda_1\gamma_x^2\gamma_y(-2R_0+2\gamma_x+3\gamma_y)+\gamma_x^3\gamma_y^2(\gamma_x+\gamma_y)\right)}{(F\lambda_1+\gamma_x\gamma_y)^3\left(F\lambda_1+\gamma_x(\gamma_x+\gamma_y)\right)}, \\
\mathrm{cov}(XY) &= \frac{F\gamma_x(R_0\gamma_y-\lambda_1\lambda_2)}{(\gamma_x\gamma_y+F\lambda_1)(\gamma_x^2+\gamma_x\gamma_y+F\lambda_1)}.
\end{aligned} \tag{1.16}$$

Inserting above variances into Eq. (1.7), we will obtain the corresponding mutual information. The above example illustrates how one can obtain a variety of signaling properties analytically using the chemical Langevin approach. We will generalize this technique to a more realistic many-species model in Chapter 2, and also validate the approach by comparison to kinetic Monte Carlo simulations.

1.2 Single-molecule heterogeneity

Functional heterogeneity of single bio-molecules is significant variation in functional properties (i.e. catalytic rates, bonding lifetimes) among covalently identical bio-molecules, arising from multiple, distinct (and sometimes long-lived) structural conformations. Functional heterogeneity is widely observed in many classes of bio-molecules such as protein enzymes [8–10], ribozymes [11], DNA [12], motor proteins[13], and adhesion complexes [14]. This type of heterogeneity allows molecules to have different functional responses to changes in the external environment (i.e. differences in applied tension on an adhesion complex). It is also effectively a source of epigenetic variation that can play

a role in evolution, since the same genetic sequence can lead to a protein that exhibits a variety of phenotypes.

The question we consider in this thesis is can we devise a method to quantify the extent of heterogeneity from experiments, specifically single-molecule force spectroscopy conducted by atomic force microscopy (AFM). The experimental data is the rupture times (or equivalently forces) one collects from an AFM pulling experiment as shown as Fig. 1.4.

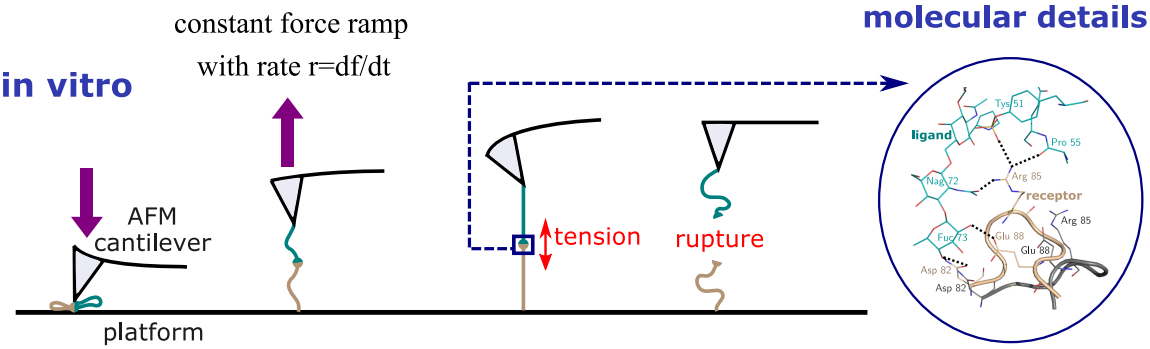

Figure 1.4. Schematic of an atomic force microscopy pulling experiment.

Typically, one connects the bio-molecule to the AFM cantilever and platform through protein or nucleic acid linkers of known stiffness. Suppose the cantilever is pulled at a constant velocity v, applying a force ramp with slope $df/dt = \omega_s(f)v$, where $\omega_s(f)$ is the effective stiffness of the setup (linkers plus the AFM cantilever). While the AFM cantilever is approximately a Hookean spring, the $\omega_s(f)$ may in general depend on the force because of the polymeric properties of the linkers. For simplicity, we define a characteristic stiffness $\bar{\omega}_s \equiv$ the mean $\omega_s(f)$ over the range of forces probed in the experiment (note that the precise value of $\bar{\omega}_s$ is not crucial in this work). This allows us to introduce a constant characteristic force loading rate r proportional to the velocity, $r = \bar{\omega}_s v$. Therefore, we can write the force ramp $r = df/dt$. The force is ramped up until rupture

occurs (in the case of a complex of bio-molecules adhered together) or until unfolding occurs (in the case of a single bio-molecule). For simplicity we will refer to both cases as "rupture" times. If the molecule or molecular complex can exist in distinct conformational states, differences in the non-covalent bonding in those states can lead to very different distributions of rupture times. Our focus will be in systems where such states are long-lived relative to the experimental run time, since multiple states that interconvert rapidly before rupture lead to results that effectively look like a single state [15]. Moreover, as was shown in Ref. [15], there are ways to rule out the rapid interconversion scenario based on the data. Hence the assumption will be that if we start in a certain conformational state at the beginning of the experimental run, we will be in the same state at the moment of rupture.

This experimental procedure is repeated multiple times, collecting a set of rupture times (or forces) for the bio-molecule of interest. Now, we can pose the above question more specifically. Suppose one did the AFM pulling experiments 200 times with one certain bio-molecule, is it possible that one can quantify the heterogeneity of the bio-molecule by analyzing the 200 rupture times at loading rate r?

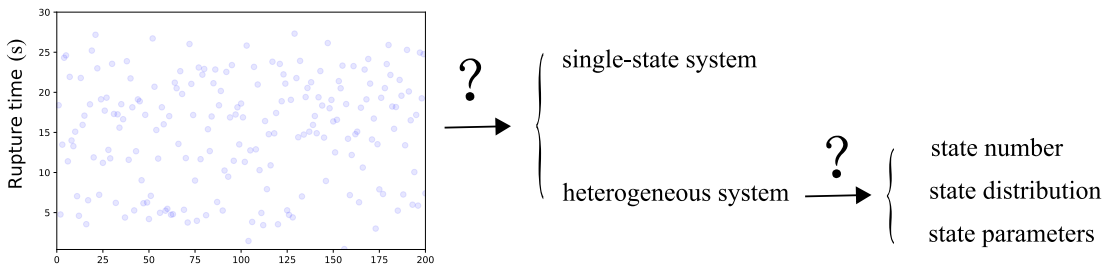

Figure 1.5. The left panel visualizes raw data one collects from 200 repetitions of an AFM pulling experiment. On the right is a schematic of questions we try to answer through the analysis of the rupture time data.

As shown in Fig. 1.5, the first question we want to answer is: can we tell if the data is collected from a single-state system or a heterogeneous (multiple state) system? If we identify that the bio-molecule has multiple functional states, can we specify how many functional states exist? If we could do this, i.e. deduce there are three states, can our analysis give information about the state probability distribution $\boldsymbol{p} = \{p_1, p_2, p_3\}$, where p_i is the probability that an experimental run involves a molecule in state i. Furthermore, if the answers are yes for all above questions, can we provide physical parameters that characterize each of the functional states? In the context of pulling experiments, these parameters would describe the different rupture time distributions associated with each state.

In this thesis, we provide potential solutions for the above questions with the help of two machine learning techniques: deep learning and non-parametric Bayesian learning. To set up the detailed discussion of these in Chapter 3, in the following sections we introduce some basics of machine learning.

1.2.1 Machine learning

Machine learning is a subdomain of computer science whose goal is to develop algorithms that learn automatically from data—for example using data to identify hidden patterns, infer underlying models, or make predictions. Of course such data analysis has been an essential part of many disciplines—science, statistics, economics—long before the electronic computer was invented. However, in recent years, thanks to the rapid development of information technology and computer science, machine learning has dramatically broadened the scope of data analysis techniques. Our increasing ability to generate big data sets for training purposes, and rising computational capacity, have altered problem solving strategies in many fields. Machine learning algorithms are now

used in a wide array of applications [16], such as medical diagnostics, spam detection, computer vision, natural language processing, autopiloting, news recommendation, social network filtering, finance, material science, game design and more. Physics, with its close ties to mathematics and computer science, and its rich troves of experimental data, is no exception to this trend.

This section introduces basic concepts of machine learning, some of them are general principles that can be applied to many different varieties of such algorithms. However, the main focus will be on machine learning techniques that are essential or closely related to our single-molecule heterogeneity work. Readers who seek a more complete and comprehensive coverage of machine learning are encouraged to explore machine learning textbooks like Refs. [17, 18].

Machine learning by its nature is a type of statistical learning that requires the use of computers. It is divided into two broad categories: supervised or unsupervised. In simple words, supervised learning is where you have input variables $\mathbf{x}$ and an desired target $\mathbf{Y}$ and you use an algorithm to learn the mapping function from the input to the target. In contrast, unsupervised learning has no predefined desired target and the learning task is to model the underlying structure, pattern, features or distribution of the input data set. We use both types of learning algorithms in our study of single-molecule heterogeneity, and explore their relative strengths and weaknesses. Brief introductions for both our supervised (deep learning) and unsupervised (non-parametric Bayesian) learning algorithms are provided in this section, and the details of how to apply them to AFM pulling experimental data will be described in Chapter 3. Generally, a machine learning algorithm consists of multiple components, including but not limited to a mathematical model, training data set, a cost function, and an optimizer (or optimization algorithm).

We will highlight how these work by first considering the example of deep learning below.

1.2.2 Deep learning

In 2012, Jeff Dean and Andrew Y. Ng created a network that is able to detect higher-level categories, such as cat faces and human bodies, through processing unlabeled images and YouTube videos [19]. This was one of the first widely publicized successes of "deep learning", covered in venues like *Scientific American*. The term "deep learning" has become more and more popular in recent years, thanks to the increased computing power from GPUs and distributed computing, allowing the design of large-scale networks for progressively more complex tasks. Though developments in modern deep learning techniques are not necessarily inspired by analogies to neuroscience, as they were at their origin [17], the field was once called artificial neural networks and most deep learning algorithms are still based on artificial neural networks [16]. It all started in 1943 when Warren McCulloch and Walter Pitts posited that biological neural networks could be described by means of propositional logic and mathematical modeling [20]. In trying to simulate learning, it is natural to draw a connection to the organ that spectacularly achieves this task: the human brain. Covering the historical evolution of deep learning is beyond the scope of this thesis, however it is quite important to note that a lot of early developments in learning algorithms were motivated by modeling how learning works in the brain. In recent years, artificial neural networks are usually not designed to precisely and realistically reproduce the biological function of the nervous system. Nevertheless, the original biological motivation still provides several key ideas. First, by reverse engineering the mathematical principles and logic that underlie the biological function of the brain, one can obtain a kind of artificial intelligence (AI) that is able to

tackle complex tasks that were traditionally the domain of of natural intelligence. This is optimistically supported by the conjecture: "Every aspect of learning or any other feature of intelligence can be so precisely described that a machine can be made to simulate it" in the proposal of the 1956 Dartmouth Summer Research Project on Artificial Intelligence [21], considered the founding event of AI as a field. Second, exploring neural networks would be beneficial for understanding the brain and hence uncovering the true nature of intelligence. Finally, machine learning methods might inspire new methods to solve fundamental scientific questions in various disciplines.

1.2.3 Artificial neural network

In a generic supervised learning problem, we have an input vector $\mathbf{x}$ and a desired target $\mathbf{Y}$. For our single-molecule heterogeneity study, these would be the experimental observations and the underlying state probability distribution respectively. Though artificial neural networks might look mysterious and profound to people who are new to the machine learning field, they are generally just nonlinear statistical models designed to approximate a certain function F on the input. For example, assume there exists a function F that outputs the correct target given any input, $\mathbf{Y} = F(\mathbf{x})$. Artificial neural networks are designed to implement a mapping that outputs $\mathbf{y} = f(\mathbf{x};\theta)$ based on some unknown parameters θ. The learning process then attempts to optimize these parameters to find a good approximation for the function F.

In this section, we will describe a simple example of an artificial neural network called the single hidden layer back-propagation network (or single layer perceptron), which is a representative deep learning model. As shown in Fig 1.6, the single layer perceptron typically has a 3-layer architecture. For our input layer, we have an input vector

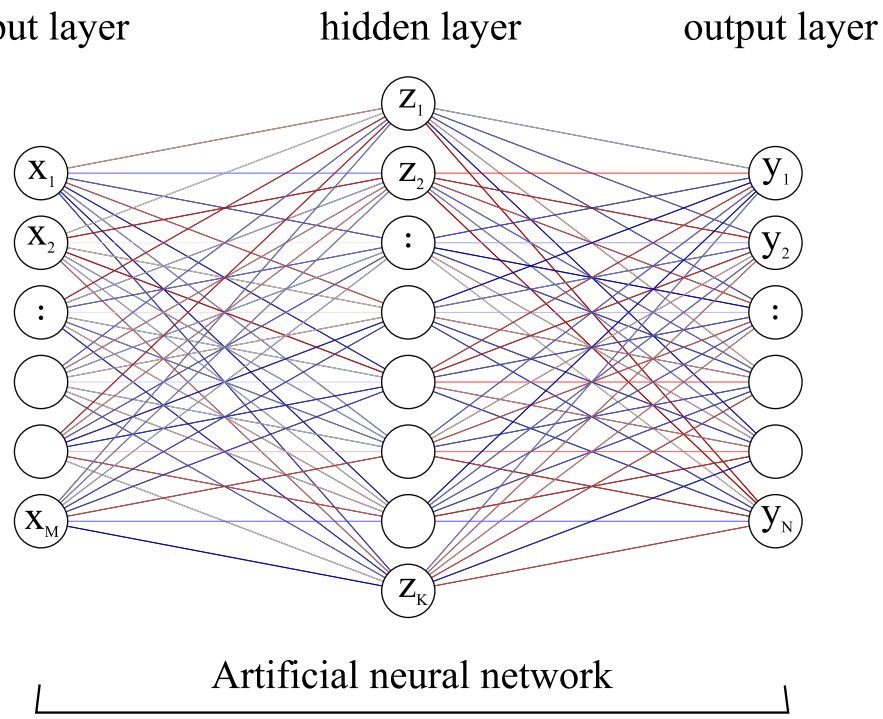

Figure 1.6. Schematic of a single hidden layer, feed-forward neural network.

$\mathbf{x} = \{x_1, x_2, ..., x_M\}$ with M components. The network outputs a vector $\mathbf{y}$ with N components. For example, if this network were used for a regression problem, we would choose $N = 1$, while $N \geq 2$ for an N-class classification problem. In the latter case, the n-th component of $\mathbf{y}$ could represents the probability of being in the nth class. The hidden layer consists of K neurons, where each neuron outputs a value z_k, $k = 1, 2, ..., K$, based on linear combinations of the inputs. The output components y_n, $n = 1, 2, ..., N$, are obtained from linear combinations of the values z_k. Mathematically, this can be expressed as:

$$\begin{aligned} z_k &= h(\alpha_{0k} + \alpha_k^{\mathsf{T}}\mathbf{x}), \qquad k = 1, 2, ..., K \\ y_n &= \sigma(\mathbf{v})_n, \qquad n = 1, 2, ..., N, \end{aligned} \tag{1.17}$$

where the n-th component of the vector $\mathbf{v}$ is defined as $v_n \equiv \beta_{0n} + \beta_n^\mathsf{T}\mathbf{z}$, the vectors α_k^T and β_n^T have same dimensionality as $\mathbf{x}$ and $\mathbf{z}$ respectively. The functions h and σ will be described below. We now look more closely at the various aspects of the network.

Hidden layer: The middle layer is called the hidden layer because the vector $\mathbf{z}$ that best implements the correct input-to-target mapping is not known beforehand. In our single layer perceptron example we have only one hidden layer, but in real-world applications of artificial neural networks multilayer perceptrons are widely used. The total layer number represents the "depth" of the model, and that is where the term "deep learning" comes from. In a multilayer network the input to each hidden layer after the first is just the output of the previous one. The original input $\mathbf{x}$ is progressively transformed layer by layer, allowing the network to modify which features of the input data are given greater or lesser weight, or implement an alternative representation for $\mathbf{x}$.

Neurons: Each individual unit of a hidden layer is called a neuron because they are conceptually inspired by biological neurons. As shown in Eq. (1.17), the hidden layer collectively involves many units working simultaneously, each implementing a vector ($\mathbf{x}$) to scalar (z_k) transformation. The units are similar to biological neurons in the sense that they process a signal from the external input (or the output of other neurons in the multilayer case) and "fire" a corresponding output value. The idea of using many layers of vector-valued representation is also derived from neuroscience, with the connections (colorful lines in Fig. 1.6) analogous to synapses.

Activation function for neurons: The output z_k of each neuron, as shown in Eq. (1.17), depends on an activation function h. Historically, activation functions were inspired by modeling the rate of action potential firing in biological neurons [22]. For the simplest case where the neuron only has two output states (on or off), the activation function

could be written as a Heaviside step function. The non-linear aspect of the activation function turned out to be crucial for certain learning tasks, but the step function fell out of favor over time, replaced by other alternatives that gave better results. For example, the sigmoid function $h(v) = 1/(1+e^{-v})$ was once widely used. In addition to being nonlinear, it is smooth (good for optimization algorithms), and the fact that it outputs a value between 0 to 1 makes the result easily interpretable as a probability. However in modern deep learning models, the rectified linear unit (ReLU) is recommended by default [23–25]. Both sigmoid and ReLU type activation function are depicted in Fig. 1.7. Since the ReLU function is piecewise linear, it preserves some appealing features of linear models: easier optimization with gradient-based methods, and better generalization [17]. It is important to note that, even though the choice of the function and how we build the neural network was originally guided by observed features of biological neural networks, modern artificial neural network design generally does not have precise modeling of the brain as its objective.

Bias: α_{0k} and β_{0n} are bias terms we add for the vector to scalar transformation.

Output function: The final output vector $\mathbf{y}$ in Eq.(1.17) depends on an output function σ. For example, if this network is designed for a regression problem, the output function σ can simply be an identity function. For N-class classification, the σ is often chosen to depend on the entirety of the vector $\mathbf{v}$, where $v_n \equiv \beta_{0n} + \beta_n^{\mathsf{T}}\mathbf{z}$ through the so-called softmax function:

$$\sigma(\mathbf{v})_n = \frac{e^{v_n}}{\sum_{j=1}^{N} e^{v_j}}. \tag{1.18}$$

By construction $\sum_{n=1}^{N} \sigma(\mathbf{v})_n = 1$, which allows the softmax output to represent a probability vector. Note that if we choose both h and σ to be identity functions, the whole

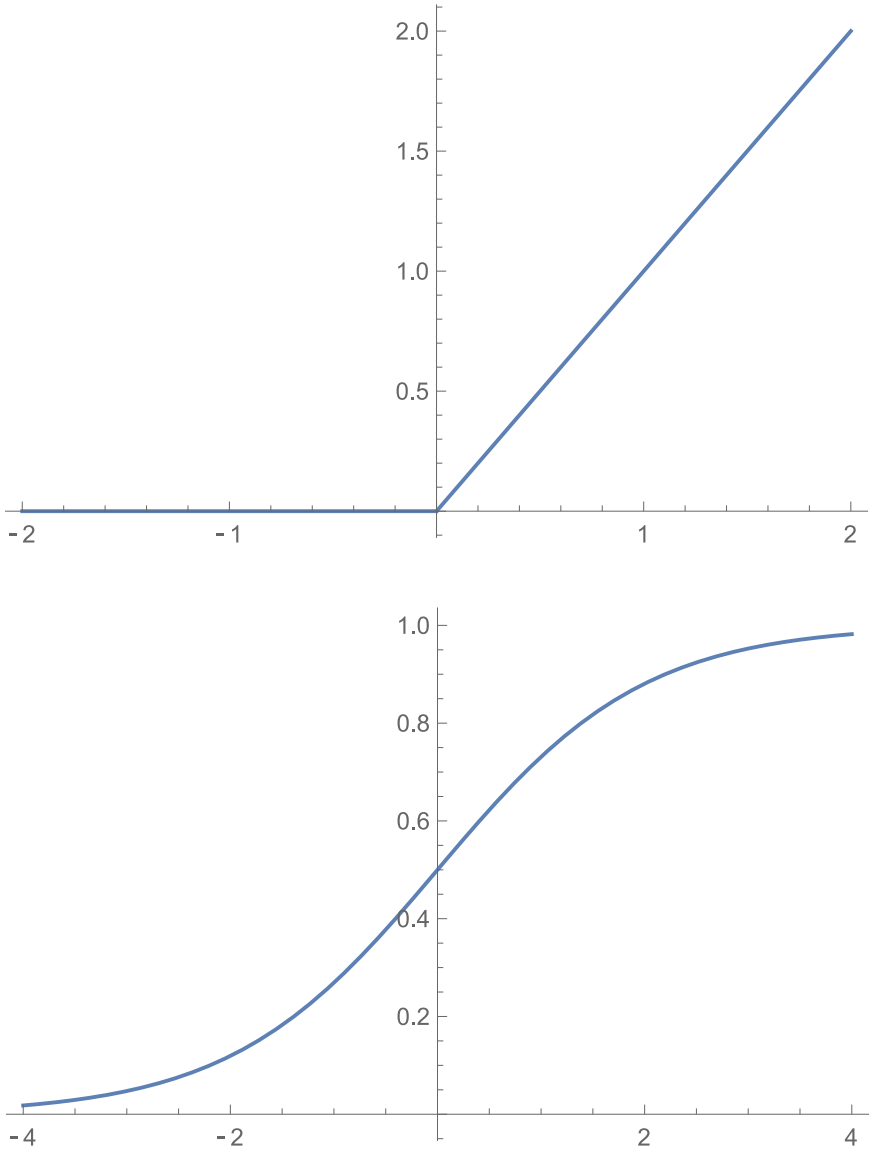

Figure 1.7. (Top) The rectified linear (ReLU) activation function, the default functional form recommended for neural networks. (Bottom) The sigmoid function, an alternative activation function.

network would reduce to a linear model, and the results would be an implementation of conventional linear regression. It is useful to think of the neural network as a nonlinear generalization of the linear model that enlarges the model capacity. For instance, a

linear model could have trouble taking the interaction of two input variables into consideration, while an artificial neural network with nonlinear activation functions would have none. Ultimately, enlarging the model capacity increases our chance of finding a good approximation for the function F that outputs our desired target $\mathbf{Y}$.

1.2.4 Training the artificial neural network

As mentioned above, the goal of training our artificial neural network is to find a function f that gets the output of the network $\mathbf{y} = f(\mathbf{x};\theta)$ as close as possible to our desired target $\mathbf{Y}$. As one can see from Eq. (1.17), the parameters such as $\alpha_{0k}, \alpha_k, \beta_{0n}, \beta_n$ are unknown for each neuron. These unknown parameters are often referred to as "weights" and constitute the parameter vector θ. Deep learning involves optimizing the values for these weights by fitting training data.

Training data: For supervised learning, training data is a set of data that consists of labeled input and labeled target, where the ith input vector corresponds to the ith labeled target vector, $\{\mathbf{x}^{(i)}\} \rightarrow \{\mathbf{Y}^{(i)}\}$.

Weights: These are the parameters that regulate the strength and sign of connections between neurons, depicted as colorful lines in Fig. 1.6. For our example network the weights are the parameters $\alpha_{0k}, \alpha_k, \beta_{0n}, \beta_n$ in Eq. (1.17),

Since we need to find the good fit for the training data, the first step is to define a quantity that measures how good our fit is. This quantity is normally called a loss function or cost function. The weights are obtained by minimizing the loss function. For instance, if we are facing a classification problem, the target $\mathbf{Y}^{(i)}$ is just a vector of zeroes except for a value of 1 at the position corresponding to the class to which the ith

input belongs. The loss function in this case could be the mean squared error,

$$L(\theta) = \sum_{i \in \text{data}} (\mathbf{Y}^{(i)} - f(\mathbf{x}^{(i)}; \theta))^2, \tag{1.19}$$

or the commonly used cross-entropy,

$$L(\theta) = - \sum_{i \in \text{data}} \sum_{n=1}^{N} Y_n^{(i)} \log f_n(\mathbf{x}^{(i)}; \theta). \tag{1.20}$$

Note that the cross-entropy loss function is just the negative log-likelihood, so minimizing the cross-entropy is the same as maximizing the likelihood.

Normally, the global minimum of a loss function cannot be analytically derived, especially for a non-linear model. Generically, variants of gradient descent are used as an iterative numerical optimization procedure to minimize the loss function $L(\theta)$.

Here we show the simplest gradient descent optimization approach for our example, though in practice one would typically use an alternative like stochastic gradient descent that allows us to escape local minima. For our weights α_{km} ($k = 1, 2, ..., K; m = 1, 2, ..., M$) and β_{nk} ($n = 1, 2, ..., N; k = 1, 2, ..., K$), we can write derivatives $\frac{\partial L(\theta)}{\partial \alpha_{km}}$ and $\frac{\partial L(\theta)}{\partial \beta_{nk}}$ respectively. For the i-th iteration update of gradient descent optimization, we have the weights:

$$\begin{aligned} \alpha_{km}^{(i)} &= \alpha_{km}^{(i-1)} - l_r \frac{\partial L(\theta)}{\partial \alpha_{km}^{(i-1)}}, \\ \beta_{nk}^{(i)} &= \beta_{nk}^{(i-1)} - l_r \frac{\partial L(\theta)}{\partial \beta_{nk}^{(i-1)}}, \end{aligned} \tag{1.21}$$

where the l_r is called learning rate, a positive value determining the update size. Usually we set l_r to be a small constant when searching for a minimum. However, the precise value choice for the learning rate can be tricky and it generally varies case by case. In fact, the learning rate itself can be optimized by an algorithm: a line search that minimizes the loss function for each update. The gradient descent update converges when

the gradient vanishes. If this happens to be the global minimum, this is where the output $\mathbf{y}$ is closest to the desired target $\mathbf{Y}$.

1.2.5 Non-parametric Bayesian learning

Consider the rupture time data we mentioned above in the discussion of AFM pulling experiments. If this data originates from multiple conformational states, then all the rupture times that correspond to the same state presumably have some features in common (i.e. are drawn from the same underlying rupture time distribution). If the algorithm could somehow use these features to group data by their state (a type of unsupervised learning called clustering) and estimate the probability p_i of each state, then we would have an effective method for identifying heterogeneity. Our approach to implement this clustering is known as Bayesian non-parametric learning.

Before we get into the details of Bayesian non-parametric learning, let us first briefly review some concepts in Bayesian statistics. Bayesian statistics is based on the notion that a probability can be interpreted as the degree of belief in an event. When we need to predict the underlying model based on observations (data collected), we should consider all possible models (meaning all possible parameter choices). Our differing levels of belief in different models, given the observations, forms a probability distribution over all possible models. Bayesian inference is used to update the probabilities for each model if more observations become available.

Ordinary (so-called *parametric*) Bayesian inference methods work when we have a finite number of parameters that defines the space of all possible models. The problem for our heterogeneity analysis is that the number of parameters is not known beforehand. For example each conformational state might have an associated distribution of rupture times that depends on a small set of parameters, but we have no idea if the

number of such states is 2 or 100. If the truth is somewhere in the middle, assuming 2 would lead to underfitting, while assuming 100 would lead to overfitting. In principle our learning algorithm should be open to *all* possible numbers of states, and hence an arbitrarily large (effectively infinite) number of parameters. This is precisely the problem that *non-parametric* Bayesian inference methods were designed to tackle. We will concentrate in particular on one such method, known as a Dirichlet process mixture model, that is ideally suited for our heterogeneity problem.

1.2.6 Dirichlet process mixture model

A mixture model is a description of a set of data (like our AFM rupture times) in terms of a collection of subpopulations, where each subpopulation has its own characteristic distribution. In our case, all the experimental runs where the molecule was in state i would form the ith subpopulation. The relative fractions of the subpopulations in the total are known as mixture weights, which correspond to the state probabilities p_i discussed above. Since we do not know the true number of subpopulations beforehand, we do not want to use a restrictive prior on the number of mixture weights (for example assuming that the $\mathbf{p} = (p_1, p_2, p_3)$ vector always has length three). Instead we use a more flexible prior that allows for a countably infinite number of subpopulations, known as Dirichlet process. This leads to a so-called Dirichlet process mixture model. Historically, such models date back to the work of Antoniak [26] and Ferguson [27] in the 1970s. However, analyzing data with such models was not computationally feasible until Markov chain sampling methods (described below) were developed two decades later. In 1995, the Gibbs sampling approach introduced by Escobar and West [28] finally made this model practical, and since then a variety of more efficient sampling techniques have been developed. Our description in this section is based on the excellent review by Neal [29].

To understand the model in more detail, let us define a number of basic quantities. Our data set consists of N observations, which we will denote as a vector $\boldsymbol{t} = (t_1, t_2, ...t_N)$. This is an exchangeable sequence, in the sense that each rupture time t_i was calculated using an independent experimental run, so the joint probability of observing a certain sequence does not depend on the order in which we write down the times. The hypothesis is that each time t_i is drawn from some unknown mixture distribution. We can describe the mixture distribution using two vectors: (i) the first is the "class" vector $\boldsymbol{c} = (c_1, ..., c_N)$, where c_i is the label of the class (subpopulation) to which the ith data point belongs. For example imagine there are three subpopulations, corresponding to three conformational states, which we label with integers $1-3$. Then $c_i = 2$ would indicate that the ith experiment involved a molecule in conformational state 2. (ii) Each distinct class with label c has an associated set of physical parameters ϕ_c (which could be multidimensional, though we write it for simplicity as a scalar). These parameters characterize the distribution of rupture times from that state: $F(t|\phi_c)$, the probability density that one would observe a rupture time t given the parameter set ϕ_c. Note that this distribution is normalized so that $\int_0^\infty dt\, F(t|\phi_c) = 1$. The functional form for $F(t|\phi_c)$ will depend on the problem, but in the AFM pulling case as we describe in Chapter 3 it can be well approximated using a Bell model [30]. For our discussion here the precise nature of $F(t|\phi_c)$ is not important, but we assume the functional form is known (or guessed) from physical considerations, even if we do not know the parameters ϕ_c that characterize each class. We collect the parameters for all the distinct classes into a vector $\boldsymbol{\phi}$:

$$\boldsymbol{\phi} \equiv \left\{\phi_c : c \in \{c_1, ..., c_N\}\right\} \tag{1.22}$$

Note that $\boldsymbol{\phi}$ might have different dimensionality depending on how many distinct classes there exist in the set $\{c_1,\ldots,c_N\}$. Finding the true underlying $\boldsymbol{c}$ and $\boldsymbol{\phi}$ for a given data set would be the ultimate (and typically unattainable) goal of the heterogeneity analysis. If for example there are $N = 7$ observations, then the ground truth might look like:

$$\boldsymbol{c} = (2,1,2,2,3,2,1), \qquad \boldsymbol{\phi} = (\phi_1,\phi_2,\phi_3). \tag{1.23}$$

Here there are three distinct classes, with parameter sets (ϕ_1,ϕ_2,ϕ_3), and $\boldsymbol{c}$ identifies which experimental run came from which class. In a Bayesian context the goal is more modest: we will not necessarily be able to zero in on the true $(\boldsymbol{c},\boldsymbol{\phi})$ of the mixture model that produced the data set, but ideally we should be able to evaluate the posterior probability $P(\boldsymbol{c},\boldsymbol{\phi}|\boldsymbol{t})$. This is the probability of having a certain $(\boldsymbol{c},\boldsymbol{\phi})$, given our experimental observations $\boldsymbol{t}$. If we could maximize $P(\boldsymbol{c},\boldsymbol{\phi}|\boldsymbol{t})$ with respect to $(\boldsymbol{c},\boldsymbol{\phi})$, we could find the most likely values for $(\boldsymbol{c},\boldsymbol{\phi})$, which would be our best estimate of the truth.

To tackle this maximization question, let us rewrite $P(\boldsymbol{c},\boldsymbol{\phi}|\boldsymbol{t})$ using Bayes's theorem:

$$P(\boldsymbol{c},\boldsymbol{\phi}|\boldsymbol{t}) = \frac{P(\boldsymbol{t}|\boldsymbol{c},\boldsymbol{\phi})P(\boldsymbol{c},\boldsymbol{\phi})}{P(\boldsymbol{t})} \equiv AP(\boldsymbol{t}|\boldsymbol{c},\boldsymbol{\phi})P(\boldsymbol{c},\boldsymbol{\phi}). \tag{1.24}$$

Here the constant $A = 1/P(\boldsymbol{t})$ reflects the fact that $P(\boldsymbol{t})$ does not depend on $(\boldsymbol{c},\boldsymbol{\phi})$, and hence is irrelevant for the maximization. However if we want to know that actual value of $P(\boldsymbol{c},\boldsymbol{\phi}|\boldsymbol{t})$ we will need to worry about this factor as a normalization constant. Because $P(\boldsymbol{c},\boldsymbol{\phi}|\boldsymbol{t})$ must be a properly normalized probability distribution over $(\boldsymbol{c},\boldsymbol{\phi})$, the constant has to satisfy:

$$A = \int d(\boldsymbol{c},\boldsymbol{\phi})\, P(\boldsymbol{t}|\boldsymbol{c},\boldsymbol{\phi})P(\boldsymbol{c},\boldsymbol{\phi}), \tag{1.25}$$

where the integral is over all possible choices of $(\boldsymbol{c},\boldsymbol{\phi})$. In practice it is often difficult or impossible to evaluate Eq. (1.25) directly.

The first factor $P(\boldsymbol{t}|\boldsymbol{c},\boldsymbol{\phi})$ in the numerator of Eq. (1.24) is the likelihood of observing data $\boldsymbol{t}$ given $(\boldsymbol{c},\boldsymbol{\phi})$. This can be expressed as:

$$P(\boldsymbol{t}|\boldsymbol{c},\boldsymbol{\phi}) = \prod_{i=1}^{N} F(t_i|\phi_{c_i}). \tag{1.26}$$

The second factor $P(\boldsymbol{c},\boldsymbol{\phi})$ is the prior distribution of $(\boldsymbol{c},\boldsymbol{\phi})$, reflecting our assumptions about the possible values of $(\boldsymbol{c},\boldsymbol{\phi})$ before taking into account any experimental observations. If we knew that there were a fixed number of classes, for example 3, then $P(\boldsymbol{c},\boldsymbol{\phi})$ would reflect this constraint, only taking non-zero values when the length of $\boldsymbol{\phi}$ is 3. In our case we will use a less restrictive prior, based on a Dirichlet process, described in detail in the next section. This prior allows $(\boldsymbol{c},\boldsymbol{\phi})$ choices where $\boldsymbol{\phi}$ can have various lengths. For now however let us assume we have specified a suitable prior.

Even after choosing a prior, the problem is that the posterior distribution in Eq. (1.24) may be difficult to directly evaluate (particularly because of the normalization constant in Eq. (1.25)), or even directly sample from. There are certainly exceptions to this, such as when the prior is specifically compatible with the likelihood (a so-called conjugate prior) in a way that allows for easier evaluation of the posterior. However when we have some non-traditional, problem-specific F function determining the likelihood in Eq. (1.26), figuring out a conjugate prior becomes challenging. The alternative in this case is to devise a numerical scheme that will generate samples $(\boldsymbol{c},\boldsymbol{\phi})$ from the posterior distribution in Eq. (1.24). Then quantities that depend on knowledge of the posterior, like the integral in Eq. (1.25) or expectation values or marginal distributions with respect to the posterior, can all be estimated assuming we have a large enough set of samples.

How to generate such samples is the goal of a set of techniques known as Markov chain Monte Carlo (MCMC) methods. We will show an algorithm that implements a specific MCMC sampling method for our heterogeneity problem in Chapter 3. The general idea is as follows: imagine that you start with a particular choice of $(\boldsymbol{c}, \boldsymbol{\phi})$. The MCMC method is a stochastic algorithm that allows us to choose a new sample $(\boldsymbol{c}', \boldsymbol{\phi}')$ given that the current sample is $(\boldsymbol{c}, \boldsymbol{\phi})$. This constitutes one iteration of the algorithm, and we can then go from $(\boldsymbol{c}', \boldsymbol{\phi}')$ to $(\boldsymbol{c}'', \boldsymbol{\phi}'')$ and continue iterating as long as we desire. The end result is a chain of K samples $(\boldsymbol{c}^{(j)}, \boldsymbol{\phi}^{(j)})$, $j = 1, \ldots, K$. Because the algorithm is stochastic, the $(\boldsymbol{c}', \boldsymbol{\phi}')$ value you get starting from a given $(\boldsymbol{c}, \boldsymbol{\phi})$ could be different every time you run it. Let us imagine that the probability of getting $(\boldsymbol{c}', \boldsymbol{\phi}')$ at the next iteration, starting from $(\boldsymbol{c}, \boldsymbol{\phi})$, is denoted by $W_{\boldsymbol{c},\boldsymbol{\phi};\boldsymbol{c}',\boldsymbol{\phi}'}$. Because the probability of the next sample only depends on the current one, the process is Markovian, and $(\boldsymbol{c}^{(j)}, \boldsymbol{\phi}^{(j)})$ constitutes a Markov chain whose transition matrix is W.

For this procedure to be useful, the generated samples should be distributed according to the posterior distribution in Eq. (1.24). To ensure this, our algorithm must be designed so that the posterior distribution $P(\boldsymbol{c}, \boldsymbol{\phi}|\boldsymbol{t})$ is the stationary distribution of the Markov chain. This occurs if the matrix W satisfies the following detailed balance condition with respect to the posterior:

$$P(\boldsymbol{c}, \boldsymbol{\phi}|\boldsymbol{t}) W_{\boldsymbol{c},\boldsymbol{\phi};\boldsymbol{c}',\boldsymbol{\phi}'} = P(\boldsymbol{c}', \boldsymbol{\phi}'|\boldsymbol{t}) W_{\boldsymbol{c}',\boldsymbol{\phi}';\boldsymbol{c},\boldsymbol{\phi}}. \tag{1.27}$$

In other words if we are at stationarity the probability of starting at $(\boldsymbol{c}, \boldsymbol{\phi})$ and jumping to $(\boldsymbol{c}', \boldsymbol{\phi}')$ at the next iteration is exactly counterbalanced by the probability of starting at $(\boldsymbol{c}', \boldsymbol{\phi}')$ and jumping to $(\boldsymbol{c}, \boldsymbol{\phi})$. One of the nice aspects of Eq. (1.27) is that fulfilling the condition does not depend on knowing the precise value of the normalization constant

A from Eq. (1.25), since it can be cancelled out on both sides of the equation. Of course there are many ways to construct an algorithm whose W matrix satisfies Eq. (1.27). In physics, the most well known example is the Metropolis-Hastings algorithm [31], historically the first general-purpose MCMC method. Often MCMC methods are tailored to specific problems in order to try to sample the stationary distribution as efficiently as possible. Note that the it may take many steps for the Markov chain to relax to the stationary state. Thus typically for a chain $(\boldsymbol{c}^{(j)}, \boldsymbol{\phi}^{(j)})$ generated by the method, we throw away samples with $j \leq K_b$, where K_b defines a "burn-in period" that is sufficiently long to guarantee equilibration [32]. The samples for $j > K_b$ are assumed to be representative draws from the posterior distribution.

Once we have numerically calculated a Markov chain $(\boldsymbol{c}^{(j)}, \boldsymbol{\phi}^{(j)})$, $K_b < j \leq K$, we can use it to do various kinds of analysis involving the posterior. For example, let $n_i(\boldsymbol{c})$ be a function that counts the number of times class i appears in the vector $\boldsymbol{c}$. In the case of the vector shown in Eq. (1.23), we have: $n_1(\boldsymbol{c}) = 2$, $n_2(\boldsymbol{c}) = 4$, $n_3(\boldsymbol{c}) = 1$. Note that $n_i(\boldsymbol{c}) = 0$ if the class i does not appear in $\boldsymbol{c}$. From the $n_i(\boldsymbol{c})$ values we can construct an estimate $\tilde{p}_j(\boldsymbol{c})$ for the underlying state probabilities p_j based on the sample $(\boldsymbol{c}, \boldsymbol{\phi})$. Since the state labels are arbitrary, for convenience we will sort the probability components from largest to smallest. To do this we define a sorting function $\sigma_j(\boldsymbol{c})$ that outputs the index of the jth largest value in the list $(n_1(\boldsymbol{c}), n_2(\boldsymbol{c}), \ldots)$. For Eq. (1.23) this would give: $\sigma_1(\boldsymbol{c}) = 2$, $\sigma_2(\boldsymbol{c}) = 1$, $\sigma_3(\boldsymbol{c}) = 3$. Then our $\tilde{p}_j(\boldsymbol{c})$ function can be defined as:

$$\tilde{p}_j(\boldsymbol{c}) \equiv \frac{n_{\sigma_j(\boldsymbol{c})}(\boldsymbol{c})}{N}. \tag{1.28}$$

The larger the size N of the experimental data set, the closer $\tilde{p}_j(\boldsymbol{c})$ would be to the true value of the jth largest probability, assuming we had a good estimate for the underlying $\boldsymbol{c}$. Let us look at one component of the $\tilde{\boldsymbol{p}}(\boldsymbol{c})$ vector, for example the largest one ($j = 1$) and treat it as an observable. We can calculate this observable for every sample $(\boldsymbol{c}^{(j)}, \boldsymbol{\phi}^{(j)})$, $K_b < j \leq K$ in our chain, and then plot a histogram for $\tilde{p}_1(\boldsymbol{c})$. The tallest bin in the histogram would correspond to our best estimate for the the largest probability. If the histogram was peaked near $\tilde{p}_1 \approx 1$ this would indicate that the data came from a single state system, while a peak at significantly smaller values of $\tilde{p}_1$ would point to heterogeneity. Chapter 3 discusses a variety of other observables that can be analyzed in similar fashion based on the MCMC results.

1.2.7 Dirichlet process prior and the "Chinese restaurant" analogy

The final element of our Bayesian non-parametric model that we need to specify is the prior distribution $P(\boldsymbol{c}, \boldsymbol{\phi})$ in Eq. (1.24). As mentioned earlier, this will be based on a Dirichlet process, denoted as DP(Φ, α). The process depends on two quantities: (i) the first is the prior distribution Φ for the parameter sets ϕ_c that determine the rupture time distribution $F(t|\phi_c)$ for each class c. $\Phi(\phi_c)$ would thus be the prior probability of a class having parameters ϕ_c, if we did not know anything about the experimental data points that belonged to that class. In practice, the parameters that enter into a specific $F(t|\phi_c)$ (for example the Bell model) have certain biological ranges. We use $\Phi(\phi_c)$ to encode these constraints, thus ensuring that our estimation procedure does not venture into unreasonable values for ϕ_c. (ii) The second quantity is the concentration parameter α, a positive number whose role will be highlighted below.

Perhaps the simplest way to explain the Dirichlet process DP(Φ, α) is to describe the way one would draw a sample $(\boldsymbol{c}, \boldsymbol{\phi})$ from it. This is done through a sequential algorithm,

choosing the class labels c_i for each data point t_i one by one, $i = 1,\ldots,N$, and along the way building up a parameter vector $\boldsymbol{\phi}$. At the end of the draw algorithm one has a single sample $(\boldsymbol{c},\boldsymbol{\phi})$ from the Dirichlet process. The draw algorithm has been likened to a seating system at a Chinese restaurant [33], and is sometimes called the "Chinese restaurant process" as a result. Though this hypothetical establishment is unlike any real-life Chinese restaurant, the metaphor provides a concrete illustration of the draw algorithm: imagine that the data points t_i correspond to N customers that enter the restaurant sequentially. The restaurant can provide potentially an infinite number of tables, and each table will correspond to a class c, characterized by a parameter set ϕ_c (the food at that table).

Every customer that enters the restaurant makes a stochastic decision about which table to sit at (the class c_i that will be assigned to data point t_i) and each table has infinite capacity. The first customer is special: they are immediately seated at table 1 ($c_1 = 1$) and the parameter set ϕ_1 for that table is drawn from the prior distribution Φ. However, the next customer (and subsequent ones) have two options: either sit at an occupied table or sit at a new, empty table. The probability that customer i sits at a new table is given by:

$$P(c_i \neq c_j \text{ for all } j < i \,|\, c_1,\ldots,c_{i-1}) = \frac{\alpha}{i-1+\alpha}. \tag{1.29}$$

If a new table seating occurs, the table (class) is given a label c one higher than largest previous class label, and a new parameter set ϕ_c drawn from Φ. On the other hand, there is a chance that the customer i seats at one of the occupied tables. Choosing an occupied table c occurs with probability

$$P(c_i = c|\, c_1,\ldots,c_{i-1}) = \frac{\nu_{i,c}}{i-1+\alpha}, \tag{1.30}$$

where $\nu_{i,c}$ is the number of c_j for $j < i$ where $c_j = c$. In other words, $\nu_{i,c}$ is the occupancy of the cth table, and the chance of choosing that table is proportional to its occupancy. If the customer chooses to sit at occupied table c, then we make the assignment $c_i = c$. No new parameter set needs to be drawn, since that class already has a parameter set. Note that before customer i makes their choice, $i-1$ customers have already been seated, so the sum of $\nu_{i,c}$ for all occupied c is $i-1$. Hence the probabilities of all occupied choices c from Eq. (1.30) and the probability of choosing a new table from Eq. (1.29) sum to 1.

Let us imagine how this seating process might play out in one realization. The second customer enters, and the probability they will choose to sit at table 1 is $1/(1+\alpha)$, since the first customer has been seated there. Or the second customer can choose to sit at a new table (table 2) with probability $\frac{\alpha}{1+\alpha}$. Assume the second customer chooses to sits at table 2. Then the third customer enters and has several choices: they can sit at the occupied table 1 with probability $1/(2+\alpha)$, the occupied table 2 with probability $1/(2+\alpha)$, or the new table 3 with probability $\alpha/(2+\alpha)$. This process continues until all the N customers have been seated. At the end of the algorithm we have constructed the vectors $\boldsymbol{c}$ and $\boldsymbol{\phi}$.

If we repeated this draw algorithm many times, each time we would get a $(\boldsymbol{c}, \boldsymbol{\phi})$ pair that could potentially have a different number of distinct classes. The entire set of such $(\boldsymbol{c}, \boldsymbol{\phi})$ samples can be said to be distributed according to the Dirichlet process $\mathrm{DP}(\Phi, \alpha)$. Note the role of the concentration parameter α in determining the nature of the $(\boldsymbol{c}, \boldsymbol{\phi})$ distribution. In the limit $\alpha \to 0$ no customer after the first would ever choose a new table, and $\boldsymbol{c} = (1, 1, \ldots, 1)$ for every draw. This would not be a good prior $P(\boldsymbol{c}, \boldsymbol{\phi})$ for our heterogeneity problem, because it would assume from the start that no heterogeneity is possible. Hence α should be finite, but the proper choice is problem-specific (and

there exist algorithms that do not keep α constant, but iteratively update it as part of the MCMC procedure [29]). The larger the α, the more heterogeneous the prior, in the sense that $(\boldsymbol{c}, \boldsymbol{\phi})$ with many more and varied numbers of classes are assumed possible. In Chapter 3 we discuss how we determined an appropriate value of α for our problem.

2 The price of a bit: energetic costs, bandwidth and the evolution of cellular signaling

2.1 Introduction

Survival for living cells depends in part on accurate and responsive signaling: the ability to collect enough information about the micro-environment to make decisions in response to external stimuli such nutrients, hormones, and toxic agents [34]. This capacity to react to extracellular cues developed early in evolutionary history, and is now seen at all levels of biological organization, from chemotaxis in unicellular organisms [35–37] to the pathways that regulate cell differentiation and disease in multicellular life [38–41]. Despite the resulting diversity of biochemical networks that implement this signaling, information theory provides a powerful universal framework to quantify the amount of information transferred through a network, allowing comparisons between different systems [42].

Over the last decade a remarkable experimental consensus has emerged from such comparisons: studies of both prokaryotic and eukaryotic signaling pathways have found they can transmit at most ~ 1 to 3 bits of information [43–50]. These values refer to

mutual information (MI) between pathway input (concentrations of a molecule representing the signal) and the output (concentrations of a downstream molecule produced by the network, sampled either at a single or multiple time points). MI is a measure of signal fidelity, representing the degree of correlation between input and output. Experiments have typically focused on a closely related quantity known as the channel capacity [6, 51]: the maximum MI achievable among all input distributions.

The consistently small channel capacities observed in cellular signaling pathways seem to indicate that cells operate with a fairly coarse representation of their surroundings: n bits of MI corresponds to being able to reliably distinguish between 2^n levels of the input, so a 1 bit pathway can only discriminate between "high" versus "low" concentrations of signal. Though 1 bit is typical for MI measured at single time points, one can achieve higher MIs by focusing on output responses collected over several time points [47, 48], or by designing the experiment to isolate single-cell responses (as opposed to estimating MI from the responses of a population of cells) [50]. But these enhancements, which can push values to the 2-3 bit range, do not change the fundamental order of magnitude of the MI.

The central question we explore in this work is to what extent this fundamental information scale is shaped by the energy requirements of the underlying biochemical signaling networks. In order to transmit information, these networks necessarily need to operate out of equilibrium, fueled by processes like ATP hydrolysis that consume energetic resources. Recent research highlights these costs as an essential factor in understanding constraints on signaling [35–37, 52–55], often focusing on the ATP hydrolysis chemical potential difference $\Delta\mu = \mu_{\mathrm{ATP}} - \mu_{\mathrm{P}_i} - \mu_{\mathrm{ADP}}$ between the reactant (ATP) and products (ADP and inorganic phosphate, P_i), quantifying the free energy available to drive the

system per ATP. Crossing a certain minimum threshold of $\Delta\mu$ is a prerequisite for a variety of signaling functions: accurate read-out of ligand-bound receptors [35, 36, 55], maintaining the phase coherence of oscillations in circadian clocks [52], or preserving the integrity of methylation-based "memory" to facilitate adaptation in chemotaxis [37]. This threshold is typically a few times larger (i.e. by a factor of $\sim 3-4$ [35, 55]) than the energy scale of thermal fluctuations, $k_B T$, where k_B is the Boltzmann constant and T the temperature. And indeed cells across the various domains of life maintain a sufficiently high $\Delta\mu \approx 21-29\ k_B T$ [5] to enable such functions.

The large value and remarkably narrow range of $\Delta\mu$ observed in modern organisms opens up additional questions. The metabolic cycles that sustain $\Delta\mu$, constantly replenishing ATP as it is hydrolyzed, must almost necessarily have been far more inefficient and wasteful in the earliest stages of evolutionary history [56]. To what degree could organisms operating with smaller $\Delta\mu$ still process information about their environment? What kinds of evolutionary pressures might have driven $\Delta\mu$ to its modern range? And if the costs of individual signaling systems are non-trivial [35, 37], could natural selection have driven these networks toward optimized, energy-efficient solutions?

To investigate these issues, we focus on one of the canonical signaling circuits in biology, the kinase-phosphatase "push-pull loop", which often forms a basic unit of more complicated signaling cascades [57–60]. An active kinase enzyme instigates the "push", chemically modifying a substrate protein via phosphorylation (consuming ATP in the process), while a phosphatase enzyme provides the "pull", dephosphorylating the modified substrate, reverting it to its original state. We derive the relationships between three facets of the system: i) the MI between the input (active kinase) and output (phosphorylated substrate) molecular populations; ii) the timescales over which the input signal

varies; and iii) the energy requirements, expressed in terms of $\Delta\mu$ and the rate of ATP consumption. Exploring the entire spectrum of kinase/phosphatase enzymatic parameters from bioinformatic databases, we find that physiological $\Delta\mu$ values are just large enough to enable an MI of 1-2 bits for the widest possible parameter range. However to achieve this MI for signals that vary rapidly in time becomes more challenging, requiring both precise fine-tuning of parameters and a certain minimum rate of ATP consumption. In fact, taking advantage of results from optimal noise filter theory [61, 62], we derive a remarkably simple analytical relationship that describes the tradeoffs between minimum ATP rate, the MI, and the maximum characteristic signal frequency (the so-called bandwidth) which the push-pull network can handle. Verified via extensive numerical simulations across the whole gamut of enzymatic parameters, this relation is a novel theoretical prediction that can be directly tested in future experiments. The relation rationalizes the observed range of MI by showing that values much higher than 1-2 bits would require sacrificing the ability to process fast-changing signals. Finally we explore the question of whether there exist evolutionary pressures that would push such a system to be energy efficient, optimizing the ATP consumption for a given target MI and bandwidth. Using a recently developed formalism relating metabolic costs to the strength of natural selection [63, 64], we show that these pressures can indeed be significant, particularly for single-celled organisms. We highlight a kinase-phosphatase loop in the yeast Hog1 signaling pathway as a system that may have been optimized by such pressures.

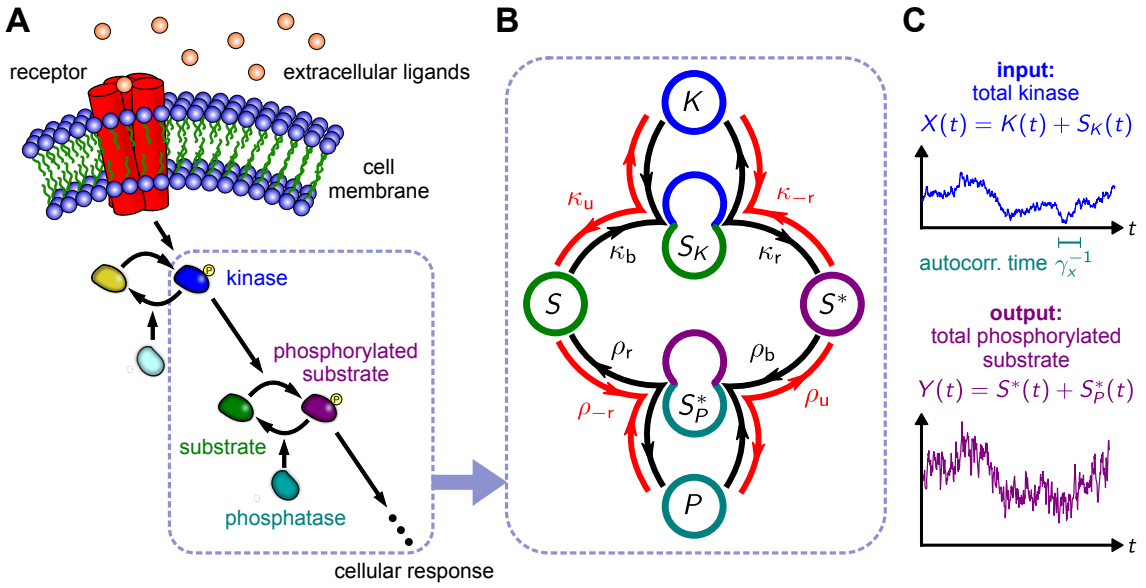

Figure 2.1. **(A)** A schematic signaling pathway involving cascades of kinase phosphorylation, initiated by a receptor embedded in the cell membrane that responds to extracellular ligands. The system we focus on will be one stage of the pathway, a kinase-phosphatase push-pull loop, highlighted in the dashed box. **(B)** The molecular species and reaction parameters of the push-pull loop. The kinase (K) binds to the substrate (S), forming the complex (S_K) that catalyzes the production of phosphorylated substrate (S^*). Phosphatase (P) binds to S^*, forming a complex (S_P^*) that catalyzes the dephosporylation of the substrate. Forward reaction / binding rates are labeled in black, while reverse reaction / unbinding rates are in red. **(C)** The loop serves to transduce an input signal, defined as the total population of kinase (bound or unbound), $X(t) = K(t) + S_K(t)$, into an output, defined as the total population of phosphorylated substrate, $Y(t) = S^*(t) + S_P^*(t)$. The input signal has a characteristic autocorrelation time γ_x^{-1}.

2.2 Theory

2.2.1 Modeling an enzymatic push-pull loop

This push-pull network consists of two opposing reactions: a kinase enzyme instigates the "push", chemically modifying a substrate protein via phosphorylation, while a phosphatase enzyme provides the "pull", dephosphorylating the modified substrate, reverting it to its original state [57–60]. Since a single kinase can catalyze the phosphorylation of many substrate proteins, this loop can effectively act like an amplifier [59], translating a weaker signal (a small cellular population of an active kinase) into a stronger one (a large population of a phosphorylated substrate). Often the substrate itself is a kinase that can exist in catalytically inactive and active states, with activation triggered by phosphorylation. In this case one can have multi-tiered signaling cascades enhancing the amplification (as shown schematically in Fig. 2.1A) with the active substrate produced by one loop serving as the kinase for a downstream loop [65]. More complex signaling networks are also possible, with multiple cascades connected by crosstalk through shared components [66], feedback from downstream to upstream populations [65], or activation requiring multisite phosphorylation [67]. However the starting point for understanding any of these more complex signaling topologies is the behavior of a single loop, with a substrate activated / deactivated through a single phosphorylation site.

The reaction scheme of a single push-pull loop is shown in Fig. 2.1B. Binding of free kinase (population $K(t)$ at time t) to substrate (population $S(t)$) occurs with rate constant κ_b, forming a kinase-substrate complex (population $S_K(t)$). Phosphorylation of the substrate and its subsequent release constitutes the catalytic step, with rate κ_r, yielding free phosphorylated substrates (population $S^*(t)$). A phosphatase can subsequently bind, with rate ρ_u, forming a phosphatase-substrate complex (population $S_P^*(t)$), and

catalyzing the dephosphorylation / release of the substrate with rate ρ_r. These reactions also can occur in reverse: kinase-substrate unbinding (rate κ_u), reverse kinase catalysis (rate κ_{-r}), phosphatase-substrate unbinding (rate ρ_u) and reverse phosphatase catalysis (rate ρ_{-r}). Under physiological conditions some of these reverse rates may be negligible compared to their forward counterparts, but accounting for them is crucial to enforce thermodynamic consistency. In fact the product of the ratios of the reverse rates relative to the forward ones must satisfy a key thermodynamic relation arising from the principle of detailed balance (closely related to the Haldane relation for enzymes) [3, 4],

$$\frac{\kappa_{-r}\rho_u\rho_{-r}\kappa_u}{\kappa_r\rho_b\rho_r\kappa_b} = e^{-\beta\Delta\mu}. \tag{2.1}$$

This relation is derived in the Supplementary Information (SI), found at the end of the chapter. It reflects the fact that for every complete traversal of the loop along the forward direction (clockwise along the black arrows in Fig. 2.1B) a single ATP molecule is removed from the environment, hydrolyzed, and the products ADP and inorganic phosphate P_i released back into the surroundings. $\Delta\mu$ depends on the concentrations [ATP], [ADP], and [P_i] through $\Delta\mu = \Delta\mu_0 + k_B T \ln([\text{ATP}](1\,\text{M})/([\text{ADP}][P_i]))$, where $\Delta\mu_0$ is the standard free energy of ATP hydrolysis ($\Delta\mu_0 \approx 12\ k_B T$ at room temperature [5]). Living systems expend energetic resources to maintain an imbalance of [ATP] relative to [ADP] and [P_i], making $\Delta\mu$ in physiological conditions larger than $\Delta\mu_0$. Despite the wide variety of metabolic pathways used to achieve this, measured $\Delta\mu$ values in organisms from *E. coli* to humans lie within a relatively narrow range, $\Delta\mu \approx 21 - 29\ k_B T$ [5]. This means reverse rates are sufficiently slow that the numerator in Eq. (2.1) is 9-12 orders of magnitude smaller than the denominator. One of the questions we tackle below is the significance of this disparity for transmitting information through the loop.

To quantitatively measure this information transfer, it is useful to explicitly describe the network behavior in terms of transducing an input signal into an amplified output, with degradation of the signal due to the stochastic nature of the reactions that mediate this process. We take the time-dependent input $X(t) = K(t) + S_K(t)$ to be the population of active kinases (both free and substrate-bound), and the corresponding output signal $Y(t) = S^*(t) + S_P^*(t)$ as the population of phosphorylated substrates (free and phosphatase-bound). For any specific system, the input kinases would be activated through a particular upstream signaling network. Here, however, we are interested here in a more general problem: what is the effectiveness of this loop in processing a variety of possible input signals, spanning different amplitudes and timescales. The simplest mechanism that allows us to tune the dynamical characteristics of the input is to imagine the kinases activated at a constant rate F and deactivated at a constant rate γ_K. We focus on the long-time limit where a stationary state has been achieved, and so F allows us to regulate the amplitude of the input signal while γ_K controls the autocorrelation time of the input fluctuations. While the analysis below could be done for other, system-specific models of the input, our choice allows us to explore a broad range of possible inputs to establish general bounds on information processing through the loop. With this input model, the reaction network model is fully specified. For a given set of parameters (drawn from distributions based on kinase/phosphatase biochemical information collected in enzymatic databases, as described below) we can derive analytical results for dynamical quantities using the linearized chemical Langevin approximation [7]. As shown in the SI, this provides excellent agreement with the exact kinetic Monte Carlo [68] simulation results in the parameter ranges of interest.

In focusing on how $X(t)$ is transduced to $Y(t)$, we frame our analysis in terms of three properties of the system. The first is the autocorrelation time of the input, γ_x^{-1}, defined through $\overline{\delta X(t+\tau)\delta X(t)} = \overline{\delta X^2}\exp(-\gamma_x|\tau|)$, where the bar denotes an average over an ensemble of trajectories in the stationary state and $\delta X(t) \equiv X(t) - \overline{X}$. Note that instantaneous averages like $\overline{X} \equiv \overline{X(t)}$ and $\overline{\delta X^2} \equiv \overline{\delta X^2(t)}$ are independent of t in the stationary state. γ_x^{-1} is the characteristic timescale of the input fluctuations, and we will denote its inverse, γ_x, as the effective "frequency" of the input. The second property is related to the mean rate at which phosphorylated substrates are produced through the catalytic reaction step, $\kappa_r\overline{S_K}$, relative to the mean total number of activated kinases $\overline{X}$. We define the gain parameter $R_0 \equiv \kappa_r\overline{S_K}/\overline{X}$ as a measure of the production of output for a given input level. Both γ_x and R_0 can be expressed, to a good approximation, in terms of the reaction rates as follows (see SI for derivation):

$$\gamma_x = \frac{C_1}{C_1 + C_2}\gamma_K, \quad R_0 = \frac{C_2}{C_1 + C_2}\kappa_r, \tag{2.2}$$

where $C_1 \equiv \kappa_-\overline{P}\gamma_K\rho_b\rho_r + F\kappa_{-r}\kappa_u\rho_-$, $C_2 \equiv \overline{S}[F\kappa_b\kappa_{-r}\rho_- + \overline{P}\gamma_K(\kappa_b\rho_b\rho_r + \kappa_{-r}\rho_{-r}\rho_u)]$. Here $\kappa_- \equiv \kappa_u + \kappa_r$, $\rho_- \equiv \rho_u + \rho_r$. Note the dependence on mean unmodified substrate $\overline{S}$ and free phosphatase $\overline{P}$ populations: these two numbers are free parameters that (along with the reaction rates) determine the network dynamics.

The final property of interest is the instantaneous stationary MI I between $X(t)$ and $Y(t)$. This is defined in terms of the joint probability $P(X,Y)$ of observing input value X and output value Y at the same moment of time, and the corresponding marginal probabilities $P(X)$ and $P(Y)$,

$$I = \sum_{X,Y} P(X,Y)\log_2\frac{P(X,Y)}{P(X)P(Y)}. \tag{2.3}$$

The value of I is non-negative in all cases, and is measured in bits, with larger values translating to a greater degree of correlation between input and output. For our parameter ranges, $P(X,Y)$ can be approximated as a bivariate Gaussian, and so we use an expression for I valid in this limit that is more convenient to evaluate [6]:

$$I \approx -\frac{1}{2}\log_2 E, \quad \text{where} \quad E \equiv 1 - \frac{(\overline{XY} - \overline{X}\,\overline{Y})^2}{\left(\overline{X^2} - \overline{X}^2\right)\left(\overline{Y^2} - \overline{Y}^2\right)}. \tag{2.4}$$

Here $E = 1-\rho^2$, where ρ is the Pearson correlation coefficient, and hence lies in the range $0 \le E \le 1$. For $E = 0$ (or equivalently $I = \infty$) we have perfect correlation between the input and output signal, while $E = 1$ ($I = 0$) corresponds to an output that is completely independent of the input.

2.2.2 Determining the enzymatic parameter range

Once the input signal is specified through F and γ_K, there are ten parameters related to the kinase, phosphatase, and substrate that determine the observables of interest γ_x, R_0, and I discussed above. These parameters are: κ_b, κ_u, κ_r, κ_{-r}, ρ_b, ρ_u, ρ_r, ρ_{-r}, $\bar{S}$, $\bar{P}$. We know from surveys of enzymatic parameters that each of these quantities can span several orders of magnitude among different systems, often with an approximately log-normal distribution [69, 70]. To understand the performance limits of enzymatic loops in general, it makes sense to explore the entire range of biologically realistic parameters, rather than focus on a single choice of parameters. Existing online databases are excellent resources for this purpose, and Fig. 2.2 shows the resulting histograms of kinase / phosphatase parameters (full extraction details are available in the SI). For the substrate protein (which we take as a kinase) and the phosphatase, the concentrations $[S]$ and $[P]$ in Fig. 2.2A are derived from the PaxDb protein abundance database [71], using UniProt gene ontology associations to identify kinases and phosphatases [72]. Enzymatic

reaction parameters are available in the Sabio-RK database [73]. The reaction rates κ_r and ρ_r (Fig. 2.2D) are typically listed directly, but the others are most often in specific combinations: the Michaelis constants $K_M^{\text{kin}} = (\kappa_r + \kappa_u)/\kappa_b$, $K_M^{\text{pho}} = (\rho_r + \rho_u)/\rho_b$ for kinase/phosphatase respectively (Fig. 2.2B) and the specificity ratios $\kappa_r/K_M^{\text{kin}}$, ρ_r/K_M^{pho} (Fig. 2.2C). For all of these parameters there is a paucity of data on phosphatases relative to kinases, but the phosphatase ranges seem to largely overlap with those of kinases. Thus for simplicity we take kinase and phosphatase parameters to have the same distributions (log-normal) and use a numerical fitting procedure to find an overall log-normal joint probability distribution for the eight underlying model parameters represented in the data: κ_b, κ_u, κ_r, ρ_b, ρ_u, ρ_r, $\bar{S}$, $\bar{P}$ (see SI). Note that data in concentrations units (like $[S]$ and $[P]$ in molars) is converted to mean abundances ($\bar{S}$ and $\bar{P}$) by assuming a volume of 30 fL (comparable to the cytoplasmic volume of yeast [5, 74]). This procedure is designed so that the resulting joint distribution yields marginal probability densities (solid curves in Fig. 2.2) that exhibit good agreement with the histogram data for any of the measured parameter combinations. Despite this agreement, we note that the joint distribution likely spans a portion of the parameter space larger than the true distribution of biological values: this is because it cannot fully capture correlations between different parameters. (Such correlations are difficult to reconstruct since many database entries are incomplete, containing some but not all of the enzymatic parameters.) For our purposes, having a distribution that effectively acts like a superset of the biological distribution is fine: whatever performance bounds we infer from the whole distribution will then also apply to the subset of the distribution that corresponds to current real-world systems. Moreover this also allows us to explore a larger enzymatic design space, which may have been accessible at earlier points in evolutionary history.

Two of the model parameters are still unaccounted for: the reverse reaction rates κ_{-r} and ρ_{-r}. Though usually small in magnitude and typically not measured in enzyme kinetic assays, we also know that they are crucially related to $\Delta\mu$ through the detailed balance relation of Eq. (2.1). Thus, as explained in the next section, these become important free parameters that we can vary to explore signaling efficiency and its dependence on $\Delta\mu$.

2.3 Results

2.3.1 Minimum cost of transmitting information

Given the model described above, with a parameter set drawn at random from the empirical joint distribution, we can ask a basic first question: what is the minimum chemical potential difference $\Delta\mu$ required to achieve a certain mutual information I? The answer will depend on the nature of the input signal $X(t)$, and thus we would like to test different effective input frequencies γ_x. To do this we will fix the mean free kinase concentration at the level of a low amplitude input, $[K] = 5$ nM, and vary γ_K, which varies γ_x according to Eq. (2.2) with $F = \gamma_K \bar{K}$ for fixed $\bar{K}$. In the SI we also show the same analysis for $[K] = 0.5$ and 50 nM, with results qualitatively similar to those described below. After drawing enzyme parameters from the joint distribution and specifying γ_x at a given $[K]$, the only two free parameters are the reverse reaction rates κ_{-r} and ρ_{-r}.

Fig. 2.3A shows a contour diagram of I as a function of κ_{-r} and ρ_{-r} for a sample enzyme parameter set and value of γ_x. Superimposed are dotted lines of constant $\Delta\mu$ from Eq. (2.1). If one were interested in achieving a particular I value, for example $I = 1$ bit, one can then numerically determine the κ_{-r} and ρ_{-r} point along the $I = 1$ bit contour where $\Delta\mu$ is smallest. For this specific enzyme parameter set and γ_x, the value turns out

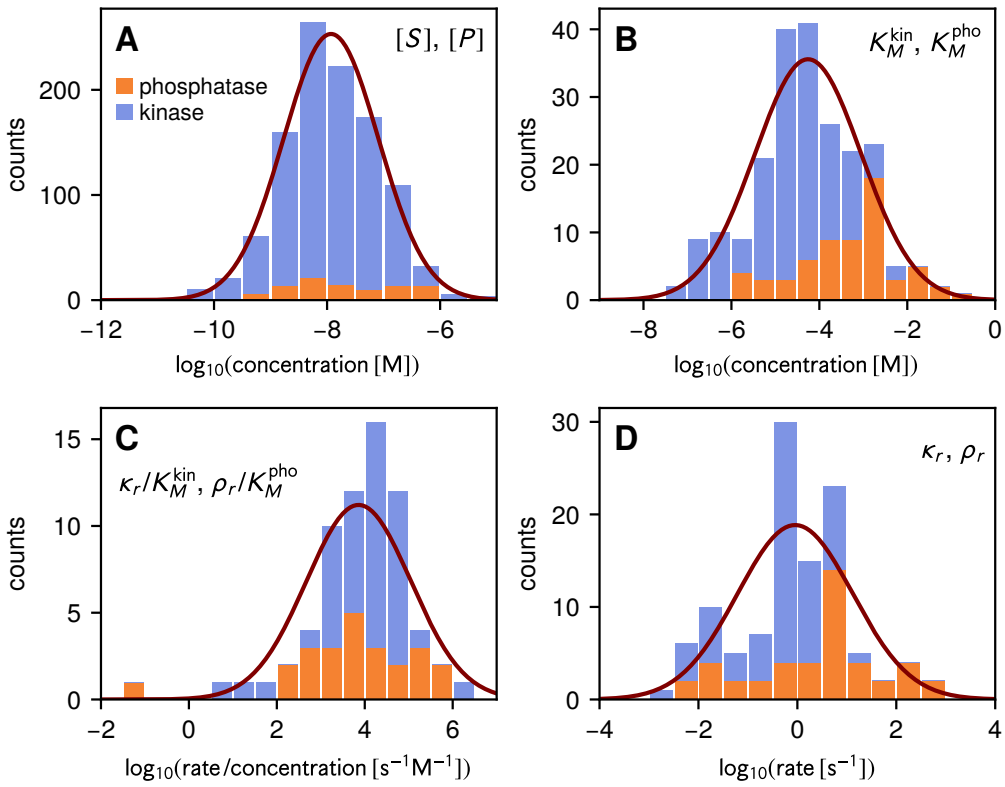

Figure 2.2. Enzymatic parameter ranges for kinases/phosphatases based on the PaxDb [71] and Sabio-RK [73] databases. Because of the relative lack of phosphatase data (orange histograms) relative to kinases (blue histograms), we fit an overall log-normal joint probability to the total data set including both kinases and phosphatases. The marginal distributions from that global fit are plotted as purple curves. The parameters are as follows: **(A)** kinase substrate $[S]$ and phosphatase $[P]$ concentrations; **(B)** kinase/phosphatase Michaelis constants K_M^{kin}, K_M^{pho}; **(C)** the corresponding specificity ratios $\kappa_r/K_M^{\text{kin}}$, ρ_r/K_M^{pho}; **(D)** kinase/phosphatase catalytic rates κ_r and ρ_r.

to be $\Delta\mu = 6.72\ k_BT$, which would then be recorded as the minimum necessary $\Delta\mu$ to achieve 1 bit of MI. Note that it is not guaranteed that a minimum $\Delta\mu$ solution exists for every parameter set sampled from the joint distribution. If the I contours plateau at a maximum less than 1 bit, no possible $\Delta\mu$ will allow that particular system to achieve the desired MI target. We will return to this important point below.

If one keeps the enzyme parameters (other than κ_{-r} and ρ_{-r}) fixed, and just varies γ_x, an interesting trend appears in the minimum $\Delta\mu$ results. Fig. 2.3B shows two examples of minimum $\Delta\mu$ curves, for target I values of 1 and 2 bits respectively. For a given I target, the minimum $\Delta\mu$ is nearly constant at low input frequencies, but then increases rapidly and diverges at a maximum frequency which we will dub the "bandwidth" of the system. This intuitively makes sense: the higher the input frequency, the more rapid the catalytic reaction rates needed to accurately transmit the signal through the system, increasing the required $\Delta\mu$ threshold. However there is an inherent limit, given finite enzyme catalysis rates. Above the bandwidth, whose value depends on the enzyme parameters, the system can no longer achieve the target I. The higher the informational burden (i.e. increasing the target I from 1 to 2 bits) the lower the bandwidth: if one desires higher fidelity transmission, the range of transmissible signal frequencies will suffer.

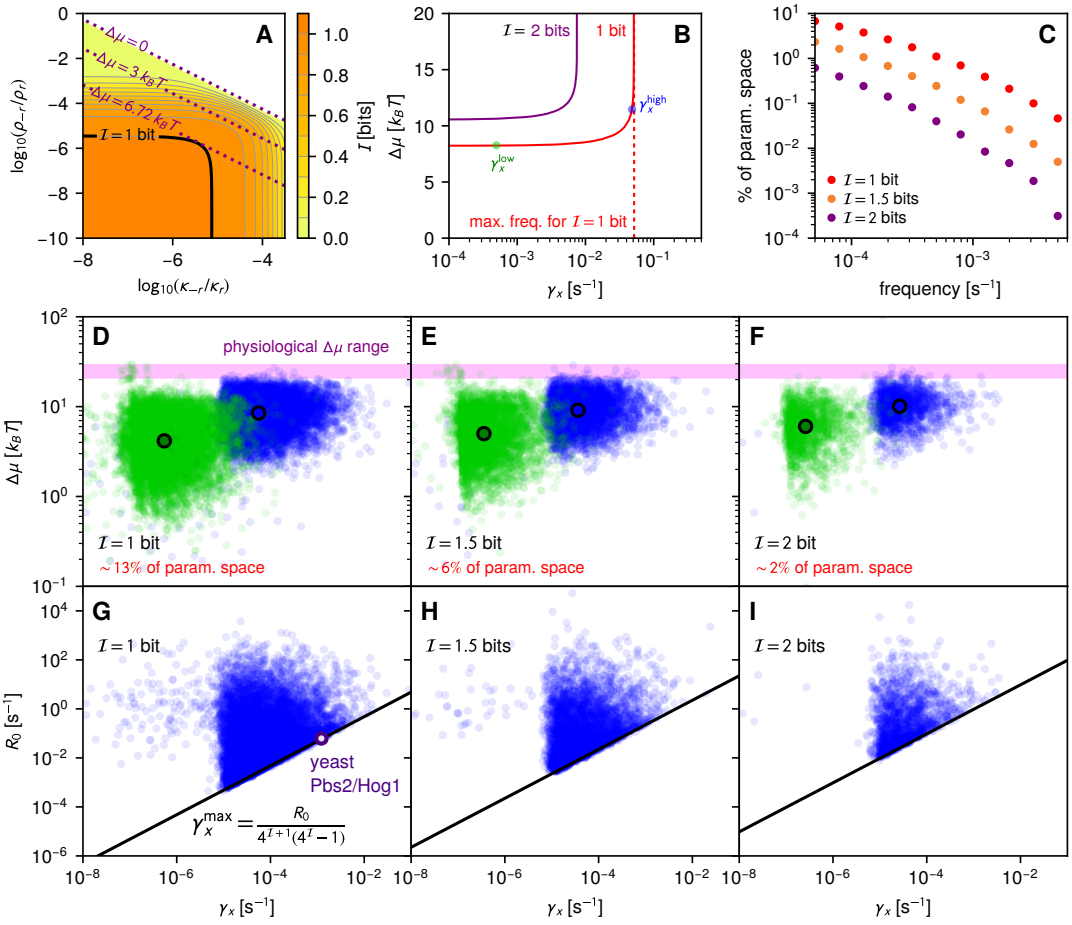

Figure 2.3. **(A)** A representative contour diagram of I (solid curves) as a function of κ_{-r} and ρ_{-r} for a parameter set drawn randomly from the joint distribution. Dotted lines denote contours of constant $\Delta\mu$. In this case $\Delta\mu = 6.72\ k_B T$ is the smallest value at which the system can achieve $I = 1$ bit. **(B)** For a sample parameter set, the minimum $\Delta\mu$ needed to achieve $I = 1$, 2 bits as a function of input frequency γ_x. For the 1 bit case, the dashed line represents γ_x^{high}, the maximum γ_x compatible with $I = 1$ bit for this parameter set. As described in the text, we highlight two points along the curve: one at a frequency γ_x^{high} at roughly 95% of the bandwidth, and the other at frequency γ_x^{low} at roughly 1% of the bandwidth. The points will be plotted for a many random draws of the enzyme parameters from the joint distribution in the lower panels of the figue. **(C)** For each target value of $I = 1$, 1.5, 2 bits, the percentage probability of randomly drawing a parameter set that has a γ_x^{high} higher than a given frequency. **(D-F)** The distribution of γ_x^{high} (blue) and γ_x^{low} (green) for many random parameter draws, keeping only those that can achieve $I = 1$ bit (D), 1.5 bits (E), or 2 bits (F). The probabilities of successfully drawing such a set are shown in red in each panel. The blue and green circles denote the median of each distribution respectively. **(G-I)** The same γ_x^{high} distributions as in panels (D-F), except plotted in terms of gain R_0 on the vertical axis. The solid line is the analytical maximum bandwidth bound γ_x^{max} of Eq. (2.5). The purple circle in panel G shows the estimated result for the near-optimal yeast Pbs2/Hog1 system.

To make more sense of these results, it is useful to look at a broad sample of enzyme parameters rather than a single set. To visualize global behaviors, we will calculate two numerical results for each set drawn from our joint distribution. The procedure is as follows: i) Sample an enzyme parameter set from the distribution; ii) Determine if it can achieve our target I for any input frequency; iii) If the answer is yes, vary γ_K until one finds the maximum possible value $\gamma_K^{\max}$ where one can still achieve the I target. iv) Calculate the minimum $\Delta\mu$ for an input signal very near the bandwidth frequency, where $\gamma_K = 0.95\gamma_K^{\max}$. We will call this result $\Delta\mu^{\text{high}}$. The corresponding input frequency is γ_x^{high}. v) Analogously, calculate the minimum $\Delta\mu$ for an input signal with a frequency much lower than the bandwidth, where $\gamma_K = 0.01\gamma_K^{\max}$. This set of results we denote as $\Delta\mu^{\text{low}}$ and γ_x^{low}. Fig. 2.3B shows the two points $(\gamma_x^{\text{high}}, \Delta\mu^{\text{high}})$ and $(\gamma_x^{\text{low}}, \Delta\mu^{\text{low}})$ as blue and green dots respectively for that particular parameter set at $I = 1$ bit. These two points encapsulate several key features of the minimum $\Delta\mu$ versus γ_x curve: $\Delta\mu^{\text{low}}$ roughly corresponds to an "entry level" price, the minimum ATP hydrolysis chemical potential necessary to transmit the signal at any frequency, while the difference $\Delta\mu^{\text{high}} - \Delta\mu^{\text{low}}$ is the premium one has to pay to transmit signals near the highest possible frequencies. The value γ_x^{high} approximately corresponds to the bandwidth.

If one were to make numerous draws from the parameter distribution, and plot $(\gamma_x^{\text{high}}, \Delta\mu^{\text{high}})$ and $(\gamma_x^{\text{low}}, \Delta\mu^{\text{low}})$ for each draw, one would get a cloud of blue and green dots. These are shown in Fig. 2.3D-F for target I of 1, 1.5, and 2 bits respectively. As mentioned above, not every draw will lead to a parameter set that can achieve the target, and the plots are labeled by the fraction of draws that are capable of reaching that particular value of I. That fraction decreases with I, from 13% for $I = 1$ bit down to

only 2% for $I = 2$ bits. As I increases not only does it become progressively more difficult to find enzymatic parameters compatible with higher fidelity, but the accessible frequency range becomes more restricted. Fig. 2.3C shows the percentage of the parameter space that can achieve bandwidths higher than a given frequency for different I. For example let us consider the frequency 1.22×10^{-3} s^{-1}, which is the bandwidth estimated for the Hog1 signaling pathway in yeast using periodic osmolyte shocks [75]. Note that this is the bandwidth for the entire pathway, which must be a less than or equal to the bandwidth of the individual enzymatic loops that compose the pathway. From Fig. 2.3C it is evident that only about 0.41% of the draws from the parameter distribution have $\gamma_x^{\text{high}} \geq 1.22 \times 10^{-3}$ s^{-1} for a target $I = 1$ bit. If one were to attempt to transmit signals at such high frequencies for $I = 2$ bits, the fraction of compatible parameter space shrinks to a miniscule 9×10^{-3}%. This reflects the exquisite fine-tuning required to put together a set of enzymatic loops capable of responding to quick, life-or-death variations of the external environment on time scales of a couple of minutes. Going much beyond $I = 1$ bit and maintaining fast response times for a single push-pull loop is extremely difficult, and hence it makes sense that biology settles for I in the vicinity of 1 bit in many circumstances. Going much below 1 bit poses another set of difficulties, since such systems would not even be able to reliably transmit the difference between high and low values of input signal. For signaling that can occur over longer timescales (hours instead of minutes) it becomes much easier to find compatible parameter sets, with the median of the distribution of γ_x^{high} for $I = 1$ bit around $\sim 6 \times 10^{-5}$ s^{-1}.

From the perspective of costs, the bulk of the distribution of entry level prices $\Delta\mu^{\text{low}}$ for $I = 1$ bit is $\gtrsim 1\ k_B T$. Any system much below this would be too close to equilibrium (reverse rates comparable to forward rates) for effective information transfer to occur.

The median of the $\Delta\mu^{\text{low}}$ distribution in Fig. 2.3C is 4 k_BT, increasing to about 6 k_BT for $I = 2$ bits in Fig. 2.3E. These values are on the same scale as estimates of minimum $\Delta\mu \sim 4\ k_BT\ln 2$ required for 99% accurate readout of a ligand-bound receptor via the activation of a downstream molecule, assuming an arbitrarily slow readout process [55]. In that system (as in ours), processing information at faster time scales requires large $\Delta\mu$. Indeed we find that the median values for $\Delta\mu^{\text{high}}$ range between $8-10\ k_BT$ for $I = 1-2$ bits. The minimum $\Delta\mu$ near the bandwidth is typically shifted up by about 4 k_BT, reflecting the premium necessary to transmit near the frequency limit. Paying this premium is worthwhile: frequencies γ_x^{low} accessible at $\Delta\mu^{\text{low}}$ prices are likely far too low to have biological relevance, with the distributions of γ_x^{low} largely below 10^{-5} s^{-1}. To get the ability to respond to signals at more biologically reasonable time scales thus means being capable of transmitting closer to the bandwidth, making $\Delta\mu^{\text{high}}$ a more useful measure of minimum biological costs.

The $\Delta\mu^{\text{high}}$ distributions show that it is possible to have signaling systems that transmit at least 1 bit of MI and operate at $\Delta\mu$ lower than the current physiological range ($\Delta\mu \approx 21-29\ k_BT$ [5], indicated in pink in Fig. 2.3D-F). This is true even for systems with the fastest responses (large γ_x^{high} near the right edges of the distribution). This means the one can imagine enzymatic signaling systems in the earliest stages of evolutionary history that can reliably distinguish high and low inputs even before ATP metabolism (maintaining high ATP concentrations relative to ADP and P_i) reached its modern levels of efficiency.

In fact a fascinating universal feature of the distributions is that the physiological $\Delta\mu$ range lies just above the top edge of the distributions. Naively it would seem as if the physiological values are just high enough to allow these signaling loops to transmit

$I = 1-2$ bits across the broadest possible parameter subset. This gives evolution the largest possible space in which to tweak tradeoffs between fidelity and response times without running into chemical potential limitations. Of course $\Delta\mu$ influences not just signaling networks but the entire range of cellular functions, so it is impossible to say with certainty what factors played the largest role in determining the values of $\Delta\mu$ we see in present-day organisms. But at least from the perspective of signaling at the level of a push-pull loop, it is clear that $\Delta\mu \approx 21-29\ k_BT$ is more than good enough for basic information transfer needs, and there would be no benefit in having a system with substantially higher $\Delta\mu$. To maintain $\Delta\mu = 40$ or $50\ k_BT$ for example, would require significant additional metabolic resources, with little payoff in terms of either I or bandwidth.

2.3.2 Analytical bound describes tradeoff between bandwidth and information

The results above already illustrated the tradeoff between bandwidth and MI, with parameter sets that achieve very large γ_x^{high} becoming progressively harder to find as the target I increases. Can we understand this relationship in more detail? For this purpose we take advantage of optimal noise filter theories, originally developed in the context of signal processing [76–78], and in recent years applied to a variety of biological signaling networks [61, 62, 79–82]. The original motivation involved designing a filter for a signal corrupted by noise, such that the output matched the uncorrupted input signal as closely as possible. In the biological context, this same framework allows us to put bounds on the maximum MI achievable between input and output signals for given input and enzymatic parameters. As shown in the SI, our enzymatic push-pull loop can be approximately mapped onto an effective two-species input-output system, which is

then amenable to analytical treatment using the Wiener-Kolmogorov optimal filter theory [61, 76–78].

The end result is a remarkably simple analytical relation between the maximum possible bandwidth $\gamma_x^{\max}$ achievable given a target value of I,

$$\gamma_x^{\max} = \frac{R_0}{4^{I+1}(4^I - 1)}. \tag{2.5}$$

The only other enzymatic parameter that appears in the relation is the gain R_0, a measure of output production relative to the input. Fig. 2.3G-I shows the same parameter set distribution as the $(\gamma_x^{\text{high}}, \Delta\mu^{\text{high}})$ points in Fig. 2.3D-F, except replotted in terms of $(\gamma_x^{\text{high}}, R_0)$, where R_0 is the gain for each parameter set. The solid line is the bound of Eq. (2.5). Even though this bound is based on an approximation of the full enzymatic system, and hence is not guaranteed to be exact, it still provides an excellent cutoff for the distribution of $(\gamma_x^{\text{high}}, R_0)$ points. For systems at a certain R_0, we see that as I is increased and the denominator in Eq. (2.5) gets larger, the maximum bandwidth $\gamma_x^{\max}$ shifts to lower values. If we are interested in a fast response time, increasing I systematically reduces the compatible parameter space, since we are forced to rely on cases with larger and larger R_0. Thus Eq. (2.5) rationalizes the earlier observation of limited options for networks that can simultaneously respond to signals fluctuating on minute time scales and achieve I significantly larger than 1 bit.

2.3.3 Optimality and the yeast Pbs2/Hog1 push-pull loop

There is an alternative way of thinking about the R_0 versus γ_x^{high} results in Fig. 2.3G-I. Imagine a system working at γ_x^{high} with a certain gain parameter R_0 and achieving a target value I. Comparing other parameter sets with the same bandwidth γ_x^{high} and target I (taking a vertical slice of one of the panels in Fig. 2.3G-I), they will have a variety

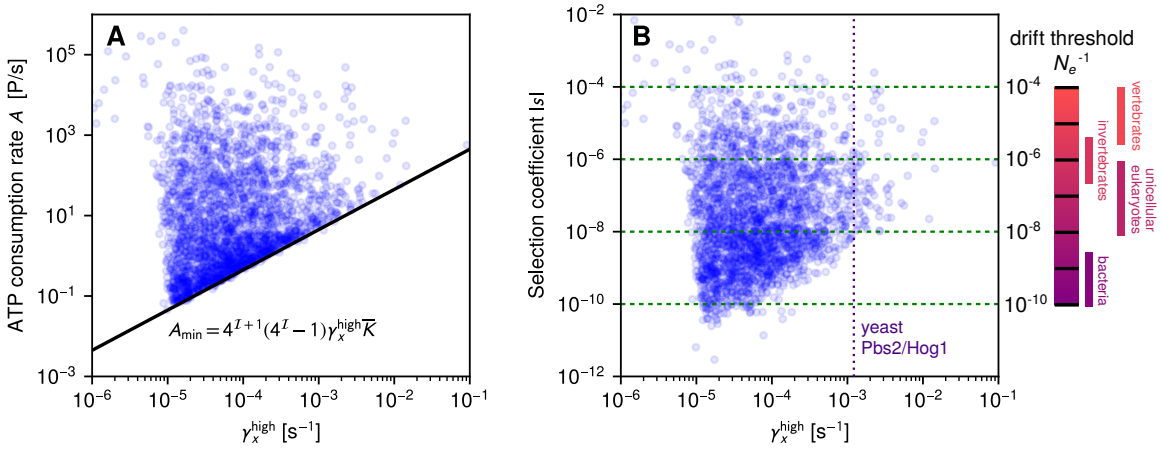

Figure 2.4. **(A)** The same $(\gamma_x^{\text{high}}, \Delta\mu^{\text{high}})$ point distribution as in Fig. 2.3C for $I = 1$ bit, except plotted in terms of ATP consumption rate A on the vertical axis. The solid line is the approximate lower bound $A_{\min}$ on ATP consumption given by Eq. (2.7). **(B)** This distribution replotted with selection coefficient $|s|$ on the vertical axis. $|s|$ quantifies the fitness cost associated with a system that achieves the target $I = 1$ bit but is sub-optimal in ATP consumption, relative to an optimal variant where $A = A_{\min}$. The value of $|s|$ becomes evolutionarily significant when it is higher than a "drift threshold" N_e^{-1}, where N_e is the effective population of the organism (a measure of genetic diversity). The ranges of N_e^{-1} for different classes of organisms are shown on the right [63, 83]. The vertical dotted line corresponds to the estimated γ_x^{high} for the yeast Pbs2/Hog1 system.

of different R_0 values, but all of these will be bounded from below by the minimum value

$$R_0^{\min} = 4^{I+1}(4^I - 1)\gamma_x^{\text{high}}. \tag{2.6}$$

When $R_0 = R_0^{\min}$, the system sits on the optimality line of Eq. (2.5), with $\gamma_x^{\text{high}} = \gamma_x^{\max}$.

The discrepancy between R_0 and $R_0^{\min}$ for a given system allows us to see how close the signaling behavior is to optimality. Let us take a concrete biological example: the Pbs2/Hog1 enzymatic push-pull loop from yeast, part of the Hog1 signaling pathway

that allows the organism to respond to osmotic stress. As described in the SI, key parameters for this system can be estimated based on an earlier model [74] fit to microfluidic experimental data where yeast was exposed to periodic salt shocks [84]. The results for the bandwidth and gain for $I = 1$ bit are: $\gamma_x^{\text{high}} = (1.22 \pm 0.04) \times 10^{-3}$ s^{-1} and $R_0 = 0.0621 \pm 0.0001$ s^{-1}, with the error bars reflecting uncertainties due to unknown parameters (where we used priors based on the log-normal distributions Fig. 2.2.) The scale of the predicted bandwidth γ_x^{high} is consistent with microfluidic estimates. Ref. [75] found a steep dropoff in the mean amplitude of the Hog1 response to periodic step-like changes in external osmolyte concentrations when the frequencies of the changes increased from 10^{-3} s^{-1} to 10^{-2} s^{-1}. At frequencies beyond the dropoff the Hog1 output can no longer reproduce the osmolyte input at high fidelity. Though the form of the input in this case is different than in our model, and the experiment probes the entire pathway rather than just the Pbs2/Hog1 component, the similarity in scales to our γ_x^{high} value suggests that the Pbs2/Hog1 system may play a major role in determining the bandwidth of the whole pathway (since the bandwidth of the whole is constrained by the bandwidths of the components).

Intriguingly, the estimated gain R_0 is very close to the minimum possible value R_0^{min} for signaling at the bandwidth γ_x^{high} with $I = 1$ bit, as seen in Fig. 2.3G. Using Eq. (2.6), we find $R_0^{\text{min}} = 0.059 \pm 0.002$ s^{-1}. This naturally leads to the question: is the fact that this system lies so close to optimality a coincidence, or are there reasons why natural selection might favor minimizing R_0 in this case? To answer this question, we first have to consider the relationship between gain and ATP consumption.

2.3.4 Minimum ATP consumption to achieve a certain signaling fidelity and bandwidth

This bound on the gain parameter in Eq. (2.6) is directly related to the metabolic cost of signaling, since higher production of the output per given input level will generally require a higher rate of phosphorylation events. We can roughly quantify the average rate of phosphorylation: in the stationary state this is just the mean rate of the kinase-catalyzed reaction step, $A = \kappa_r \overline{S_K}$. Assuming one ATP hydrolyzed per reaction, A is the mean rate at which ATP is consumed by the system, and is related to R_0 through $A = \kappa_r R_0 \overline{K}/(\kappa_r - R_0)$, as shown in the SI. In the enzymatic parameter ranges we consider, κ_r is typically much larger than R_0, so we can approximate this relation as $A \approx R_0 \overline{K}$. Using Eq. (2.6) we can then estimate the minimum possible ATP consumption rate given a target I and bandwidth γ_x^{high}:

$$A_{\text{min}} \approx R_0^{\text{min}} \bar{K} = 4^{I+1}(4^I - 1)\gamma_x^{\text{high}} \overline{K}. \tag{2.7}$$

Fig. 2.4A shows the same parameter set values as the $(\gamma_x^{\text{high}}, \Delta\mu^{\text{high}})$ points in Fig. 2.3D for $I = 1$ bit, except plotted in terms of $(\gamma_x^{\text{high}}, A)$. The A values are exact, but the approximate relation of Eq. (2.7) provides an excellent lower bound on the distribution. Qualitatively, the individual elements of Eq. (2.7) all make intuitive sense. An increase in any of the constituent factors (the mean free input kinase population $\overline{K}$, the target information I, the bandwidth γ_x^{high}) puts greater demands on the signaling system, requiring more catalytic activity and hence faster ATP consumption. Note that the above results are easily generalized if the reaction step consumes more than one ATP: for example the effective model for yeast Pbs2/Hog1 discussed above involves phosphorylation at two sites, which would lead to the expressions for A and A_{min} getting a prefactor of two.

2.3.5 Evolutionary pressure on the metabolic costs of signaling

It is clear from Fig. 2.4A that for many parameter set choices the ATP consumption rate A is significantly larger than for a system near optimality ($A \approx A_{\mathrm{min}}$) given the same I and γ_x^{high}. Let us consider a specific scenario where the bandwidth γ_x^{high} and the target I are sufficient for the biological function of the signaling i.e. there are rapidly diminishing fitness returns in going to higher bandwidth and signal fidelity. In this scenario a system with $A > A_{\mathrm{min}}$ has no significant adaptive advantage over one with $A \approx A_{\mathrm{min}}$, but instead incurs a fitness penalty because of the superfluous ATP consumption. Would there be evolutionary pressure on this sub-optimal system to move toward optimality?

The answer to this question has practical ramifications, because it will allow us to predict whether we should expect to see natural enzymatic push-pull loops cluster around the optimality line (as we saw in the yeast Pbs2-Hog1 example). The alternative, in the absence of strong evolutionary pressure to optimize, is a wider dispersion, more similar to Fig. 2.4A where the points are drawn at random from the enzymatic parameter distribution. Note that this is a question that is directly amenable to future kinetic experiments: for systems where we can fully characterize the enzymatic parameters of the push-pull loop (for both the kinase and phosphatase), all the relevant quantities like γ_x^{high}, A, and I can be calculated.

Naively one might expect evolution to always drive systems to optimality due to natural selection, but genetic drift can play a significant competing role, allowing sub-optimal variants to flourish and even fix in a population [85]. To be specific, let us consider a unicellular organism that reproduces via binary fission, and two genetic variants of that organism that differ in the enzymatic parameters of a push-pull signaling loop: both variants achieve the same γ_x^{high} and I, but one has $A > A_{\mathrm{min}}$ and one has $A = A_{\mathrm{min}}$.

Let us denote the relative fitness of the sub-optimal versus the optimal type as $1+s$, defining a selection coefficient s. In other words the sub-optimal variant will have on average $1+s$ offspring relative to the optimal one during the generation time of the optimal type. In the scenario described above, where the extra production does not confer any adaptive advantage and only imposes a metabolic cost, we will have $s<0$, because the superfluous ATP consumption will lead to slower growth.

The magnitude of s determines the degree of selective pressure on the sub-optimal variant. The key quantity that sets the relevant scale for s is the effective population N_e of the organism, the size of an idealized population that exhibits the same changes in genetic diversity per generation due to drift as the actual population [83]. When $s<0$ and $|s| \gg N_e^{-1}$, natural selection dominates drift, exponentially suppressing the probability of a sub-optimal mutant fixing in a population of optimal organisms. On the other hand if $|s| \ll N_e^{-1}$, drift dominates, and the fixation probability of sub-optimal mutants is roughly the same as for a neutral ($s=0$) mutation [86]. In this case it would be difficult to maintain optimality in a population over the long term. N_e for organisms is typically smaller than their actual population in the wild, and varies by several orders of magnitude among different classes: for unicellular species it can be as high as $\sim 10^9 - 10^{10}$ in bacteria down to $\sim 10^6 - 10^8$ in single-celled eukaryotes [63, 83]. (It becomes even smaller among higher eukaryotes, going down to $\sim 10^4$ in vertebrates.) The corresponding ranges for the "drift threshold" N_e^{-1} [63] are shown on the right in Fig. 2.4B.

The question then becomes: how do we estimate s and how does it compare to the relevant N_e^{-1} for the class of interest? For the case where a variant imposes metabolic costs but no adaptive advantage, there is a very useful relation that posits $s \sim -\delta C_T / C_T$ [63, 87, 88]. Here C_T is the total resting metabolic expenditure of an organism

during a generation time, measured for example in units of P, where 1 P = one phosphate bond hydrolyzed (ATP or ATP equivalent consumed). δC_T is the extra expenditure incurred by the more costly mutant. This relation has already been used to explore selective pressures in yeast [88], unicellular prokaryotes and eukaryotes [63], and viral infections [89]. It was recently derived from first principles through a general bioenergetic growth model [64], where the relation was refined with a more accurate prefactor: $s \approx -\ln(R_b)\delta C_T / C_T$. Here R_b is the mean number of offspring per individual (i.e. $R_b = 2$ for binary fission).

The value of C_T can be readily estimated for single-celled organisms, where it scales roughly with cell volume [63, 64]. Given the 30 fL cell volume used in our calculations, and assuming a generation time (cell division time) $t_r = 1$ hr, we find $C_T \approx 7 \times 10^{11}$ P (see details in the SI), comparable in magnitude to experimental estimates for yeast [63]. Since δC_T reflects the extra ATP consumed by the costly mutant (with consumption rate A) versus the optimal variant (rate $A_{\min}$) over one generation time, we can write $\delta C_T = (A - A_{\min})\, t_r$. We can thus calculate s for all the near-bandwidth $I = 1$ bit parameter sets represented in Fig. 2.4A. The results for $|s|$ versus γ_x^{high} are plotted in Fig. 2.4B. Because increased ATP consumption is required to achieve larger bandwidths (as seen in Eq. (2.7)), the distribution of selective penalties $|s|$ for being sub-optimal is pushed to larger values with greater γ_x^{high}. In other words, higher bandwidths make the energetic stakes more significant.

We can now rationalize why the yeast Pbs2/Hog1 loop might be close to optimality. The bandwidth for that system (indicated by a vertical dashed line in Fig. 2.4B) is near the higher end of the spectrum. Suboptimal parameter values that achieve approximately the same bandwidth at $I = 1$ bit span a range of $|s|$ values between 10^{-8} and 10^{-4}.

Given $N_e = 10^6 - 10^8$ for single-celled eukaryotes [63, 83], and estimates of $N_e \approx 10^7$ for wild yeast populations [90], these suboptimal systems likely have $|s|$ near or above the drift threshold N_e^{-1}. Thus we would expect yeast to be under evolutionary pressure to optimize the energy expenditures associated with the enzymatic loop.

2.4 Discussion and Conclusions

The kinase-phosphatase push-pull signaling network, which maintains a certain value of mutual information I between input and output, incurs energetic costs in the form of ATP consumption. These costs have two related facets: (i) the free energy expenditure $\Delta\mu$ for each hydrolysis reaction, and (ii) the number of such reactions A per unit time. Achieving empirical values like $I = 1 - 2$ bits requires satisfying both aspects of the cost. There is a minimal price in terms of $\Delta\mu$ to achieve any given I, and this price increases if one demands either greater fidelity (larger I) or the ability to process faster signals (larger γ_x). Modern cells are more than willing to pay this part of the price, with $\Delta\mu$ sufficiently high to meet the minimal requirements for any enzymatic parameter set that hits a target I on the order of 1 bit. However, as the distributions in Fig. 2.3D-F illustrate, there are certainly options for signaling systems that work at similar fidelities under conditions of smaller $\Delta\mu$, the presumptive scenario earlier in evolutionary history. In all cases we require some degree of fine-tuning of enzymatic parameters: the higher the fidelity or frequency demands, the smaller the fraction of parameter space that satisfies them. This leaves vanishingly small room to achieve networks that operate at I significantly larger than the known empirical range.

For particular parameter combinations the system is optimal, exhibiting the maximum possible bandwidth ($\gamma_x^{\max}$ of Eq. (2.5)) with the minimal ATP consumption ($A_{\min}$

of Eq. (2.7)). Is such optimality widely realized in nature? Analyzing the selective pressures due to superfluous ATP expenditures indicates that this is a worthwhile question to pursue. We have already identified one near-optimal candidate in the yeast Hog1 signaling pathway. Based on the results of the previous section, we predict that the best place to look for others is among signaling pathways with high bandwidths, for example $\sim 10^{-3} - 10^{-2}$ s^{-1} at the extremes of the current biological distribution. Here the metabolic costs of being suboptimal would be significant for single-celled organisms.

More broadly, strong selective pressure on the costs of running signaling networks in single-celled organisms is likely to be a widespread phenomenon. To give another example, the expenditure of running the chemotaxis machinery in *E. coli* has been estimated to be about $\sim 10^7$ P per ~ 1 hr cell cycle [35, 37]. Compared to a value of $C_T \approx 2 \times 10^{10}$ P for *E. coli* [63, 64], we get an $|s| \sim 10^{-4}$, which is definitely significant for a bacterial population. We have barely begun to understand the kinds of optimization that such selective pressure has induced. Our approach readily generalizes beyond the kinase-phosphatase system, setting the stage for exploring these issues in a much wider array of biochemical networks.

2.5 Supplementary information for this chapter

2.5.1 Derivation of the detailed balance relation

To derive the detailed balance relation of Eq. (2.1), it is convenient to focus on the reactions from the perspective of an individual substrate molecule [4]. A given molecule in our model can be in one of four states, indicated in Fig. 2.5 with corresponding forward and reverse transition rates. For example if the molecule is an unmodified substrate

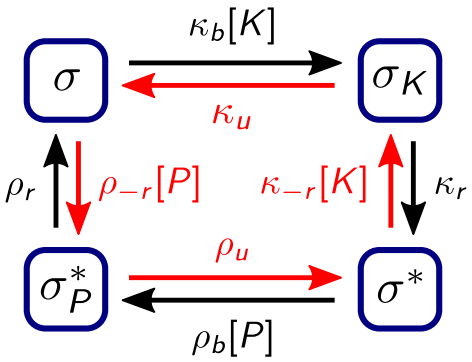

Figure 2.5. The enzymatic push-pull loop from the perspective of an individual substrate molecule. The protein can exist in one of four states: unmodified substrate (σ), bound to kinase (σ_K), phosphorylated (σ^*), and bound to phosphatase while phosphorylated (σ_P^*). The forward (clockwise) transition rates between these states are indicated in black, while the reverse (counterclockwise) rates are in red.

(state σ) it can transition to a kinase-bound substrate (state σ_K) with rate $\kappa_b[K]$, proportional to the surroudning concentration $[K]$ of kinase molecules. It can revert from σ_K to σ with rate κ_u. The other transitions in Fig. 2.5 are defined analogously, with forward rates colored black and reverse rates in red. Detailed balance entails that product of reverse rates divided by the product of forward rates is equal to $\exp(\beta\Delta G)$, where ΔG is the free energy change of the system associated with a single forward traversal of the loop and $\beta = (k_B T)^{-1}$ [4]. Since after one loop from σ to σ the substrate is back in the same state (as well as the kinase and phosphatase), there is no free energy contribution from these molecules. However a single loop leads to the hydrolysis of a single molecule of ATP, so $\Delta G = -\Delta\mu$, as defined in the main chapter text. Putting everything together, the detailed balance relation reads

$$e^{-\beta\Delta\mu} = \frac{\kappa_u}{\kappa_b[K]}\frac{\kappa_{-r}[K]}{\kappa_r}\frac{\rho_u}{\rho_b[P]}\frac{\rho_{-r}[P]}{\rho_r} = \frac{\kappa_{-r}\rho_u\rho_{-r}\kappa_u}{\kappa_r\rho_b\rho_r\kappa_b}, \tag{2.8}$$

yielding Eq. (2.1).

2.5.2 Chemical Langevin approach for the kinase-phosphatase push-pull loop

In this section we derive the stationary state properties of the kinase-phosphatase push-pull loop via the chemical Langevin approximation. The derivation will follow analogously to Ref. [61], except here the system is more complicated due to the inclusion of reverse enzymatic reactions. The end goal will be a method to estimate the mutual information I, given by Eq. (2.4),

$$I \approx -\frac{1}{2}\log_2 E, \quad \text{where} \quad E \equiv 1 - \frac{(\overline{XY} - \overline{X}\,\overline{Y})^2}{\left(\overline{X^2} - \overline{X}^2\right)\left(\overline{Y^2} - \overline{Y}^2\right)}, \tag{2.9}$$

which requires evaluating the variances of the input and output, $\mathrm{var}(X) = \overline{X^2} - \overline{X}^2$, $\mathrm{var}(Y) = \overline{Y^2} - \overline{Y}^2$, as well as the covariance $\mathrm{cov}(X,Y) = \overline{XY} - \overline{X}\,\overline{Y}$. The quantity E here will be referred to as the "error" in signal propagation between input and output, and can be equivalently expressed as $E = 1 - \rho^2$, where ρ is the Pearson correlation coefficient between X and Y.

Dynamical equations. Our starting point is the full system of reactions for the enzymatic push-pull loop,

$$\begin{aligned} &\varnothing \underset{\gamma_K}{\overset{F}{\rightleftharpoons}} K, \\ &K + S \underset{\kappa_u}{\overset{\kappa_b}{\rightleftharpoons}} S_K \underset{\kappa_{-r}}{\overset{\kappa_r}{\rightleftharpoons}} K + S^*, \\ &P + S^* \underset{\rho_u}{\overset{\rho_b}{\rightleftharpoons}} S_P^* \underset{\rho_{-r}}{\overset{\rho_r}{\rightleftharpoons}} P + S, \end{aligned} \tag{2.10}$$

where $\varnothing$ represents the void (upstream deactivated kinase which does not enter into our model). The corresponding chemical Langevin equations [7] are given by:

$$\begin{aligned}
\frac{dK}{dt} &= F - \gamma_K K - \kappa_b KS + (\kappa_u + \kappa_r) S_K - \kappa_{-r} K S^* + n_1 + n_2 + n_3, \\
\frac{dS_K}{dt} &= \kappa_b KS - (\kappa_u + \kappa_r) S_K + \kappa_{-r} K S^* - n_2 - n_3, \\
\frac{dS^*}{dt} &= \kappa_r S_K - \rho_b P S^* + \rho_u S_P^* - \kappa_{-r} K S^* + n_3 + n_4, \\
\frac{dS_P^*}{dt} &= \rho_b P S^* - (\rho_u + \rho_r) S_P^* + \rho_{-r} SP - n_4 + n_5, \\
\frac{dP}{dt} &= -\frac{dS_P^*}{dt}, \quad \frac{dS}{dt} = -\frac{dS_K}{dt} - \frac{dS^*}{dt} - \frac{dS_P^*}{dt},
\end{aligned} \tag{2.11}$$

where the last line ensures that the total populations of free or bound phosphatase ($P + S_P^*$) and free or bound substrate in all its forms ($S + S_K + S^* + S_P^*$) remain constant. The noise terms $n_i(t) = \sqrt{\Pi_i}\eta_i(t)$, where $\eta_i(t)$ are Gaussian noise functions with zero mean and correlations $\overline{\eta_i(t)\eta_j(t')} = \delta_{ij}\delta(t-t')$. The five noise terms are associated with reactions in the system, and the corresponding prefactors represent the sum of the mean production (forward) and deactivation/unbinding (backward) contributions to each reaction:

$$\begin{aligned}
&\Pi_1 = F + \gamma_K \overline{K}, \quad \Pi_2 = \kappa_b \overline{K}\,\overline{S} + \kappa_u \overline{S}_K, \quad \Pi_3 = \kappa_r \overline{S}_K + \kappa_{-r} \overline{K}\,\overline{S}^*, \\
&\Pi_4 = \rho_b \overline{P}\,\overline{S}^* + \rho_u \overline{S}_P^*, \quad \Pi_5 = \rho_r \overline{S}_P^* + \rho_{-r} \overline{S}\,\overline{P}.
\end{aligned} \tag{2.12}$$

Setting the left-hand sides of Eq. (2.11) to zero, and taking the average of the right-hand sides, we can solve for the stationary state populations:

$$\overline{K} = \frac{F}{\gamma_K}, \quad \overline{S}_K = \frac{FC_2}{\gamma_K C_1}, \quad \overline{S}^* = \frac{C_3}{C_1}, \quad \overline{S}_P^* = \frac{C_4}{C_1}. \tag{2.13}$$

with the following definitions:

$$\begin{aligned}
&\kappa_- \equiv \kappa_u + \kappa_r, \quad \rho_- \equiv \rho_u + \rho_r \\
&C_1 \equiv \kappa_- \overline{P}\gamma_K \rho_b \rho_r + F\kappa_{-r}\kappa_u \rho_- \\
&C_2 \equiv \overline{S}\left[F\kappa_b \kappa_{-r}\rho_- + \overline{P}\gamma_K(\kappa_b \rho_b \rho_r + \kappa_{-r}\rho_{-r}\rho_u)\right] \\
&C_3 \equiv \overline{S}(\overline{P}\gamma_K \kappa_- \rho_{-r}\rho_u + F\kappa_b \kappa_r \rho_-) \\
&C_4 \equiv \overline{P}\overline{S}\left[\overline{P}\gamma_K \kappa_- \rho_{-r}\rho_b + F(\kappa_b \rho_b \rho_r + \kappa_{-r}\rho_{-r}\kappa_u)\right]
\end{aligned} \tag{2.14}$$

The input (total kinase) is $X = K + S_K$ and the output (total activated substrate) is $Y = S^* + S_P^*$, and hence Eq. (2.13) can be used to calculate the stationary values $\overline{X}$ and $\overline{Y}$.

Second moments. In order to calculate the variance and covariance of the input and output, we also need to know $\overline{X^2}$, $\overline{Y^2}$, $\overline{XY}$. To estimate these quantities, the first step is to switch variables in Eq. (2.11) to focus on deviations from the stationary state values: $\delta K \equiv K - \overline{K}$, $\delta S_K \equiv S_K - \overline{S_K}$, $\delta S^* \equiv S^* - \overline{S}^*$, $\delta S_P^* \equiv S_P^* - \overline{S}_P^*$. We can in turn rewrite these four variables in terms of the input and output deviations $\delta X = X - \overline{X}$ and $\delta Y = Y - \overline{Y}$:

$$\begin{aligned}
\delta K &= \frac{C_1}{C_1 + C_2}\delta X + \delta X_q, \\
\delta S_K &= \frac{C_2}{C_1 + C_2}\delta X - \delta X_q, \\
\delta S^* &= \frac{C_3}{C_3 + C_4}\delta Y + \delta Y_q, \\
\delta S_P^* &= \frac{C_4}{C_3 + C_4}\delta Y - \delta Y_q,
\end{aligned} \tag{2.15}$$

where we have introduced two additional auxiliary variables δX_q and δY_q. Plugging Eq. (2.15) into Eq. (2.11), we simplify the system through linearization, ignoring any terms of second order or higher in the deviations. As demonstrated below in comparisons with kinetic Monte Carlo (KMC) simulations of the original system, this linearized chemical Langevin approximation works well for our parameter ranges. Finally,

we Fourier transform the linearized Eq. (2.11), and the resulting system of equations takes the form

$$M(\omega)\begin{pmatrix}\widetilde{\delta X}\\ \widetilde{\delta X}_q\\ \widetilde{\delta Y}\\ \widetilde{\delta Y}_q\end{pmatrix}=\begin{pmatrix}-\widetilde{n}_1\\ \widetilde{n}_2+\widetilde{n}_3\\ -\widetilde{n}_3-\widetilde{n}_5\\ \widetilde{n}_4-\widetilde{n}_5\end{pmatrix} \tag{2.16}$$

where $\widetilde{Q}(\omega)$ denotes the Fourier transform of quantity $Q(t)$. The matrix M is given by:

$$M(\omega)=$$

$$\begin{pmatrix} i\omega-\frac{C_1\gamma_K}{C_1+C_2} & \gamma_K & 0 & 0\\ \frac{C_1(\kappa_b\bar{S}+\kappa_{-r}\bar{S}^*)-C_2(\kappa_b\bar{K}+\kappa_{--}i\omega)}{C_1+C_2} & \kappa_b(\bar{K}+\bar{S})+\kappa_{-r}\bar{S}^*+\kappa_{--}i\omega & \frac{C_3\bar{K}\kappa_{-r}}{C_3+C_4}-\kappa_b\bar{K} & \bar{K}\kappa_{-r}\\ \frac{C_2\kappa_r-C_1\kappa_{-r}\bar{S}^*}{C_1+C_2} & -\kappa_r-\kappa_{-r}\bar{S}^* & -\frac{C_3\bar{K}\kappa_{-r}+C_4\rho_r}{C_3+C_4}+i\omega & \rho_r-\kappa_{-r}\bar{K}\\ 0 & 0 & \frac{C_3\bar{P}\rho_b-C_4(\bar{S}^*\rho_b+\rho_{--}i\omega)}{C_3+C_4} & \bar{P}\rho_b+\bar{S}^*\rho_b+\rho_{--}i\omega \end{pmatrix}. \tag{2.17}$$

The Fourier-space system of equations Eq. (2.16)-(2.17) can be solved for $\widetilde{\delta X}(\omega)$ and $\widetilde{\delta Y}(\omega)$. The expressions are complicated, but take the form of a linear combination of Fourier-space noise terms:

$$\widetilde{\delta X}(\omega)=\sum_{i=1}^{5}a_i^X(\omega)\widetilde{n}_i,\quad \widetilde{\delta Y}(\omega)=\sum_{i=1}^{5}a_i^Y(\omega)\widetilde{n}_i, \tag{2.18}$$

where $a_i^X(\omega)$ and $a_i^Y(\omega)$ are some prefactors which can be expressed as rational functions of ω. The prefactors have the property $a_i^X(-\omega)=(a_i^X(\omega))^*$, $a_i^Y(-\omega)=(a_i^Y(\omega))^*$. In Fourier space the correlations among the noise terms take the form $\overline{\widetilde{n}_i(\omega)\widetilde{n}_j(\omega')}=\delta_{ij}\Pi_i\delta(\omega+\omega')$. Hence we can calculate the input power spectral density (PSD) $P_X(\omega)$, the output PSD $P_Y(\omega)$ and the cross PSD $P_{XY}(\omega)$, defined via

$$\begin{aligned}&\overline{\widetilde{\delta X}(\omega)\widetilde{\delta X}(\omega')}=2\pi P_X(\omega)\delta(\omega+\omega'),\quad \overline{\widetilde{\delta Y}(\omega)\widetilde{\delta Y}(\omega')}=2\pi P_Y(\omega)\delta(\omega+\omega'),\\ &\overline{\widetilde{\delta X}(\omega)\widetilde{\delta Y}(\omega')}=2\pi P_{XY}(\omega)\delta(\omega+\omega').\end{aligned} \tag{2.19}$$

Plugging Eq. (2.18) into Eq. (2.19), we find expressions for the PSDs in terms of the prefactor functions:

$$P_X(\omega) = \sum_{i=1}^{5} |a_i^X(\omega)|^2 \Pi_i, \quad P_Y(\omega) = \sum_{i=1}^{5} |a_i^Y(\omega)|^2 \Pi_i, \quad P_{XY}(\omega) = \sum_{i=1}^{5} a_i^X(\omega) a_i^Y(-\omega) \Pi_i. \tag{2.20}$$

The final step is to calculate the second moments from integrals of the PSDs, using the inverse Fourier transform of Eq. (2.19) evaluated at $t = t'$:

$$\overline{X^2} = \int_{-\infty}^{\infty} \frac{d\omega}{2\pi} P_X(\omega), \quad \overline{Y^2} = \int_{-\infty}^{\infty} \frac{d\omega}{2\pi} P_Y(\omega), \quad \overline{XY} = \int_{-\infty}^{\infty} \frac{d\omega}{2\pi} P_{XY}(\omega). \tag{2.21}$$

Given the explicit expressions for the prefactor functions in Eq. (2.20) (which are available as part of the *Mathematica* notebooks in the Github repository associated with the manuscript), one can numerically evaluate the integrals in Eq. (2.21) to get the moments.

Comparison to kinetic Monte Carlo simulations for mutual information. The chemical Langevin calculation of the second moments allows us to use Eq. (2.9) to estimate the mutual information I. We can then check whether this estimate is consistent with the results we would get from KMC simulations of the full system. Fig. 2.6 shows this comparison for two sample parameter sets drawn from the enzymatic parameter distribution described in Sec. 2.5.4. Since we are interested in exploring the full range of chemical potentials $\Delta\mu$, in each case we calculate I varying the reverse-to-forward rate ratio κ_{-r}/κ_r, keeping all other parameters constant. Through Eq. (2.1), increasing κ_{-r}/κ_r corresponds to decreasing the magnitude of $\Delta\mu$. At very large $\Delta\mu$ (small κ_{-r}/κ_r) the I curves saturate at the maximum possible mutual information for that parameter set, while at small $\Delta\mu$ (large κ_{-r}/κ_r) the mutual information approaches zero, the equilibrium limit.

Across the whole range we see that the chemical Langevin theoretical prediction is in close agreement with the KMC results.

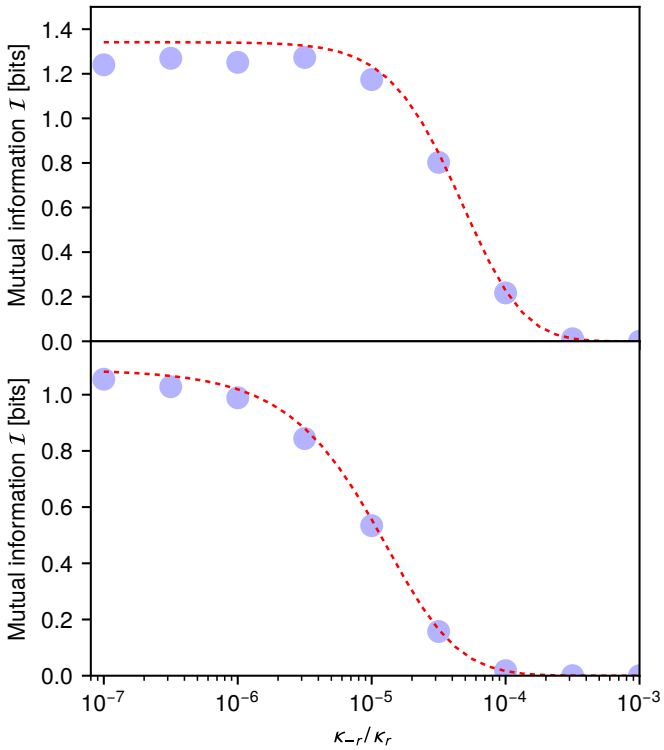

Figure 2.6. The mutual information I for the enzymatic push-pull loop as a function of the reverse-forward rate ratio $\frac{\kappa_{-r}}{\kappa_r}$. The predictions from the chemical Langevin approach (dashed line) are compared against the corresponding KMC simulation results (circles). The parameters sets are as follows (all units are s^{-1} except for the mean populations; molar units have been converted to populations by assuming a cell volume of 30 fL): (top) $\kappa_b = 2.94 \times 10^{-6}$, $\rho_b = 3.68 \times 10^{-7}$, $\kappa_u = 1.58 \times 10^{-2}$, $\rho_u = 4.42 \times 10^{-4}$, $\kappa_r = 12.8$, $\rho_r = 1.34$, $\rho_{-r} = 2.50 \times 10^{-5}$, $F = 2.49 \times 10^{-3}$, $\gamma_k = 2.68 \times 10^{-5}$, $\bar{S} = 614$, and $\bar{P} = 45$; (bottom) $\kappa_b = 2.32 \times 10^{-5}$, $\rho_b = 1.46 \times 10^{-4}$, $\kappa_u = 6.94 \times 10^{-2}$, $\rho_u = 5.48$, $\kappa_r = 0.994$, $\rho_r = 5.05 \times 10^{-2}$, $\rho_{-r} = 2.06 \times 10^{-8}$, $F = 2.46 \times 10^{-2}$, $\gamma_k = 2.65 \times 10^{-4}$, $\bar{S} = 2380$, and $\bar{P} = 127$.

2.5.3 Characteristic frequency γ_x, gain R_0, and the conditions for Wiener-Kolmogorov noise filter optimality

Deriving the γ_x and R_0 expressions in Eq. (2.2). Since the effective frequency γ_x of the input and the gain R_0 play central roles in the analysis, having simple closed form approximations for them [Eq. (2.2)] is useful. The original definitions of these two variables, as described in the main chapter text, are as follows: (i) γ_x is related to the autocorrelation of input fluctuations, $\overline{\delta X(t+\tau)\delta X(t)} = \overline{\delta X^2}\exp(-\gamma_x|\tau|)$; (ii) $R_0 \equiv \kappa_r \overline{S_K}/\overline{X}$ measures output production for a given level of input. As demonstrated in the next section, both of these can be calculated from KMC simulations (at significant computational expense for each different set of parameters). Alternatively, the chemical Langevin approximation of Sec. 2.5.2 can be used to derive somewhat cumbersome analytical expressions.

However the most convenient option is to take advantage of the meaning of γ_x and R_0 in an effective, two-species description of the kinase-phosphatase reaction network. Imagine a system with an input species population $X(t)$, output $Y(t)$, and a simplified chemistry with only four reactions: production of input at rate F, deactivation of input at rate $\gamma_x X(t)$, production of output at rate $R_0 X(t)$, and deactivation of output at rate $\gamma_y Y(t)$. In this two-species system the inverse input autocorrelation time is given by the deactivation rate parameter γ_x, and the coefficient R_0 in the output production rate is also the gain parameter. To relate this simplified model to the full reaction network of Sec. 2.5.2, we compare analogous quantities in the simplified and full schemes. For example, let us take the mean input population $\overline{X}$. In the simplified scheme this is given by

$$\overline{X} = \frac{F}{\gamma_x}. \tag{2.22}$$

In the full network $\overline{X} = \overline{K} + \bar{S}_K$ can be calculated from Eq. (2.13) as

$$\overline{X} = \frac{F(C_1 + C_2)}{C_1 \gamma_K}, \tag{2.23}$$

where the C_i are expressed in terms of full network parameters in Eq. (2.14). Comparing Eqs. (2.22) and (2.23) we see that γ_x should be given by

$$\gamma_x = \frac{C_1 \gamma_K}{C_1 + C_2}, \tag{2.24}$$

which is the first expression in Eq. (2.2). Similarly the mean production rate of the output in the simplified scheme is $R_0 \bar{X}$. In the full system the mean output production is the average rate at which new phosphorylated substrate is produced via catalysis by the kinase-substrate complex,

$$\kappa_r \bar{S}_K = \kappa_r \frac{F C_2}{\gamma_K C_1} = \kappa_r \frac{C_2}{C_1 + C_2} \bar{X}, \tag{2.25}$$

where we have again used Eqs. (2.13)-(2.14). Comparing Eq. (2.25) to $R_0 \bar{X}$, we see that R_0 should correspond to

$$R_0 = \kappa_r \frac{C_2}{C_1 + C_2}, \tag{2.26}$$

which is the second expression in Eq. (2.2).

Validation through kinetic Monte Carlo simulations. To verify that the expressions for γ_x and R_0 derived above are good approximations, we ran KMC simulations for various parameter sets drawn at random from the enzymatic parameter distribution detailed in the Sec. 2.5.4. For each parameter set the simulation was run long enough after reaching the stationary state to collect sufficient statistics for both the mean population values and the input autocorrelation function. As described above, these allow us to calculate γ_x and R_0. The simulation results are compared against the approximation from Eqs. (2.24) and (2.26) in Fig. 2.7. The agreement is excellent for both quantities, across

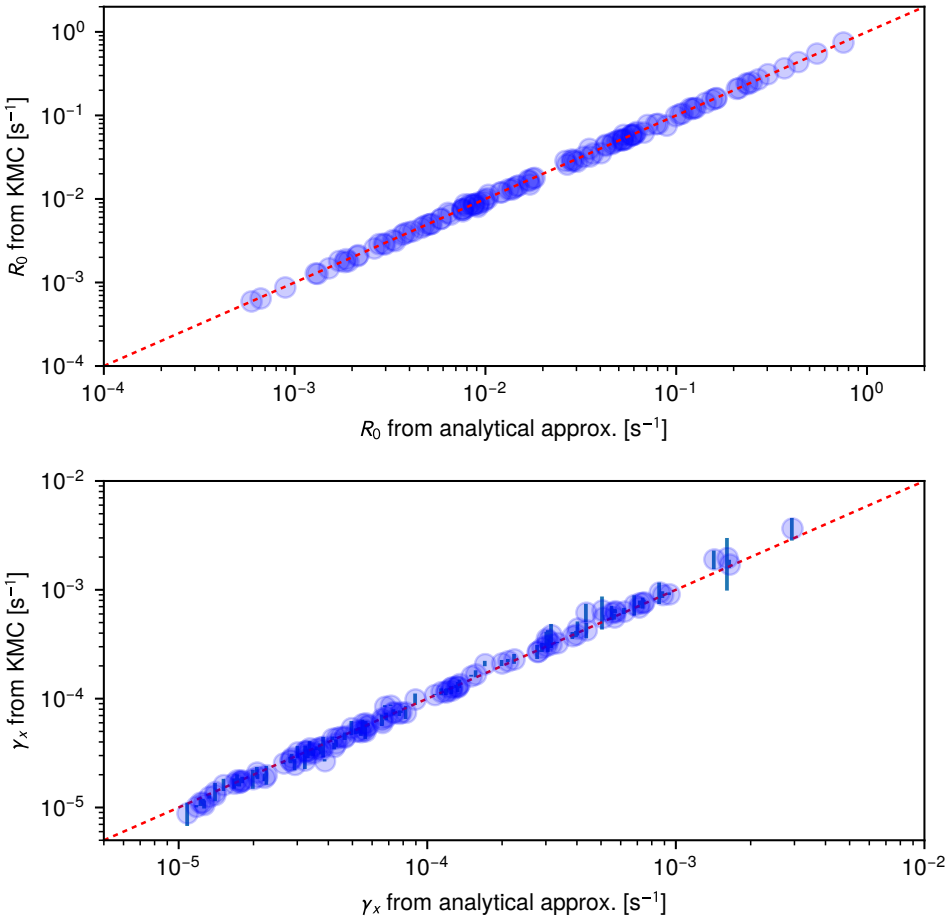

Figure 2.7. Comparison of the simple analytical approximations for R_0 from Eq. (2.26) (top) and γ_x from Eq. (2.24) (bottom) versus KMC simulation results. Each point corresponds to a parameter set drawn randomly from the enzymatic parameter distribution described in Sec. 2.5.4. Error bars for R_0 are smaller than the symbol size, and hence not indicated in the figure.

the entire range of γ_x and R_0 values. Thus we can confidently use the simple analytical expressions of Eqs. (2.24) and (2.26) to predict γ_x and R_0 for any given parameter set.

Relating maximum bandwidth, minimum ATP consumption rate, and mutual information via Wiener-Kolmogorov optimal noise filter theory. One of the benefits of the approximate relation between the full system and the two-species model described in Sec. 2.5.3 is that it allows us to use results from the two-species case to make predictions

for the behavior of the kinase-phosphatase push-pull loop. The two-species model has been analyzed in detail in Refs. [61, 82], where it was shown to be able to map onto a Wiener-Kolmogorov optimal noise filter. The error E from Eq. (2.9) for the two-species case can be evaluated in closed form as [61]:

$$E = 1 - \frac{\gamma_y^2 R_0}{(\gamma_x + \gamma_y)^2} \left[\gamma_y + R_0 \frac{\gamma_y}{\gamma_y + \gamma_x} \right]^{-1}. \tag{2.27}$$

It achieves its minimum value (hence maximizing the mutual information I) when the following condition is fulfilled:

$$\gamma_y = \gamma_x \sqrt{1 + \Lambda}, \tag{2.28}$$

where $\Lambda = R_0/\gamma_x$. The corresponding minimum E, where the system behaves like an optimal Wiener-Kolmogorov (WK) noise filter is given by:

$$E_{\mathrm{WK}} = \frac{2}{1 + \sqrt{1 + \Lambda}}. \tag{2.29}$$

Interestingly, this remains the bound even if we generalize the output production term $R_0 X(t)$ to be nonlinear in $X(t)$ [61]. Using the relation between E and I in Eq. (2.9), we can translate the bound $E \geq E_{\mathrm{WK}}$ into an equivalent statement that $\gamma_x \leq \gamma_x^{\max}$ at a given value of mutual information I. The value of $\gamma_x^{\max}$ is shown in Eq. (2.5):

$$\gamma_x^{\max} = \frac{R_0}{4^{I+1}(4^I - 1)}. \tag{2.30}$$

As shown in Figs. 2.3G-I, the above $\gamma_x^{\max}$ expression provides an excellent approximate upper bound on the γ_x^{high} values calculated for the full enzymatic system. Even though the effective two-species model lacks reverse rates, it provides a useful tool for deriving this bound, since the maximum bandwidth is achieved when the reverse rates are negligible (large $\Delta\mu$).

As mentioned in the discussion around Eq. (2.6), the expression for $\gamma_x^{\max}$ in Eq. (2.30) also has an alternative interpretation. This gives the minimum production rate $R_0^{\min}$ necessary to achieve mutual information I at a certain bandwidth γ_x^{high}:

$$R_0^{\min} = 4^{I+1}(4^I - 1)\gamma_x^{\text{high}}. \tag{2.31}$$

By relating R_0 in turn to the ATP consumption rate $A = \kappa_r \bar{S}_K$, we can convert Eq. (2.31) into an expression for the minimum necessary ATP consumption rate $A_{\min}$. To accomplish this, note that A can be rewritten as:

$$A = \kappa_r \bar{K} \frac{C_2}{C_1} = \kappa_r \bar{K} \frac{R_0}{\kappa_r - R0}, \tag{2.32}$$

where we have used Eqs. (2.13) and (2.26). Finally, taking advantage of the fact that typically $\kappa_r \gg R_0$ for the parameter distributions of interest, we make the approximation $A \approx R_0 \bar{K}$. This allows us to derive Eq. (2.7):

$$A_{\min} \approx R_0^{\min} \bar{K} = 4^{I+1}(4^I - 1)\gamma_x^{\text{high}} \bar{K}. \tag{2.33}$$

2.5.4 Enzymatic parameter distribution

Earlier surveys of enzymatic kinetic parameters in Refs. [69, 70], over broader classes than just kinases and phosphatases, showed that their distributions could be approximately described by log-normal distributions. For a given parameter x, we will denote this as $\log_{10} x \sim N(\log_{10} \tilde{x}, \sigma_x^2)$, or in other words that the base-10 logarithm of x is distributed according to a normal distribution with mean $\log_{10} \tilde{x}$ and standard deviation σ_x. The value $\tilde{x}$ is the median of the resulting log-normal distribution for x.

For our work the focus is on kinases and phosphatases, and we are interested in looking at the push-pull loop signaling behavior over the entire distribution of biologically plausible parameters. The parameter data we collected, summarized in the histograms

Parameter x	Unit	$\log_{10}\tilde{x}$	σ_x	Data source
direct fits to database values:				
kinase/phosphatase concentrations [S], [P]	[M]	-7.93	0.84	PaxDb [71]
Michaelis constants K_M^{kin}, K_M^{pho}	[M]	-4.26	1.21	Sabio-RK [73]
specificity ratios $\kappa_r/K_M^{\text{kin}}$, ρ_r/K_M^{pho}	[$\text{M}^{-1}\,\text{s}^{-1}$]	3.86	1.19	Sabio-RK [73]
reaction rates κ_r, ρ_r	[s^{-1}]	-0.04	1.16	Sabio-RK [73]
results of joint fitting:				
reaction rates κ_r, ρ_r	[s^{-1}]	-0.06	1.18	joint fit
binding rates κ_b, ρ_b	[$\text{M}^{-1}\,\text{s}^{-1}$]	3.94	1.12	joint fit
dissociation constants K_D^{kin}, K_D^{rho}	[M]	-7.00	1.31	joint fit

Table 2.1. Results of log-normal fits to various kinase/phosphatase enzymatic parameters. For each fit the mean $\log_{10}\tilde{x}$ and standard deviation σ_x are listed. The top rows of the table correspond to individual fits to parameters collected from the PaxDb and Sabio-RK databases. The bottom rows show the results of a joint fit, described in the text of Sec. 2.5.4.

of Fig. 2.2, had far more representation of kinases than phosphatases, which is a well known limitation of the existing experimental literature. Despite this sampling issue, the orders of magnitude spanned by phosphatase parameters were comparable to those of the kinases. For each parameter type, we thus decided to fit both types of enzyme with a single overall distribution, based on pooling of all the available kinase and phosphatase data together. The data available from the databases took the forms listed below (all raw data and the files used to process it are included in the Github repository associated with the manuscript). The mean $\log_{10}\tilde{x}$ and standard deviation σ_x values from the log-normal fits for the different parameter classes are listed in the first four rows of Table 2.1.

Enzymatic data:

- Mean substrate [S] and phosphatase [P] concentrations, where the substrate is taken to be a kinase [Fig. 2.2A]. These numbers were derived from the PaxDb

protein abundance database [71], taking advantage of UnitProt gene ontology associations to focus on just kinases and phosphatases in signal transduction pathways [72]. Each PaxDb data entry is in terms of ppm (parts per million) of abundance, relative to the total number of proteins in the cell. To convert from ppm to molar concentrations, we looked at data from human cells (which had the best representation in the database), and used the estimated total concentration of 2.7×10^6 proteins per μm^3 for human cells [91]. The latter concentration corresponds to 4.48×10^{-3} M. If y is the abundance in ppm units, then $4.48(y/10^6) \times 10^{-3}$ M is the corresponding molar concentration. Note that total concentrations are very similar across many different types of species [91], so there should not be a strong species-dependence in the analysis. For example the same analysis in mouse cells rather than human ones yields quantitatively similar results: a mean kinase/phosphatase concentration $10^{-8.31}$ M (versus $10^{-7.93}$ M in human cells), and a log-normal standard deviation of 1.03 (versus 0.84 in human cells).

- Reaction parameters [Fig. 2.2B-D]. These values were taken from the Sabio-RK database [73], where they were most often available in the following forms: for the kinase/phosphatase, Michaelis constants $K_M^{\text{kin}} = (\kappa_r + \kappa_u)/\kappa_b$, $K_M^{\text{pho}} = (\rho_r + \rho_u)/\rho_b$ (Fig. 2.2B), the corresponding specificity ratios $\kappa_r/K_M^{\text{kin}}$, ρ_r/K_M^{pho} (Fig. 2.2C), and the reaction rates κ_r and ρ_r (Fig. 2.2D). The resulting distributions were entirely consistent (though slightly narrower) with the distributions for the same parameter types analyzed in Ref. [69], which considered all enzymes (not just kinases and phosphatases).

Note that the six reaction parameter types that were collected from the Sabio-RK database (K_M^{kin}, K_M^{pho}, $\kappa_r/K_M^{\text{kin}}$, ρ_r/K_M^{pho}, κ_r, ρ_r) are not directly in the form that we need to calculate push-pull loop signaling properties. For the latter we would like to know (κ_b, ρ_b, κ_u, ρ_u, κ_r, ρ_r), or equivalently (κ_b, ρ_b, K_D^{kin}, K_D^{pho}, κ_r, ρ_r). Here the dissociation constants are defined as $K_D^{\text{kin}} = \kappa_u/\kappa_b$ and $K_D^{\text{pho}} = \rho_u/\rho_b$. Let us denote the parameter vector (κ_b, ρ_b, K_D^{kin}, K_D^{pho}, κ_r, ρ_r) as $\mathbf{v}$, with components v_α, $\alpha = 1,\ldots,6$. We would like to find a joint distribution for $\mathbf{v}$ that is self-consistent with the individual log-normal distributions for the alternative parameter types fitted directly from the database values (first 4 rows of Table 2.1). We will assume the simplest form for the joint distribution Φ: a product of individual log-normal distributions for each parameter v_α, with median values $\tilde{v}_\alpha$ and standard deviations σ_α:

$$\Phi(\mathbf{v}) = \prod_{\alpha=1}^{6} \frac{1}{v_\alpha \ln(10)\sqrt{2\pi\sigma_\alpha^2}} \exp\left(-\frac{(\log_{10} v_\alpha - \log_{10}\tilde{v}_\alpha)^2}{2\sigma_\alpha^2}\right). \tag{2.34}$$

Note that the $v_\alpha \ln(10)$ term in the denominator of the prefactor comes from the Jacobian due to the variable change between $\log_{10} v_\alpha$ and v_α. This ensures that the probability is properly normalized: $\int_0^\infty \prod_\alpha dv_\alpha \Phi(\mathbf{v}) = 1$. As explained above, kinases and phosphatase parameters are assumed to be drawn from the same distributions, so we enforce that $\tilde{v}_1 = \tilde{v}_2$, $\tilde{v}_3 = \tilde{v}_4$, $\tilde{v}_5 = \tilde{v}_6$, and analogously for the standard deviations σ_α. This leaves six distinct values that determine the distribution: $\tilde{v}_1$, $\tilde{v}_3$, $\tilde{v}_5$, σ_1, σ_3, σ_5.

To estimate these six distribution parameters, we use the following iterative numerical fitting procedure. We start with a guess for ($\tilde{v}_1$, $\tilde{v}_3$, $\tilde{v}_5$, σ_1, σ_3, σ_5) and then draw 10^4 parameter sets $\mathbf{v}$ from the resulting distribution $\Phi(\mathbf{v})$. For each parameter set we can calculate the alternative parameter types (K_M^{kin}, K_M^{pho}, $\kappa_r/K_M^{\text{kin}}$, ρ_r/K_M^{pho}, κ_r, ρ_r). We then

fit the resulting 10^4 values for these alternative types to individual log-normal distributions, and compare the means and standard deviations to the empirical results in the top half of Table 2.1. The sum of the relative absolute errors between the new joint fit values and the empirical results for the means / standard deviations is our overall goodness-of-fit measure. We perturb our guess for ($\tilde{\nu}_1$, $\tilde{\nu}_3$, $\tilde{\nu}_5$, σ_1, σ_3, σ_5) and accept the perturbation if it improves the goodness-of-fit. This procedure is iterated until convergence. The results of this joint fit are shown in the bottom half of Table 2.1. The joint fit predictions for the binding rate (κ_b, ρ_b) and dissociation constant ($K_D^{\rm kin}$, $K_D^{\rm pho}$) distributions are consistent with earlier estimates of these parameters in specific kinase/phosphatase systems [92]. As another consistency check, the joint fit distribution for the reaction rates (κ_r, ρ_r) is nearly identical to the individual empirical fit based on the Sabio-RK database values.

Finally we note that the simple joint distribution $\Phi(\mathbf{v})$ in Eq. (2.34) is by construction too broad: it may produce the correct marginal distributions for quantities collected from the Sabio-RK database, but it ignores any correlations between those individual parameters that may be present in natural systems. Estimating these correlations from the existing database entries is quite challenging, because relatively few entries have a complete list of all the parameters of interest. Hence, as explained in the main chapter text, we take $\Phi(\mathbf{v})$ to be effectively a superset: it should contain the true, presumably narrower, biological distribution plus parameter sets that are less likely to be observed in nature. A convenient aspect of this interpretation is that any collective conclusion we draw from the entire distribution $\Phi(\mathbf{v})$ should also be true for the subset of biological parameters. Moreover we can thus explore a larger design space (potentially available for evolution) than what we currently observe in modern biological systems.

2.5.5 Results for alternative input kinase concentrations

The results in Fig. 2.3D-F were for a mean input kinase concentration $[K] = 5$ nM. In Fig. 2.8 we show the analogous results for two different choices: $[K] = 0.5$ nM (left column) and $[K] = 50$ nM (right column). The main conclusions remain unchanged: the physiological $\Delta\mu$ range (highlighted in pink) is always just above the upper edge of the γ_x^{high} cloud, and the number of available parameter sets decreases rapidly as the mutual information I is increased.

2.5.6 Analysis of the Pbs2-Hog1 push-pull loop in yeast

To illustrate our theoretical framework in a concrete biological example, let us consider a kinase-phosphatase loop from one of the most extensively studied signaling pathways: the Hog1 mitogen-activated protein kinase (MAPK) pathway that allows yeast to adapt to extracellular osmotic changes [74, 75, 84]. We will focus in particular on the final portion of the pathway, where the active (phosphorylated) kinase Pbs2pp catalyzes the conversion of inactive Hog1 into phosphorylated Hog1pp. The latter protein is interchanged quickly between cytoplasm and nucleus, where it regulates a variety of responses to osmotic stress. Hog1pp is dephosphorylated by a combination of phosphatases Ptp2 (mainly in the nucleus) and Ptp3 in the cytoplasm [93]. Thus Pbs2pp will play the role of K in our model, Hog1 will be S, Hog1pp will be S^*, and Ptp2/Ptp3 will be P. To parameterize our model, we start with a more detailed theoretical description of the entire pathway developed by Zi *et al.* [74]. A key appeal of this work is that its parameters were carefully fit to extensive experimental data from yeast cells exposed to different time

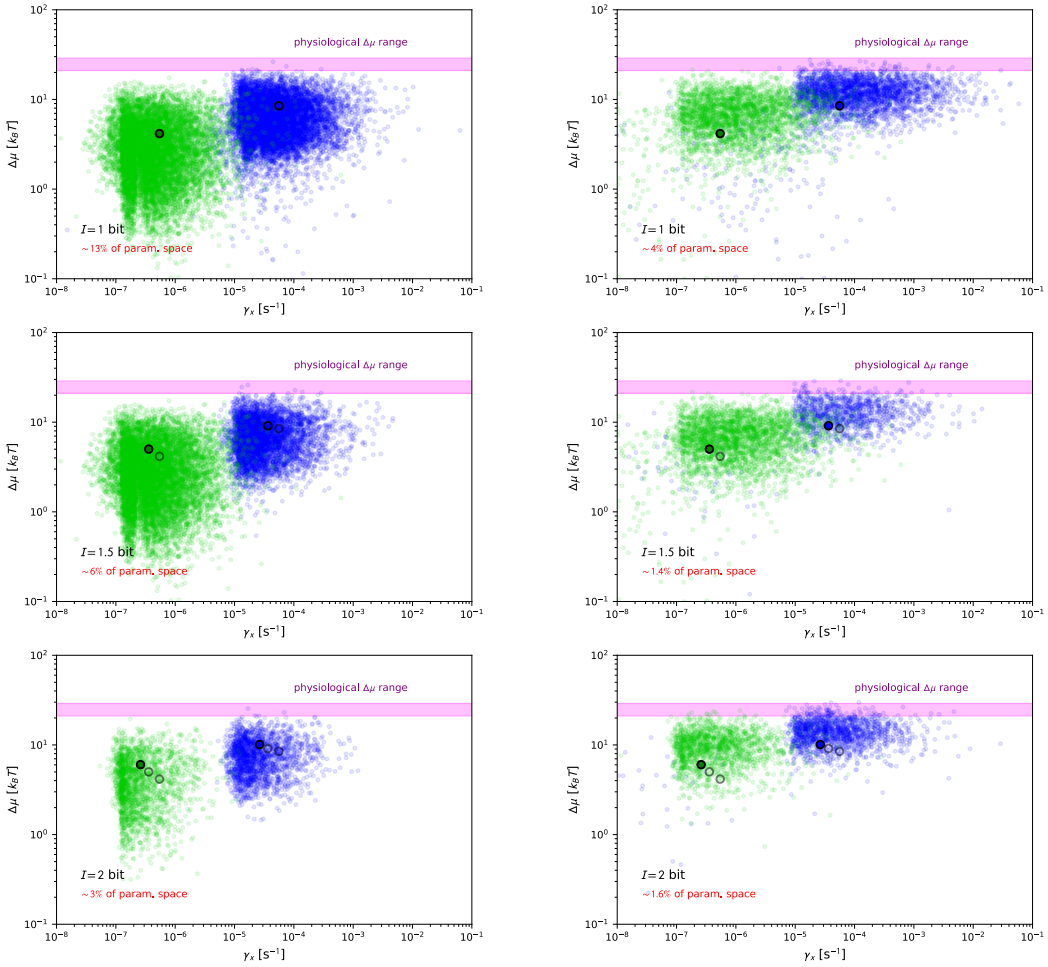

Figure 2.8. Analogous to Fig. 2.3D-F, except for input kinase concentration [K] = 5 nM (left column) and 50 nM (right column). The rows correspond to mutual information I = 1, 1.5, and 2 bits respectively. The probabilities of successfully drawing such a parameter set that achieves the specified I value are shown in red in panel.

series of external salt shocks in microfluidic experiments [84]. However since the parameters of Zi *et al.* are not expressed in the same form as the enzymatic reaction rates of our model, we do have to convert from their framework to ours, as described below.

Parameter estimation based on earlier literature. Ref. [74] explicitly distinguishes between the concentration of Hog1 and Hog1pp in the cytoplasm and nucleus, denoted with c and n superscripts respectively: [Hog1^c], [Hog1^n], [Hog1pp^c], [Hog1pp^n]. If we

are interested in the average concentrations overall, we can denote these as:

$$[S] \equiv \frac{[\text{Hog1}^{\text{c}}]V_c + [\text{Hog1}^{\text{n}}]V_n}{V_c + V_n}, \quad [S^*] \equiv \frac{[\text{Hog1pp}^{\text{c}}]V_c + [\text{Hog1pp}^{\text{n}}]V_n}{V_c + V_n}, \tag{2.35}$$

where V_c and V_n are the volumes of the cytoplasm and nucleus respectively, taken to have a ratio of $V_n/V_c = 0.14$ [74]. Eq. (2.35) also implies:

$$\frac{d[S]}{dt} = \frac{d[\text{Hog1}^{\text{c}}]}{dt} f + \frac{d[\text{Hog1}^{\text{n}}]}{dt}(1-f), \quad \frac{d[S^*]}{dt} = \frac{d[\text{Hog1pp}^{\text{c}}]}{dt} f + \frac{d[\text{Hog1pp}^{\text{n}}]}{dt}(1-f), \tag{2.36}$$

where $f = V_c/(V_c + V_n) = 0.88$. As a simplification of Eq. (2.35), we note in Ref. [74] import and export of the Hog1 proteins is fast relative to other reactions, and for a given input level the system rapidly reaches a stationary state with $[\text{Hog1}^{\text{n}}] \approx [\text{Hog1}^{\text{c}}] \approx [S]$, $[\text{Hog1pp}^{\text{n}}] \approx [\text{Hog1pp}^{\text{c}}] \approx [S^*]$.

We can now look at individual reactions that contribute to the time derivatives on the right-hand sides of Eq. (2.36) and find their analogues in our model. For example the phosphorylation step that converts Hog1^{c} to Hog1pp^{c} is expressed in Ref. [74] as an effective second order reaction of the form $K_{\text{pho}}^{\text{Hog1}}[\text{Pbs2pp}][\text{Hog1}^{\text{c}}]$, with rate constant $K_{\text{pho}}^{\text{Hog1}} = 11.2\ \mu\text{M}^{-1}\cdot\text{min}^{-1}$. This contributes positively to $d[\text{Hog1pp}^{\text{c}}]/dt$ and with a minus sign to $d[\text{Hog1}^{\text{c}}]/dt$, and so leads to contributions magnitude $f K_{\text{pho}}^{\text{Hog1}}[\text{Pbs2pp}][\text{Hog1}^{\text{c}}]$ to the right-hand sides of Eq. (2.36). Note that even though activation of Hog1 is actually a double phosphorylation (of a threonine and tyrosine residue), the entire process in this case can be well approximated through a single rate constant.

In our model the conversion of S to S^* occurs through the intermediate state S_K. However if we want to compare to the phosphorylation step of Ref. [74] in order to match parameters, we can look at the deterministic contribution to the dynamics (ignoring fluctuations) in the Michaelis-Menten approximation for enzyme kinetics [3]. In

this picture the phosphorylation reaction contributes to $d[S]/dt$ and $d[S^*]/dt$ through a term of magnitude $\kappa_r[K][S]/(K_M^{\text{kin}} + [S]) \approx (\kappa_r/K_M^{\text{kin}})[K][S]$, where the last simplification is valid when $K_M^{\text{kin}} \gg [S]$. If we compare $(\kappa_r/K_M^{\text{kin}})[K][S]$ to $fK_{\text{pho}}^{\text{Hog1}}[\text{Pbs2pp}][\text{Hog1}^{\text{c}}]$, noting that $[K] = [\text{Pbs2pp}]$ and $[S] \approx [\text{Hog1}^{\text{c}}]$, we can make the following identification:

$$\frac{\kappa_r}{K_M^{\text{kin}}} \approx fK_{\text{pho}}^{\text{Hog1}} = 1.64 \times 10^5\,\text{M}^{-1}\text{s}^{-1}. \tag{2.37}$$

The dephosphorylation steps in Ref. [74] are modeled as two pseudo-first-order reactions: conversion of Hog1pp^{c} to Hog1^{c} with rate $K_{\text{depho}}^{\text{Hog1pp}^{\text{c}}}[\text{Hog1pp}^{\text{c}}]$, and the conversion of Hog1pp^{n} to Hog1^{n} with rate $K_{\text{depho}}^{\text{Hog1pp}^{\text{n}}}[\text{Hog1pp}^{\text{n}}]$. The pseudo-first-order rate constants are given by: $K_{\text{depho}}^{\text{Hog1pp}^{\text{c}}} = 0.0906\ \text{min}^{-1}$ and $K_{\text{depho}}^{\text{Hog1pp}^{\text{c}}} = 4.14\ \text{min}^{-1}$. These reactions will lead to contributions of magnitude

$$(fK_{\text{depho}}^{\text{Hog1pp}^{\text{c}}}[\text{Hog1pp}^{\text{c}}] + (1-f)K_{\text{depho}}^{\text{Hog1pp}^{\text{n}}}[\text{Hog1pp}^{\text{n}}])$$

to the right-hand sides of Eq. (2.36). In our model (using a similar Michaelis-Menten approximation to the one described above, with $K_M^{\text{pho}} \gg [P]$), the analogous expression for dephosphorylation is effectively a second-order reaction with rate $(\rho_r/K_M^{\text{pho}})[P][S^*]$. Comparison of the two expressions, using the approximation $[\text{Hog1pp}^{\text{n}}] \approx [\text{Hog1pp}^{\text{c}}] \approx [S^*]$, leads to the identification:

$$\frac{\rho_r}{K_M^{\text{pho}}} \approx [P]^{-1}\left(fK_{\text{depho}}^{\text{Hog1pp}^{\text{c}}} + (1-f)K_{\text{depho}}^{\text{Hog1pp}^{\text{n}}}\right) = 1.69 \times 10^5\,\text{M}^{-1}\text{s}^{-1}. \tag{2.38}$$

Here we set $[P] = 0.058\ \mu$M as an average measure of phosphatase concentrations, to facilitate the conversion from pseudo-first-order to second-order rate constants. The value of $[P]$ is based on estimates of the concentrations of the two phosphatases in yeast from Ref. [94]: 0.049 μM for Ptp3 in the cytoplasm, and 0.067 μM for Ptp2 in the nucleus, where we have used $V_c = f(V_c + V_n)$, $V_n = (1-f)(V_c + V_n)$ and $V_c + V_n \approx 30$ fL [5, 74] to

Parameter	Value	Data source
substrate concentration $[S]$	0.38 μM	Hog1 abundance from Ref. [94]
phosphatase concentration $[P]$	0.058 μM	Ptp2/Ptp3 abundance from Ref. [94]
kinase specificity ρ_r/K_M^{kin}	1.64×10^5 M^{-1}s^{-1}	analysis of Ref. [74] model fit to experiments of Ref. [84]
phosphatase specificity ρ_r/K_M^{pho}	1.69×10^5 M^{-1}s^{-1}	analysis of Ref. [74] model fit to experiments of Ref. [84]

Table 2.2. Summary of parameters for the yeast Pbs2/Hog1 system estimated from earlier literature.

convert from populations to concentrations. Since the concentrations were of similar scale, we let $[P]$ be the mean of the two values.

As a consistency check to make sure the final estimates of the specificity ratios $\kappa_r/K_M^{\text{kin}}$ and ρ_r/K_M^{pho} in Eqs. (2.37)-(2.38) are biologically plausible, we can compare them with the distribution of these ratios among kinases/phosphatases from the Sabio-RK database in Fig. 2.2C. The values for the Hog1/Pbs2 system are not unusual, and lie near the higher end of the range, at about the 0.87 quantile. The final parameter value we can estimate from the literature is the mean Hog1 concentration $[S] = 0.38$ μM, based on the abundance reported in Ref. [94].

Estimation of remaining parameters. Based on the above analysis, we have estimates for four quantities in the Pbs2/Hog1 system drawn from the earlier literature: $\kappa_r/K_M^{\text{kin}}$, ρ_r/K_M^{pho}, $[S]$, $[P]$. These are summarized in Table 2.2. The relationship of the enzymatic reaction/binding/unbinding rate parameters to the estimated values then takes

the form:

$$\begin{aligned}
\kappa_r &= \left(\frac{\kappa_r}{K_M^{\text{kin}}}\right) K_M^{\text{kin}} = (1.64\times 10^5 \text{M}^{-1}\text{s}^{-1}) K_M^{\text{kin}}, \\
\kappa_b &= \left(\frac{\kappa_r}{K_M^{\text{kin}}}\right) \frac{K_M^{\text{kin}}}{K_M^{\text{kin}} - K_D^{\text{kin}}} = (1.64\times 10^5 \text{M}^{-1}\text{s}^{-1}) \frac{K_M^{\text{kin}}}{K_M^{\text{kin}} - K_D^{\text{kin}}}, \\
\kappa_u &= \left(\frac{\kappa_r}{K_M^{\text{kin}}}\right) \frac{K_M^{\text{kin}} K_D^{\text{kin}}}{K_M^{\text{kin}} - K_D^{\text{kin}}} = (1.64\times 10^5 \text{M}^{-1}\text{s}^{-1}) \frac{K_M^{\text{kin}} K_D^{\text{kin}}}{K_M^{\text{kin}} - K_D^{\text{kin}}}, \\
\rho_r &= \left(\frac{\rho_r}{K_M^{\text{pho}}}\right) K_M^{\text{pho}} = (1.69\times 10^5 \text{M}^{-1}\text{s}^{-1}) K_M^{\text{pho}}, \\
\rho_b &= \left(\frac{\rho_r}{K_M^{\text{rho}}}\right) \frac{K_M^{\text{rho}}}{K_M^{\text{rho}} - K_D^{\text{rho}}} = (1.69\times 10^5 \text{M}^{-1}\text{s}^{-1}) \frac{K_M^{\text{rho}}}{K_M^{\text{rho}} - K_D^{\text{rho}}}, \\
\rho_u &= \left(\frac{\rho_r}{K_M^{\text{rho}}}\right) \frac{K_M^{\text{rho}} K_D^{\text{rho}}}{K_M^{\text{rho}} - K_D^{\text{rho}}} = (1.69\times 10^5 \text{M}^{-1}\text{s}^{-1}) \frac{K_M^{\text{rho}} K_D^{\text{rho}}}{K_M^{\text{rho}} - K_D^{\text{rho}}},
\end{aligned} \tag{2.39}$$

The above parameters depend on the values of K_M^{kin}, K_M^{pho}, K_D^{kin}, K_D^{kin}. While we do not know what these are for the Pbs2/Hog1 system, we can draw their values from the corresponding empirical log-normal distributions described in Table 2.1. By repeating the draw many times, we can check how our final optimality analysis (see below) depends on the precise values of the unknown parameters. As it turns out the dependence of R_0, γ_x^{high} and R_0^{min} on the unknown values is quite weak, and we will be able to make robust estimates for these quantities. In the cases of K_M^{kin} and K_M^{pho}, we constrain the random draw from their log-normal distributions to enforce $K_M^{\text{kin}} \geq 100[S]$ and $K_M^{\text{pho}} \geq 100[P]$. This ensures self-consistency with the assumptions $K_M^{\text{kin}} \gg [S]$ and $K_M^{\text{pho}} \gg [P]$, which were used in the previous subsection to match the form of the phosphorylation / dephosphorylation reactions between Ref. [74] and our model. The final two parameters are the reverse reaction rates κ_{-r} and ρ_{-r}. Since we do not have any experimental estimates of these for the Pbs2/Hog1 system, we assume that the physiological value of $\Delta\mu$

in yeast (around 21 k_BT [5]) is sufficiently high that κ_{-r} and ρ_{-r} are negligible under normal conditions.

Bandwidth and gain. Given the parameter estimation procedure described above, we can calculate γ_x^{high}, R_0, R_0^{min} for each draw of the unknown parameters. The results remain within a narrow distribution, relatively insensitive to the values of the unknown parameters. The mean and standard deviations for 50 draws are: $\gamma_x^{\text{high}} = (1.22 \pm 0.04) \times 10^{-3}$ s^{-1}, $R_0 = 0.0621 \pm 0.0001$ s^{-1}, $R_0^{\text{min}} = 0.059 \pm 0.002$ s^{-1}.

2.5.7 Estimation of total resting metabolic expenditure

For single-celled organisms, the total resting metabolic expenditure C_T can be estimated by the approach outlined in Ref. [64]. C_T has two contributions: $C_T = C_G + t_r C_M$. Here C_G is the expenditure involved in growth during one generation time t_r, and C_M is the maintenance cost per unit time. Using a large collection of metabolic data from Ref. [63], covering both prokaryotes and single-celled eukaryotes, one can observe that both C_M and C_T scale approximately linearly with cell volume V, agreeing with the prediction of the bioenergetic growth model of Ref. [64]. The expression for C_T based on the results of these linear fits is [64]:

$$C_T = (2.3 \times 10^{10}\,\text{P/fL})\,V + (9.2 \times 10^{4}\,\text{P/(s}\cdot\text{fL)})\,t_r V. \tag{2.40}$$

where the unit P corresponds to the hydrolysis of a phosphate bond (i.e. the consumption of one ATP or ATP equivalent). Using the main chapter text values of $V = 30$ fL and $t_r = 3600$ s, we get $C_T = 7.0 \times 10^{11}$ P.

3 Machine learning methods for exploring single-molecule heterogeneity

3.1 Introduction

Recent decades have seen huge triumphs for single-molecule experimental techniques in the life sciences. Methods such as Förster resonance energy transfer (FRET), atomic force microscopy (AFM), optical tweezers, and single particle tracking (SPT) offer the opportunity to infer structures, conformations, and also dynamics of single bio-molecules. One of the most interesting discoveries from single-molecule experiments is the existence of functional heterogeneity: multiple, distinct (and sometimes long-lived) structural conformations of molecules can have significantly different functional properties (for example catalytic rates changing by several orders of magnitude). This is true despite the fact all conformations correspond to covalently identical bio-molecules, coded by the same genetic sequence. The biological consequences of this novel form of epigenetic variation are just beginning to be explored. Many questions remain regarding how the conformational changes are regulated by cell signaling networks or other cellular micro-environments, and how the changes couple with biological function [11]. The problem is made more complex by the fact that functional heterogeneity exists in

different incarnations in many classes of bio-molecules: protein enzymes [8–10], ribozymes [11], DNA [12], motor proteins[13], and adhesion complexes [14].

Along with in-depth studies focusing on the biological roles of heterogeneity in specific systems, one needs general methods to identify heterogeneity in single-molecule experimental data. The fact that such heterogeneity can be overlooked was demonstrated in Ref. [15], which focused on analyzing ten previously published data sets from AFM pulling experiments—one of the most well-established single-molecule techniques, with an extensive research literature. Heterogeneity was discovered in half of the data sets, most of which were not flagged as heterogeneous in the original studies. The method introduced in Ref. [15] has two nice features: (i) it extracts a single non-dimensional parameter $\Delta \geq 0$ from the pulling data (histograms of the rupture times/forces). Values of $\Delta \ll 1$ indicate that all the experimental trajectories come from a single conformational state (or a group of rapidly interconverting states that effectively act as a single state). For $\Delta \gtrsim 1$, the system must have more than one long-lived conformational state. (ii) The method allows one to put upper bounds on the interconversion rates in heterogeneous systems. Among those data sets that were identified as such, the rates were all less than $10\ s^{-1}$, slow enough that each pulling trajectory would involve only a single state.

However a key drawback of the Ref. [15] approach is that the information it provides about the heterogeneous states is fairly limited: it tells us nothing about the number of states, their relative proportions, or the parameters that characterize each state. The goal of the study presented here is to fill in those details, using two different machine learning techniques to analyze single-molecule AFM pulling data: a supervised

deep learning algorithm and an unsupervised non-parametric Bayesian approach. Using large sets of synthetic data covering a wide range of possible experimental conditions and bio-molecular parameters, we demonstrate the effectiveness of both methods at characterizing heterogeneity, and investigate their relative strengths and weaknesses. The result is a robust, automated system that is ready to be deployed for the analysis of empirical data.

Deep learning algorithms now play important roles in many fields [16] from facial recognition [19] to drug discovery [95, 96]. Applications of deep learning algorithms in the field of biophysics recently exploded, owing to their unprecedented ability to extract patterns in noisy biological data. Non-parametric Bayesian inference, which dates back to the 1970s [26, 27], is also gaining a foothold in biophysics [97, 98]. This is particularly true for analysis of time series data [99–105]. However to our knowledge the current work is the first application of either deep learning or non-parametric Bayesian ideas to the problem of heterogeneous states in single-molecule AFM pulling data.

3.2 Modeling the rupture time distribution in an AFM pulling experiment

Before discussing the algorithms, let us give a brief overview of AFM pulling experiments and the biophysical models used to describe them. It will be simplest to start with the case of a bio-molecule with a single state, and then generalize to a heterogeneous system.

3.2.1 Rupture time distribution for a single state system

Consider a generic free-energy landscape for a bio-molecular system with a single functional state S, corresponding to the deep well in Fig. 3.1A. Note that a “state” in more

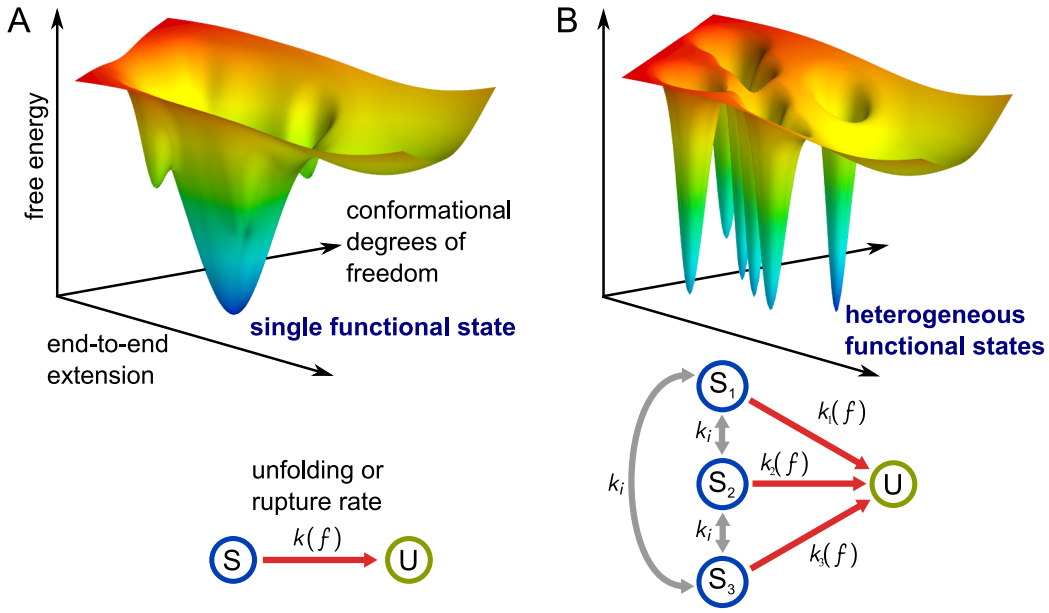

Figure 3.1. (A) Schematic bio-molecular free energy landscape with a single functional state, S. Under an adiabatically increasing external force f, there is a instantaneous rupture rate $k(f)$ describing transitions between S and the unfolded/unbound ensemble U. (B) Schematic free-energy landscape of a heterogeneous system with multiple functional states. Each functional ensemble S_i will have a state-dependent adiabatic rupture rate $k_i(f)$. A single overall scale rate k_i is introduced to described interconversion between the various states.

technical terms always refers to a ensemble of structural conformations, since thermal fluctuations are always present and will make the molecule explore the local vicinity of the well minimum. However the functional properties (like the associated catalytic rate if the system is an enzyme, or the adhesion lifetime if it is a complex) do not change significantly in the presence of thermal fluctuations. Hence it makes sense to collectively refer to the ensemble as a state. There may of course be other (typically shallower) minima in the landscape, for example corresponding to non-functional misfolded or unfolded states. As shown in Fig. 3.2A, the AFM experiment involves connecting the bio-molecular system to the cantilever and platform through protein or nucleic acid

linkers of known stiffness. The cantilever is pulled at a constant velocity v, applying a force ramp with slope $df/dt = \omega_s(f)v$, where $\omega_s(f)$ is the effective stiffness of the setup (linkers plus the AFM cantilever). As described in Chapter 1, we define a characteristic stiffness $\bar{\omega}_s \equiv$ the mean $\omega_s(f)$ over the range of forces probed in the experiment (note that the precise value of $\bar{\omega}_s$ is not required in this work). This allows us to introduce a constant characteristic force loading rate r proportional to the velocity, $r = \bar{\omega}_s v$.

If the initial state of the system is state S at time $t = 0$, the force ramp tilts the landscape along the end-to-end extension coordinate. If we model the conformational dynamics of the system as diffusion within this landscape, the tilting eventually leads to a transition out of S into a state U, associated with unbinding of the complex or unfolding of the molecule. The rupture (or unfolding) rate $k(f)$ at a constant external force f is assumed to take the form of the Bell model [30]:

$$k(f) = k_0 e^{\beta f D}, \tag{3.1}$$

where k_0 is the escape (rupture) rate at zero force, D is the transition state distance (quantifying how sensitive state S is to the destabilization effects of external force), $\beta = 1/k_B T$, k_B is Boltzmann constant, T is temperature. More complicated rupture models can easily be substituted [106], but the Bell model provides an excellent approximation for the rupture dynamics of a wide variety of systems. We assume the force ramp is slow enough that we are in the so-called adiabatic regime, which is a typical assumption for AFM experiments [15, 106]. This entails that equilibration within the well is rapid, so that the system reaches quasi-equilibrium at the instantaneous value of the force $f(t)$ at all times t before rupture. As was shown in Ref. [15], it is possible to detect violations of this adiabatic assumption from the data, and experimental ramp values r are generally

within the adiabatic regime. Thus the instantaneous rupture rate at time t can be taken as $k(f(t))$, or equivalently we can change variables from $f(t)$ to t to get $k(t) = k_0 e^{\beta r t D}$.

Let $\Sigma_r(t)$ be the survival function of state S for ramping rate r: the probability that rupture has not occur before time t. In the adiabatic regime, the survival function satisfies the kinetic equation,

$$\frac{d\Sigma_r(t)}{dt} = -k(t)\Sigma_r(t), \tag{3.2}$$

with boundary condition $\Sigma_r(0) = 1$. This equation can be easily solved to yield $\Sigma_r(t) = \exp\left(k_0(1 - e^{\beta D r t})/(\beta D r)\right)$, which then allows us to write down our main quantity of interest, the distribution of rupture times $F_r(t|\phi)$:

$$F_r(t|\phi) = -\frac{d}{dt}\Sigma_r(t) = k_0 e^{\frac{k_0 - e^{\beta r t D} k_0 + (\beta r D)^2 t}{\beta r D}}. \tag{3.3}$$

The parameters which determine the rupture time distribution are the Bell model parameters $\phi = (k_0, D)$.

3.2.2 Rupture time distribution for a heterogeneous system

Multiple functional states lead to a free energy landscape with many deep wells (Fig. 3.1B), each corresponding to a distinct state S_i. Transitions can occur between these states, which we describe with some overall interconversion rate k_i. As discussed at the beginning of this chapter, the analysis method of Ref. [15] can put an upper bound on the scale of k_i, and our focus will be on experiments where k_i is much slower than the typical range of rupture rates $k(f(t))$. In this scenario the heterogeneous states are long-lived to the degree that each experimental run involves a single state from beginning to end. In the opposite regime of $k_\mathrm{i} \gg k(f(t))$, the energy barriers between different functional states are so small that they rapidly equilibrate among themselves before rupture, effectively acting like a single state system.

For the slow interconversion case, each rupture time recorded by AFM corresponds to one of the transitions S_i to U represented by red arrows in Fig. 3.1. Let p_i be the probability that the system started in state S_i. The associated rupture rate $k_i(f)$ will generally depend on i, since the Bell model parameters $\phi_i = (k_{0i}, D_i)$ could be different for each conformational state. Thus the rupture times t that we observe in the experiment have a mixture distribution

$$F_r(t|\boldsymbol{p},\boldsymbol{\phi}) = \sum_i p_i F_r(t|\phi_i), \tag{3.4}$$

where $\boldsymbol{\phi} = (\phi_1, \phi_2, \ldots)$ is the vector of parameters for the different states. Note that Eq. (3.4) reduces to single-state case Eq. (3.3) if some component of state probability vector $p_i = 1$.

3.3 Overview of the machine learning workflow

The main problem can now be stated concisely: given a set of N rupture times $\boldsymbol{t} = (t_1, ..., t_N)$ collected from independent AFM pulling experiments, can we infer the state probabilities $\boldsymbol{p}$ and parameters $\boldsymbol{\phi}$ that characterize the mixture distribution in Eq. (3.4) from which the times were sampled? We designed two machine learning algorithms to answer this question in part or whole, summarized in the workflow diagram of Fig. 3.2. Both algorithms operate on the same data set $\boldsymbol{t}$ from AFM pulling experiments (panel A), a generic example of which is illustrated in panel B. The analysis then proceeds through either the non-parametric Bayesian pipeline (panels B-D-F) or the deep learning pipeline (panels B-C-E-G).

For the deep learning case, since the ordering of the times is arbitrary, we preprocess the data by first sorting it by time (panel C). This is fed into our trained neural network

(panel E), which ultimately yields an estimate of the state probability vector $\boldsymbol{p}$. As explained below, the network can only handle up to some specified maximum number of states (for example $M_{\max} = 4$), so the vector $\boldsymbol{p}$ has dimensions $M_{\max}$. If the system is estimated to have fewer than $M_{\max}$ states, components of the vector will be close to zero, but the network cannot fully infer the details of systems with more than $M_{\max}$ states.

For the non-parametric Bayesian pipeline, an MCMC method is used to sample from the posterior distribution over different mixtures and state parameters (panel D), as summarized in Chapter 1. The end result is a Markov chain of $(\boldsymbol{c}, \boldsymbol{\phi})$ pairs, where $\boldsymbol{c}$ is an N-dimensional vector whose component c_i is the state label of the ith data point. As explained below, we can use this Markov chain to estimate the most likely state probabilities $\boldsymbol{p}$ and parameters $\boldsymbol{\phi}$ (panel F). Notably the number of states does not have an arbitrary upper cutoff, though the results are influenced by the concentration parameter α that is used to define the Dirichlet process prior for the posterior distribution.

In the following sections, we delve into the details of both algorithms, evaluate performance metrics for each one, and compare the results.

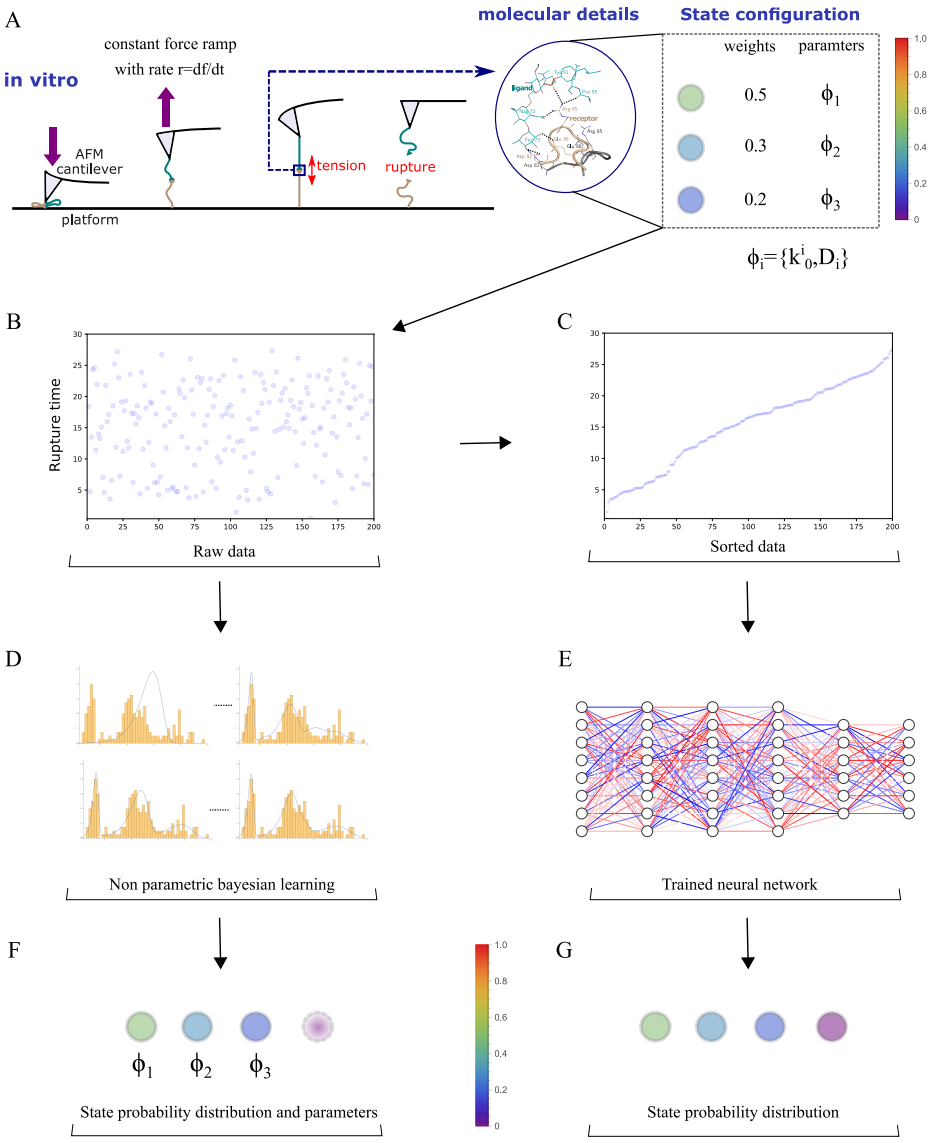

Figure 3.2. **Overview of workflow** : **A**: N repetitions of an AFM pulling experiments for a bio-molecular system. In this hypothetical example the system has three different functional states with probabilities $\boldsymbol{p} = (0.5, 0.3, 0.2)$ and corresponding Bell parameters $\boldsymbol{\phi} = (\phi_1, \phi_2, \phi_3)$. **B**: Raw rupture time data $\boldsymbol{t}$ collected from the experiments, with $N = 200$. Rupture times are found from $t = f/r$, where f is the rupture force and r is the ramping rate. **C**: Sorted rupture times as input to **E**: a trained neural network that outputs **G**: the state probabilities $\boldsymbol{p}$, where spheres represent each functional states and sphere color reflects the probability weight. The dimensionality of $\boldsymbol{p}$ reflects the cutoff $M_{\max} = 4$ for maximum number of states hardwired into the neural network. Alternatively, feeding the raw data to **D**: a non-parametric Bayesian algorithm can eventually return **F**: both $\boldsymbol{p}$ and the corresponding state parameters $\boldsymbol{\phi}$.

3.4 Data set generation

Both machine learning approaches require large data sets: for training in the deep learning approach, and for testing the algorithms in both cases. The data sets consist of $(\boldsymbol{t}, \boldsymbol{p}, \boldsymbol{\phi})$ covering many different possible mixture and parameter values, designed to mimic biological ranges. Each set of Bell model parameters $\phi = (k_0, D)$ is drawn from a prior distribution Φ defined as follows: $k_0 = 10^x$ s^{-1} and $D = 10^y$ nm, where x is a uniform random real number from the range $[-4, -2]$ and y is a uniform random real number in the range $[-2.0, -0.5]$. These cover typical Bell model parameter ranges seen in the literature [15]. The ramp rate r is not a hidden variable, since experimental data is collected for a known ramp protocol. We set $r = 100$ pN/s as a typical experimental ramp scale, though as it turns out the results scale with r in a simple way: any change in r can be compensated for by renormalization of the parameter D in Eq. (3.3) to yield the same rupture time distribution. Networks trained at one value of r should be able to successfully analyze data collected at other r, and we have verified this in practice. For N, the number of experimental runs, we will investigate a range $N = 4 - 200$. The upper end of that range covers the numbers found in typical AFM experiments [15], and the lower end is explored to see the effect of limited rupture time data sets. When choosing a number of states M for each instance of $(\boldsymbol{t}, \boldsymbol{p}, \boldsymbol{\phi})$, we let M be an integer in the range 1 to $M_{\max}$. We will choose equal numbers of examples for each value of M in that range. In the results below we set $M_{\max} = 6$, a guess at the typical extreme of what might be realistically observed in an experimental system. But we have verified that the training and analysis works at even larger $M_{\max}$. This cutoff $M_{\max}$ will also appear as a hyperparameter describing the output layer dimension in our neural network. The algorithm for generating a set of data for a certain choice of M is summarized below:

Algorithm 1: Data generation for systems with M states

Loop

1. Draw a state probability vector $\boldsymbol{p} = \{p_1, p_2, ..., p_M\}$ from the uniform distribution on the $(M-1)$-dimensional probability simplex;
2. Draw Bell parameters $\phi_i = \{k_{0i}, D_i\}$ for each $i = 1, ..., M$ from the distribution Φ;
3. **for** n=1, ..., N **do**
   Draw a rupture time t_n from the mixture distribution $\sum_{i=1}^{M} p_i F_r(t|\phi_i)$;

end

End Loop

3.5 Deep learning algorithm

It is natural to consider a supervised learning algorithm based on an artificial neural network as a potential solution for our task. If we focus on the state probability distribution $\boldsymbol{p}$ as the target, we can use a fully connected neural network with a softmax final layer of size $M_{\max}$ that outputs a probability $\boldsymbol{q} = \{q_1, q_2, ..., q_{M_{\max}}\}$. Given the experimental data $\boldsymbol{t}$ as input, the goal of the network will be to output $\boldsymbol{q}$ as close as possible to the underlying $\boldsymbol{p}$ that describes the mixture.

3.5.1 Training set format

The training data set is generated by applying Algorithm 1 and takes the form of input ($\boldsymbol{t}$) and target output ($\boldsymbol{p}$) pairs. Since the rupture time observations collected in the vector $\boldsymbol{t} = (t_1, t_2, ..., t_N)$ are independent of one another, the ordering of the sequence

is arbitrary. As a preprocessing step, we found that sorting the vector in ascending order by time, as shown in Fig. 3.2C, improved the effectiveness of the training. Similarly the ordering of the state labels for the components of $\boldsymbol{p}$ is arbitrary, so in the training data we always presented the network with $\boldsymbol{p}$ vectors whose components were sorted from largest to smallest probability. (In the discussion below we will always assume $\boldsymbol{p}$ is sorted in this manner.) Though not explicitly constrained to do so, the network will then learn to output $\boldsymbol{q}$ with components that are also sorted in descending order.

For any given network training, we always use the same N, defining the length of $\boldsymbol{t}$, and the same $M_{\max}$, defining the length of $\boldsymbol{p}$. As mentioned above, we trained a number of different networks with $N = 4-200$ to compare their performance, but kept $M_{\max} = 6$. For each training, the data set was constructed as follows: for every value of state number $M = 1,\ldots,6$ we created 2000 systems defined by distinct $(\boldsymbol{p},\boldsymbol{\phi})$, and for every such system we created 25 different sets of experimental observations $\boldsymbol{t}$. The end result is 3×10^5 examples of the form $(\boldsymbol{t},\boldsymbol{p},\boldsymbol{\phi})$.

3.5.2 Architecture

We found the best performance using the simple fully connected neural network architecture shown in Fig. 3.3. The input and output layer sizes, N and $M_{\max}$ respectively, are set by our learning task, and we choose the size of the three hidden layers to be equal to the input dimension N. The activation function of each hidden layer is rectified linear unit (ReLU), as described in Chapter 1. We explored more complex architectures up to 10 hidden layers, but found no significant improvement in the accuracy of the network, as described by the metrics defined below. The required length of training does increase with more hidden layers, so we kept the number at 3 to make the training as efficient as possible without sacrificing performance.

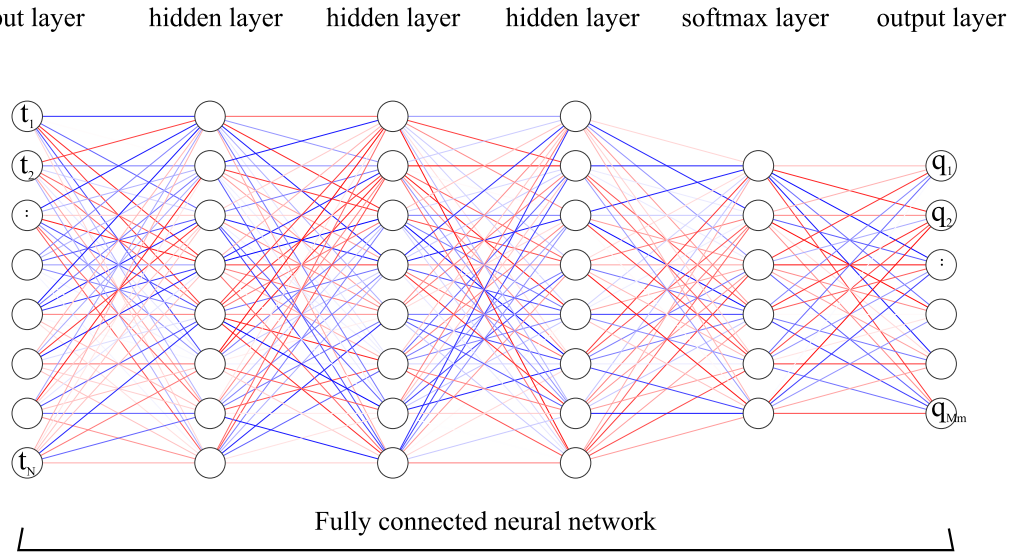

Figure 3.3. Schematic of the neural network architecture. The number of neurons in each hidden layer is the same as the input size N. The activation function for every hidden layer is ReLU. The state number cutoff $M_{\max}$ is set as the softmax layer size so that the output can take the form of a probability vector $\boldsymbol{q} = \{q_1, q_2, ..., q_{M_{\max}}\}$.

3.5.3 Loss function

In order to define the loss function that we use for our training, let us introduce three quantities that allow us to compare the network output $\boldsymbol{q}$ and the target $\boldsymbol{p}$ for a given experimental data set $\boldsymbol{t}$ as input.

The first is the fidelity $F(\boldsymbol{q}, \boldsymbol{p})$ [107], a widely used quantity in quantum information theory that also works well as a measure of similarity for classical probabilities. In the classical case it is defined as follows:

$$F(\boldsymbol{q}, \boldsymbol{p}) = \left(\sum_i \sqrt{q_i p_i} \right)^2 . \tag{3.5}$$

Notice that $F(\boldsymbol{q}, \boldsymbol{p}) = 1$ if and only if our algorithm outputs a perfect prediction, $\boldsymbol{q} = \boldsymbol{p}$. In general $0 \leqslant F(\boldsymbol{q}, \boldsymbol{p}) \leqslant 1$. However when both $\boldsymbol{q}$ and $\boldsymbol{p}$ are sorted probability vectors, as

in our case, the minimum possible fidelity is given by $1/M_{\max}$, and achieved for example when $\boldsymbol{q} = (1/M_{\max}, 1/M_{\max}, ..., 1/M_{\max})$ and $\boldsymbol{p} = (1, 0, 0, ..., 0)$.

A second way of analyzing probabilities is to ask how many states does a probability vector effectively correspond to? For example let us say the target was $\boldsymbol{p} = (1, 0, 0, ..., 0)$, a single-state ($M = 1$) system. If the network predicted $\boldsymbol{q} = (0.5, 0.5, 0, ..., 0)$ it would clearly be wrong, since it would have interpreted the data coming from an equal mixture of two states. But what about a prediction of $\boldsymbol{q} = (0.98, 0.01, 0.01, 0, ..., 0)$. Clearly this is better, but is there a way of saying such a $\boldsymbol{q}$ corresponds to (roughly) a one state prediction, rather than three (the number of non-zero components)? We thus introduce an effective dimension $D_{\text{eff}}(\boldsymbol{q})$ defined as follows:

$$D_{\text{eff}}(\boldsymbol{q}) = \frac{1}{\sum_i q_i^2}. \tag{3.6}$$

For $\boldsymbol{q} = (0.5, 0.5, 0, ..., 0)$ this would give $D_{\text{eff}}(\boldsymbol{q}) = 2$, and in general if the probability is equally distributed among M states, $D_{\text{eff}}(\boldsymbol{q}) = M$. For $\boldsymbol{q} = (0.98, 0.01, 0.01, 0, ..., 0)$ the effective dimension $D_{\text{eff}}(\boldsymbol{q}) = 1.04$, agreeing with our intuition that this is approximately a one-state prediction. From another perspective, D_{eff} provides a measure of the heterogeneity of a mixture, with larger D_{eff} corresponding to more heterogeneous cases. We can thus use the absolute difference between the predicted and actual effective dimension, $|D_{\text{eff}}(\boldsymbol{q}) - D_{\text{eff}}(\boldsymbol{p})|$, as another performance metric for the algorithm.

Finally, we can also employ the cross entropy $H(\boldsymbol{q}, \boldsymbol{p})$, traditionally used in machine learning applications for comparing probabilities:

$$H(\boldsymbol{p}, \boldsymbol{q}) = -\sum_i p_i \log q_i. \tag{3.7}$$

The smaller the value of $H(\boldsymbol{p}, \boldsymbol{q})$, the more similar the two probabilities. Eq. (3.7) can also be expressed as $H(\boldsymbol{p}, \boldsymbol{q}) = H(\boldsymbol{p}) + D_{\text{KL}}(\boldsymbol{p}||\boldsymbol{q})$, where $H(\boldsymbol{p})$ is the Shannon entropy

and $D_{\mathrm{KL}}(\boldsymbol{p}||\boldsymbol{q})$ is the Kullback-Leibler divergence. Since the target $H(\boldsymbol{p})$ is fixed by the training data set, varying the network weights to minimize $H(\boldsymbol{p},\boldsymbol{q})$ is equivalent minimizing the Kullback-Leibler divergence between $\boldsymbol{q}$ and $\boldsymbol{p}$.

In practice, we found that a loss function that linearly combines all three comparison measures discussed above works well for training:

$$L(\boldsymbol{q},\boldsymbol{p}) = -F(\boldsymbol{q},\boldsymbol{p}) + \left|D_{\mathrm{eff}}(\boldsymbol{q}) - D_{\mathrm{eff}}(\boldsymbol{p})\right| + H(\boldsymbol{p},\boldsymbol{q}). \tag{3.8}$$

Optimization of the loss function was implemented via stochastic gradient descent together with adaptive moment estimation (Adam) [108], a common choice in modern deep learning applications.

3.6 Non-parametric Bayesian learning

The second machine learning approach we deploy is non-parametric Bayesian learning, summarized in the discussion at the end of Chapter 1. The appeal of this method is that it does not require a cutoff M_{max} on the number of states, allowing us in principle to consider mixtures with arbitrary numbers of components. However though we get rid of M_{max}, we do have a different kind of hyperparameter in the form of α, which influences the nature of the Dirichlet process prior $P(\boldsymbol{c},\boldsymbol{\phi})$ used in the posterior distribution of Eq. (1.24). Increasing α has some similarities to increasing M_{max}, by allowing the search for parameter sets that describe the data to explore a larger space of heterogeneous mixtures.

As explained in Chapter 1, the practical implementation of the non-parametric Bayesian approach involves using an MCMC method to create a Markov chain $(\boldsymbol{c}^{(j)},\boldsymbol{\phi}^{(j)})$, $j = 1,\ldots,K$. After a certain burn-in period of K_b iterations (we use $K_b = 10^3$) where the

MCMC method converges to stationarity, the subsequent samples for $j > K_b$ represent draws from the posterior distribution of Eq. (1.24). The specific MCMC algorithm we use is based on Algorithm 8 of Ref. [29], and is designed to exactly satisfy the detailed balance condition in Eq. (1.27). Given that the Dirichlet process prior was used as part of the posterior, the way the MCMC method draws new samples has some similarity to the Chinese restaurant process described in Chapter 1. The full details of one MCMC iteration are described in Algorithm 2 below, but it is worth first summarizing it qualitatively using the restaurant metaphor. At each MCMC iteration, there is an existing seating arrangement ($\boldsymbol{c}$) and different food options for each table ($\boldsymbol{\phi}$). The iteration consists of updating the seating assignment c_i for each customer t_i sequentially, $i = 1, ..., N$. When being reseated, the customer can choose to sit at any of the already occupied tables, or at m new (empty) tables, where each of the new tables has food ϕ_c drawn from the distribution Φ (The hyperparameter m, which we set to $m = 2$, is discussed in more detail below). The concentration parameter α determines how likely the customer is to choose a new table, while the probability of going to an occupied table is proportional to how many people are already seated there. If the customer was dining alone previously, their current table is considered one of the "new" set for the purposes of the algorithm, but the food at that table is not changed. At the end of this whole reseating process we have an updated seating arrangement. This then becomes the starting point for the next iteration. However there is one more step not present in the original Chinese restaurant process: after reseating is completed, we do a Metropolis-Hastings update of the food at each table $\boldsymbol{\phi}_c \to \boldsymbol{\phi}'_c$, where c runs over the number of distinct classes (tables). The probabilities of the reseating process and the food update are both influenced by the customer's happiness with the food, expressed in terms $F_r(t_i|\phi_c)$, the likelihood that a

particular t_i would be observed for a Bell model parameter set ϕ_c. The net effect of the update is to make the customers happier with the food at their table (the data points in a certain class c more likely to be observed given parameters ϕ_c). In the original Chinese restaurant process, which constructed a prior without reference to the experimental data, this consideration of food preference was absent. Here it reflects the fact that the MCMC is designed to draw samples from the posterior, which involves not just the Dirichlet prior but also the likelihood function of Eq. (1.26).

Algorithm 2: MCMC update for class assignments $\boldsymbol{c}$ and parameters $\boldsymbol{\phi}$

1. *Class reassignment and addition of new classes*:
for i=1, ..., N **do**
  Create a set of m new (empty) classes; draw parameters from Φ for each one.
  if current class c_i has only one occupant **then**
    Empty the current class and move it to the new set, replacing one of the m classes there.
  end
  Choose a new value c for c_i with the following probabilities:

$$\begin{cases} b\frac{\mu_{i,c}}{N-1+\alpha}F_r(t_i|\phi_c) & \text{if } c \text{ is a class with occupants} \\ b\frac{\alpha/m}{N-1+\alpha}F_r(t_i|\phi_c) & \text{if } c \text{ is a new (empty) class} \end{cases}$$

  Here $\mu_{i,c}$ is the number of occupants in class c, excluding the current data point t_i. The prefactor b is for normalization, to ensure that the probabilities for all choices sum to 1.
end
2. Delete parameters for any empty classes from $\boldsymbol{\phi}$, and if necessary relabel occupied classes so that the sequence of class labels does not have any gaps.
3. *Parameter update via Metropolis-Hastings*:
for each class c **do**
  Create a candidate parameter set ϕ_c' by perturbing each value in ϕ_c randomly by ±1%.
  if $\Phi(\phi_c') > 0$ **then**
    Update $\phi_c \to \phi_c'$ with acceptance probability:

$$\min\left[1, \frac{\prod_{t_i \in c} F_r(t_i|\phi_c')}{\prod_{t_i \in c} F_r(t_i|\phi_c)}\right].$$

    Otherwise keep ϕ_c unchanged.
  end
end

The MCMC procedure in Algorithm 2 is iterated $K = 15,000$ times for a given a data set $\boldsymbol{t}$ collected at ramp rate r. The Markov chain sequence $(\boldsymbol{c}^{(j)}, \boldsymbol{\phi}^{(j)})$, $j = K_b, \ldots, K$ is then used to make estimates in the following manner. For each $\boldsymbol{c}^{(j)}$ in the chain, we create an estimate $\tilde{\boldsymbol{p}}(\boldsymbol{c}^{(j)})$ of the state probabilities via Eq. (1.28). This in turn allows us to estimate the effective dimension $D_{\text{eff}}(\tilde{\boldsymbol{p}}(\boldsymbol{c}^{(j)}))$ using Eq. (3.6). We construct a histogram of the effective dimensions, with bin size 0.05, and then identify the indices j that fall into the tallest bin. The mean of $\tilde{\boldsymbol{p}}(\boldsymbol{c}^{(j)})$ for all j in that bin forms our final estimate of the state probability vector for that data set. Because the $\tilde{\boldsymbol{p}}(\boldsymbol{c}^{(j)})$ could possibly have different lengths (though those that belong to the same bin are usually the same length), we pad the ends of the vectors with zeros as necessary. We can in a similar manner estimate the Bell parameters $\boldsymbol{\phi}$.

For the hyperparameter α we settled on the value $\alpha = 1$ by comparing results from smaller and larger α. We tested a modified version of the algorithm that automatically updates α as part of the MCMC iteration [28], did not see any improvement in accuracy. We chose $m = 2$ based on the recommendation of Ref. [29], as this provided a combination of good accuracy and relatively fast iteration time (since the MCMC procedure slows down as m is made larger).

3.7 Results

3.7.1 Test data set

The test data sets have the same form as the training data sets described above, except a different size: for each $M = 1, \ldots, M_{\text{max}}$, we generated 4000 different systems defined by $(\boldsymbol{p}, \boldsymbol{\phi})$, with two experimental observations $\boldsymbol{t}$ for each system. Thus we have a total of

48,000 examples of the form $(\boldsymbol{t}, \boldsymbol{p}, \boldsymbol{\phi})$. These test sets could then be used to evaluate the performance of our two machine learning approaches, as described below.

3.7.2 Performance of the deep learning algorithm

For $N = 200$, a typical experimental number of runs in actual experiments, the trained neural network performs very well in predicting the state probability distribution. For the test set, the average fidelity between the network prediction $\boldsymbol{q}$ and the true states probabilities $\boldsymbol{p}$ is $\overline{F(\boldsymbol{q}, \boldsymbol{p})} = 0.945$, and 97.6 percent of fidelities have values higher than 0.8. The average absolute effective dimension difference between predicted and true values is $\overline{\left|D_{\text{eff}}(\boldsymbol{q}) - D_{\text{eff}}(\boldsymbol{p})\right|} = 0.45$.

To better understand what these numbers mean, we show density histogram plots of predicted versus true values in Fig. 3.4 (top row) for the first three largest state probabilities p_1, p_2, p_3. (For p_2 and p_3 we only include systems that have these probability components.) Ideally, if our network performs perfectly, all the mass in each density histogram should land along the diagonal line. While there is spread, all the distributions are centered along the diagonal. In contrast, we can compare a null model that learns nothing from the input, and hence outputs a state distribution randomly. Here the mass would not distributed around the identity function, with the peaks of the histograms landing away from the diagonal, as shown in Fig. 3.4 (bottom). Though these results are all for ramping rate $r = 100$ pN/s, the performance is similar for r tested in a broader range $10 - 10^5$ pN/s (covering most AFM pulling experiments).

Fig 3.5 shows results for mean fidelity $\overline{F(\boldsymbol{q}, \boldsymbol{p})}$ and mean effective dimension difference $\overline{\left|D_{\text{eff}}(\boldsymbol{q}) - D_{\text{eff}}(\boldsymbol{p})\right|}$ with varying N, to gauge how well the algorithm can perform even with limited experimental information. There is a broad plateau for both measures for $N \gtrsim 100$, with only incremental improvements for larger N. The typical range of

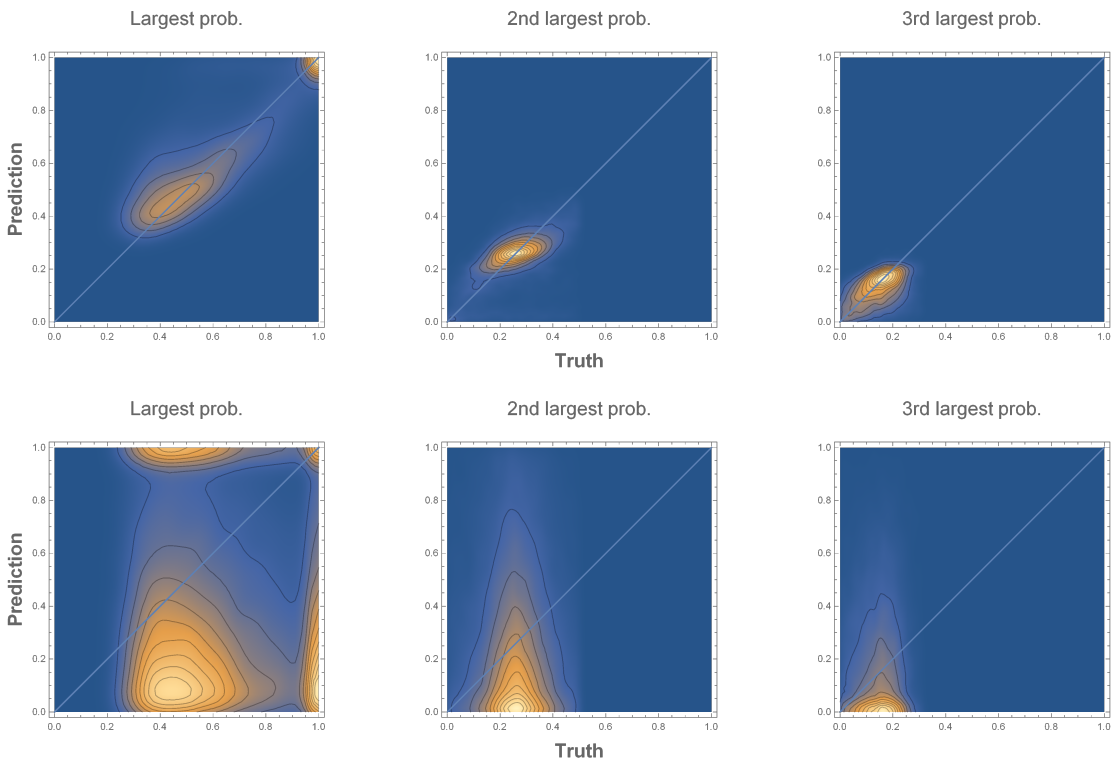

Figure 3.4. (**Top**) Density histogram plots for the first three components of the state probabilities predicted by the deep learning approach. "Truth" represent non-zero components of state probability vectors p_1, p_2, p_3 respectively while "Prediction" corresponds to the neural network output q_1, q_2, q_3. Blue solid lines: identity function. Ideally, a perfect prediction is achieved if all mass is on the blue line. Clearly, our trained network performs well in predicting the state probabilities, since most predictions are distributed in the vicinity of the identity function. The hyperparameter choices are: input size $N = 200$, total state number cutoff $M_{\max} = 6$. The ramping rate $r = 100$ pN/s for both training and test set. (**Bottom**) Density histogram plots of predictions from purely random guesses. In contrast to our deep learning model, the mass is not distributed around the identity function, with the peaks of the histograms lying in the off-diagonal regions.

experimental observations is $N \sim \mathcal{O}(100)$, and we see the algorithm performs robustly, without a strong dependence on precise value of N. For smaller N there is a gradual dropoff in performance that becomes quite steep for $N \lesssim 25$. At extremely small values

of the N, the network output is close to the mean state probability vector of the entire training set. This is effectively an educated guess in the absence of more data to determine the specific underlying system.

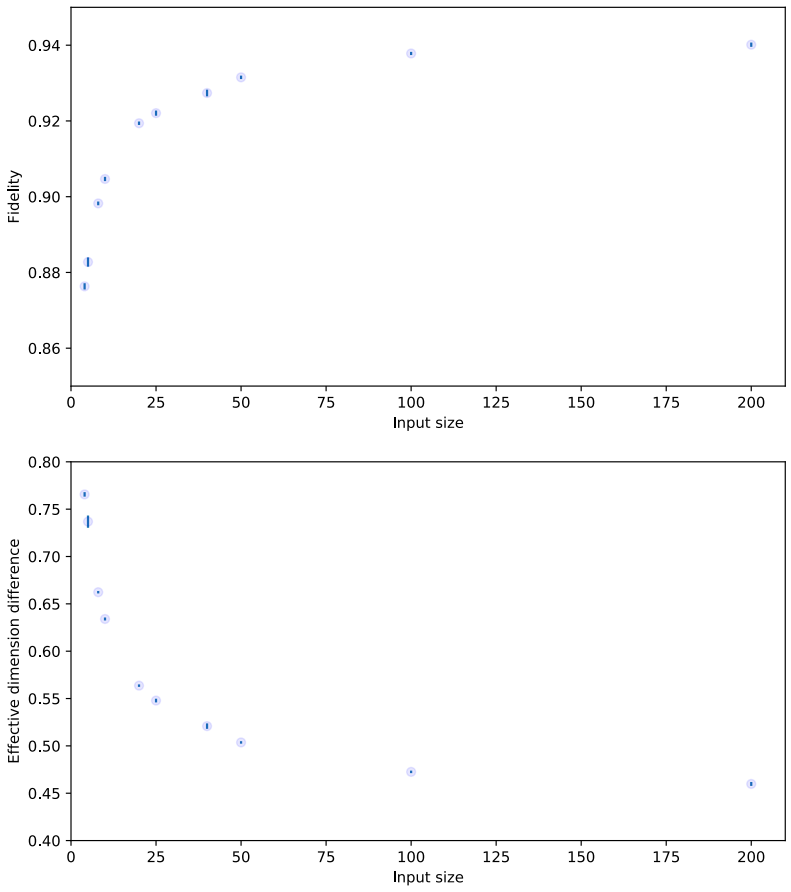

Figure 3.5. Performance of deep learning algorithm as a function of input size N. The performance metrics are average fidelity $\overline{F(\boldsymbol{q},\boldsymbol{p})}$ (**Top**) and average absolute effective dimension difference $\overline{\left|D_{\text{eff}}(\boldsymbol{q}) - D_{\text{eff}}(\boldsymbol{p})\right|}$ (**Bottom**). Error bars are standard error of the mean calculated from the results of 20 different test sets for each point.

In actual experiments, the rupture time resolution will be limited by thermal fluctuations of the AFM cantilevers, which lead to uncertainty in measurements of the rupture force f. Assuming a typical AFM cantilever with a spring constant on the order of $\omega_c \sim 10$ pN/nm, we can anticipate an error in force measurement of magnitude $\delta f = \sqrt{\omega_c k_B T} \sim 6$ pN [109]. To model the effects of noise, we added Gaussian fluctuations with standard deviation δf to our training and test sets. The error propagates into the rupture time distribution since $t = f/r$. For $N = 200$ and ramping rates r in the experimental range between 10 and 10^5 pN/s, the noise leads to no significant drop in the performance of our approach.

3.7.3 Performance of the non-parametric Bayesian learning algorithm and comparison to deep learning

Turning to the non-parametric Bayesian approach, for $N = 200$, $r = 100$ pN/s it outperforms deep learning, giving $\overline{F(\boldsymbol{q},\boldsymbol{p})} = 0.962$ and $\overline{\left|D_{\text{eff}}(\boldsymbol{q}) - D_{\text{eff}}(\boldsymbol{p})\right|} = 0.3$. We can see this clearly in Fig. 3.6, which shows density histograms at $N = 200$ for predicted vs true values of the three largest state probabilities. The non-parametric Bayesian results, shown in the top row, are more aligned along the diagonal than the deep learning results, plotted in the bottom row for comparison.

The advantage for the non-parametric Bayesian approach does not persist at all N. In Fig. 3.7 we compare performance metrics for both algorithms as a function of N. Deep learning does better for small numbers of observations, $N \lesssim 20$, but is superseded by the non-parametric Bayesian approach at larger N (more typical of actual experiments). Despite the impressiveness of the non-parametric Bayesian results at larger N, there still remains a technical challenge to real-world implementation. In order to make the algorithm capable of handling data corrupted by thermal fluctuation noise,

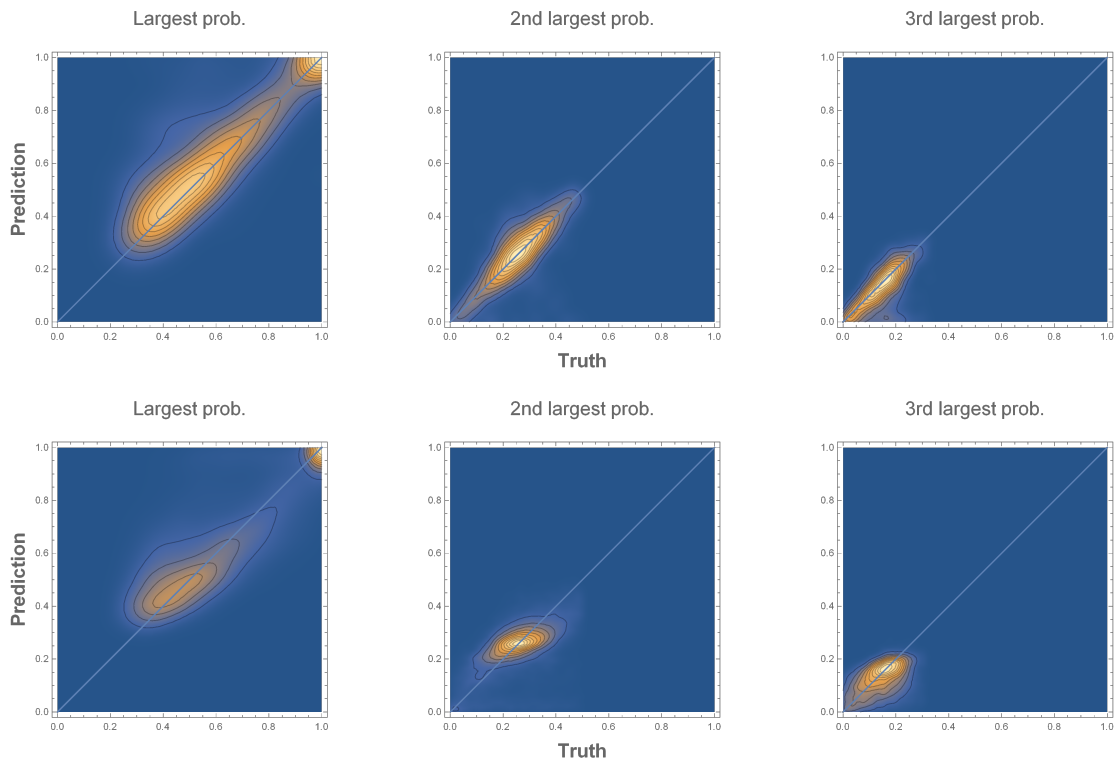

Figure 3.6. Predicted vs true value density histograms for the non-parametric Bayesian learning algorithm (**Top**) and deep learning algorithm (**Bottom**) with $N = 200$, $M_{\max} = 6$. "Truth" represents non-zero components of the state probability vector p_1, p_2, p_3, while "Prediction" corresponds to values q_1, q_2, q_3 estimated by our machine learning approaches. Blue solid lines: identity function.

an analytical form for a noise-broadened rupture time distribution $F_r(t|\phi)$ would have to be known. Convolving Eq. (3.3) with a Gaussian is not directly tractable, but future work could identify a suitable approximation. If noise is not taken into account, the original non-parametric Bayesian approach shows a significant performance dropoff: for $N = 200$ and the same kind of Gaussian noise with standard deviation $\delta f \sim 6$ pN discussed in the previous section, the mean fidelity $\overline{F(\boldsymbol{q}, \boldsymbol{p})}$ falls to 0.898 and the mean effective dimension difference $\overline{\left|D_{\text{eff}}(\boldsymbol{q}) - D_{\text{eff}}(\boldsymbol{p})\right|}$ grows to 0.63.

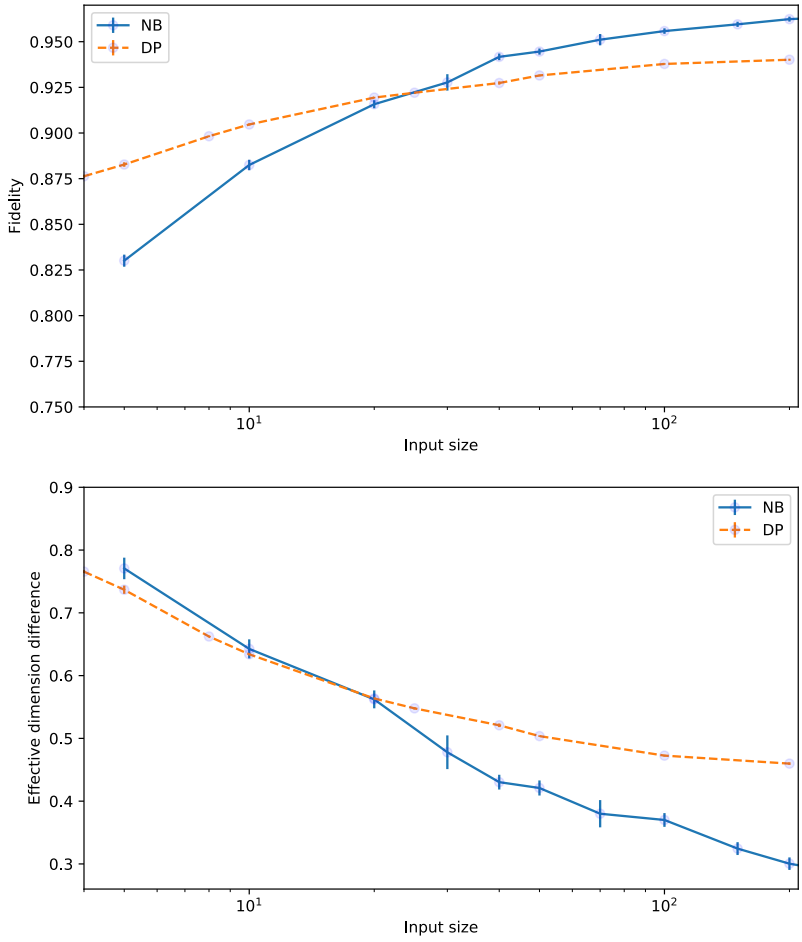

Figure 3.7. Performance comparison of the two machine learning algorithms for different input sizes. NB (orange dashed line) represents the non-parametric Bayesian learning algorithm. DP (blue solid line) represents deep learning algorithm. The performance metrics are average fidelity $\overline{F(\boldsymbol{q},\boldsymbol{p})}$ (**Top**) and average absolute effective dimension difference $\overline{\left|D_{\text{eff}}(\boldsymbol{q}) - D_{\text{eff}}(\boldsymbol{p})\right|}$ (**Bottom**). Error bars correspond to standard error of the mean calculated from the results of 20 different test sets.

There is also a significant difference between the two algorithms in terms of computational time. The deep learning approach is fairly light-weight. Training the data sets described above for $N = 200$ is accomplished in $\sim \mathcal{O}(100\,\mathrm{s})$ on GPU. Once the network is trained, analysis of a given input data set is nearly instantaneous, taking only $\sim \mathcal{O}(10^{-4}\,\mathrm{s})$. In contrast, while the non-parametric Bayesian method requires no training, the MCMC iteration for an input data set is more computationally expensive, taking ~ 50 s for $N = 200$ and $K = 15,000$. On the other hand, in the payoff for this longer computation is an estimate of both the state probabilities and their associated Bell model parameters.

The pros and cons of both machine learning algorithms can be summarized as follows:

Deep learning algorithm:

Pros:

- Straightforward and scalable: the simple neural network architecture is easy to implement, and can be readily enlarged to handle more complex tasks if required.
- Quick results: A trained network outputs the result for one experimental data set almost instantly.
- Works well for limited numbers of observations: the algorithm can handle small N better than the non-parametric Bayesian approach.
- Robust against noise: errors introduced by typical thermal fluctuations have almost no effect on the performance of this method.

Cons:

- A cutoff M_{max} for the number of possible states must be specified beforehand.

Bayesian non-parametric learning algorithm:

- **Pros:**
- Works well for larger numbers of observations: the algorithm outperforms deep learning for N typical of actual experiments.
- Handles arbitrary numbers of states: in principle, all possible state configuration can be explored.
- Estimation of state parameters: unlike the deep learning case, the parameters that characterize each state can be predicted.

  Cons:
- Conceptual and technical hurdles: non-parametric Bayesian inference is to date not widely familiar to physicists and biologists. The mathematical complexity makes it harder to modify the method, for example incorporating the handling of noise-corrupted data.
- Computational expense: processing a single data set is significantly slower compared to the deep learning algorithm.

3.8 Conclusion

In this chapter, we introduced two machine learning methods for characterizing heterogeneity in single bio-molecules based on observations collected from AFM pulling experiments. We focus on the state probability distribution $\boldsymbol{p}$, which allows us to directly quantify the degree of heterogeneity that exists in the system on the timescales of the pulling experiment. Though we did not explore it here, the estimation of state parameters is another crucial aspect, and a unique advantage of the non-parametric Bayesian method. The performance of both methods is demonstrated by synthetic data designed

to mimic a variety of experimental conditions. Our deep learning approach achieves excellent results without complex network architecture or a time-consuming training process. The more sophisticated non-parametric Bayesian approach does provide benefits for larger data sets, but is slower and more challenging to modify. The next stage is to begin testing both methods on actual experimental data, for example the data sets identified in Ref. [15]. Beyond AFM pulling data, both of our approaches can be easily generalized to data sets from other experimental systems (e.g. optical or magnetic tweezers). Identifying multiple, long-lived active conformations of bio-molecules is a critical step in understanding the full scope of their biological function, and our work provides new methods to shed light on this important problem.

4 Conclusions

In the first part of this thesis, we developed a theoretical approach to study the non-equilibrium statistical mechanics of biological signaling networks. Applying our approach to a typical canonical signaling circuit, the "enzymatic push-pull loop", we explored how the ability to transfer information through the circuit is constrained by energetic requirements. In the second part, we explored ways to analyze single-molecule heterogeneity with machine learning methods. Two different learning algorithms (both supervised and unsupervised) were introduced and applied to simulated AFM pulling experimental data, and we found that they both perform well in quantifying heterogeneity, though each method has its own advantages and tradeoffs.

In Chapter 2, we focused on the relation between information and energetic cost in a kinase-phosphatase push-pull loop signaling network. The information is quantified via the mutual information (MI) between the signal input and output, while the energetic cost takes the form of ATP consumption. The latter has two aspects: (i) the free energy $\Delta\mu$ for an ATP hydrolysis reaction, and (ii) the number of these reactions per unit time. With the help of the chemical Langevin approximation [7], our work shows that achieving the level of MI measured in experimental contexts requires crossing a threshold in both aspects of the cost. Optimal noise filter theory leads to a simple analytical

relationship capturing the tradeoffs between minimum ATP consumption, MI, and the bandwidth of the network (its ability to accurately transmit signals up to a certain maximum frequency). We show that a component of the yeast Hog1 signaling pathway has potentially minimized its energetic costs, lying close to the predicted optimality line. We rationalize this result by quantifying the evolutionary pressures that act on such systems, which can force them to optimize ATP consumption given a certain desired MI and bandwidth.

Chapter 3 details the deep learning and non-parametric Bayesian approaches we developed to analyze rupture time data from AFM pulling experiments. Covering a wide range of possible bio-molecular parameters and experimental settings, the results showed that both of our algorithms perform well in predicting the state probability distribution $\boldsymbol{p}$, a key aspect of heterogeneous systems. Even though our work focused on AFM experiments, our machine learning approach readily generalizes beyond the AFM pulling context, contributing one step to the long journey to understanding biological function at the single-molecule level.

4.1 Outlook

Many remarkable experiments over the last decade on both prokaryotic and eukaryotic signaling pathways have found they can transmit at most ~ 1 to 3 bits of information [43–50]. Trying to explain why different signaling networks can only transmit this low amount information was a crucial part of the motivation for our cell signaling work. As shown in this thesis, this has opened up another, unexpected aspect of the problem: for particular parameter combinations a signaling system can be optimal, in the sense that it can achieve a certain target value of mutual information, while exhibiting

the maximum possible bandwidth with the minimal ATP consumption. Every aspect of our theory can be directly tested, assuming the enzymatic parameters for individual systems are carefully measured. We already have some intriguing results for the yeast Hog1 system, but it would be valuable to revisit the experimental systems where mutual information was directly quantified. How many of these are operating near their performance limits? Can we understand the evolutionary pressures that may have brought them there?

For our exploration of single-molecule heterogeneity, the logical next step is to begin testing on actual empirical data. Even though we have tried to design biologically plausible synthetic data sets, there are still likely lessons to be learned from tackling the real thing. Both algorithms also have room for improvement: for the deep learning algorithm, finding an effective way to incorporate state parameter prediction can make it a even better approach; for our non-parametric Bayesian learning approach, accounting for the effects of noise in the data analysis would be a significant enhancement.

Complete References

[1] J-P Sauvage. Molecular machines and motors, volume 99. Springer, 2003.

[2] Gerard Manning, David B Whyte, Ricardo Martinez, Tony Hunter, and Sucha Sudarsanam. The protein kinase complement of the human genome. Science, 298(5600):1912–1934, 2002.

[3] Alan Fersht. Structure and mechanism in protein science: a guide to enzyme catalysis and protein folding. Macmillan, 1999.

[4] Hong Qian. Cooperativity and specificity in enzyme kinetics: a single-molecule time-based perspective. Biophys. J., 95(1):10–17, 2008.

[5] Ron Milo and Rob Phillips. Cell biology by the numbers. Garland Science, 2015.

[6] Thomas M Cover and Joy A Thomas. Elements of information theory. John Wiley & Sons, 2012.

[7] Daniel T Gillespie. The chemical langevin equation. The Journal of Chemical Physics, 113(1):297–306, 2000.

[8] H Peter Lu, Luying Xun, and X Sunney Xie. Single-molecule enzymatic dynamics. Science, 282(5395):1877–1882, 1998.

[9] Antoine M Van Oijen, Paul C Blainey, Donald J Crampton, Charles C Richardson, Tom Ellenberger, and X Sunney Xie. Single-molecule kinetics of λ exonuclease reveal base dependence and dynamic disorder. Science, 301(5637):1235–1238, 2003.

[10] Brian P English, Wei Min, Antoine M Van Oijen, Kang Taek Lee, Guobin Luo, Hongye Sun, Binny J Cherayil, SC Kou, and X Sunney Xie. Ever-fluctuating single enzyme molecules: Michaelis-menten equation revisited. Nature chemical biology, 2(2):87–94, 2006.

[11] Xiaowei Zhuang, Harold Kim, Miguel JB Pereira, Hazen P Babcock, Nils G Walter, and Steven Chu. Correlating structural dynamics and function in single ribozyme molecules. Science, 296(5572):1473–1476, 2002.

[12] Sergey V Solomatin, Max Greenfeld, Steven Chu, and Daniel Herschlag. Multiple native states reveal persistent ruggedness of an rna folding landscape. Nature, 463(7281):681–684, 2010.

[13] Bian Liu, Ronald J Baskin, and Stephen C Kowalczykowski. Dna unwinding heterogeneity by recbcd results from static molecules able to equilibrate. Nature, 500(7463):482–485, 2013.

[14] Krishna K Sarangapani, Bryan T Marshall, Rodger P McEver, and Cheng Zhu. Molecular stiffness of selectins. Journal of Biological Chemistry, 286(11):9567–9576, 2011.

[15] Michael Hinczewski, Changbong Hyeon, and Devarajan Thirumalai. Directly measuring single-molecule heterogeneity using force spectroscopy. Proceedings of the National Academy of Sciences, 113(27):E3852–E3861, 2016.

[16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436–444, 2015.

[17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT press, 2016.

[18] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. The elements of statistical learning, volume 1. Springer series in statistics New York, 2001.

[19] Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeff Dean, and Andrew Y. Ng. Building high-level features using large scale unsupervised learning, 2011.

[20] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4):115–133, 1943.

[21] John McCarthy, Marvin L Minsky, Nathaniel Rochester, and Claude E Shannon. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. AI magazine, 27(4):12–12, 2006.

[22] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. The Journal of physiology, 117(4):500, 1952.

[23] Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In 2009 IEEE 12th international conference on computer vision, pages 2146–2153. IEEE, 2009.

[24] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In ICML, 2010.

[25] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In Icml, 2011.

[26] Charles E Antoniak. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. The annals of statistics, pages 1152–1174, 1974.

[27] Thomas S Ferguson. A bayesian analysis of some nonparametric problems. The annals of statistics, pages 209–230, 1973.

[28] Michael D Escobar and Mike West. Bayesian density estimation and inference using mixtures. Journal of the american statistical association, 90(430):577–588, 1995.

[29] Radford M Neal. Markov chain sampling methods for dirichlet process mixture models. Journal of computational and graphical statistics, 9(2):249–265, 2000.

[30] George I Bell. Models for the specific adhesion of cells to cells. Science, 200(4342):618–627, 1978.

[31] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. Biometrika, 57:97, 1970.

[32] Joshua S Speagle. A conceptual introduction to markov chain monte carlo methods. arXiv preprint arXiv:1909.12313, 2019.

[33] Jim Pitman. Combinatorial Stochastic Processes: Ecole d'Eté de Probabilités de Saint-Flour XXXII-2002. Springer, 2006.

[34] Gábor Balázsi, Alexander van Oudenaarden, and James J Collins. Cellular decision making and biological noise: from microbes to mammals. Cell, 144(6):910–925, 2011.

[35] Christopher C Govern and Pieter Rein ten Wolde. Optimal resource allocation in cellular sensing systems. Proc. Natl. Acad. Sci., 111(49):17486–17491, 2014.

[36] Pieter Rein ten Wolde, Nils B Becker, Thomas E Ouldridge, and Andrew Mugler. Fundamental limits to cellular sensing. J. Stat. Phys., 162(5):1395–1424, 2016.

[37] Ganhui Lan and Yuhai Tu. Information processing in bacteria: memory, computation, and statistical physics: a key issues review. Rep. Prog. Phys., 79(5):052601, 2016.

[38] Catriona Y Logan and Roel Nusse. The wnt signaling pathway in development and disease. Annu. Rev. Cell Dev. Biol., 20:781–810, 2004.

[39] D Williams Parsons, Tian-Li Wang, Yardena Samuels, Alberto Bardelli, Jordan M Cummins, Laura DeLong, Natalie Silliman, Janine Ptak, Steve Szabo, James KV Willson, et al. Colorectal cancer: mutations in a signalling pathway. Nature, 436(7052):792, 2005.

[40] Fazlul H. Sarkar, Yiwei Li, Zhiwei Wang, and Dejuan Kong. Cellular signaling perturbation by natural products. Cell. Sign., 21(11):1541 – 1547, 2009.

[41] Celine E. Riera, Carsten Merkwirth, C. Daniel De Magalhaes Filho, and Andrew Dillin. Signaling networks determining life span. Annu. Rev. Biochem., 85(1):35–64, 2016.

[42] Shinsuke Uda. Application of information theory in systems biology. Biophys. Rev., pages 1–8, 2020.

[43] Gašper Tkačik, Curtis G Callan, and William Bialek. Information flow and optimization in transcriptional regulation. Proc. Natl. Acad. Sci., 105(34):12265–12270, 2008.

[44] Raymond Cheong, Alex Rhee, Chiaochun Joanne Wang, Ilya Nemenman, and Andre Levchenko. Information transduction capacity of noisy biochemical signaling networks. Science, 334(6054):354–358, 2011.

[45] Shinsuke Uda, Takeshi H Saito, Takamasa Kudo, Toshiya Kokaji, Takaho Tsuchiya, Hiroyuki Kubota, Yasunori Komori, Yu-ichi Ozaki, and Shinya Kuroda. Robustness and compensation of information transmission of signaling pathways. Science, 341(6145):558–561, 2013.

[46] Margaritis Voliotis, Rebecca M Perrett, Chris McWilliams, Craig A McArdle, and Clive G Bowsher. Information transfer by leaky, heterogeneous, protein kinase signaling systems. Proc. Natl. Acad. Sci., 111(3):E326–E333, 2014.

[47] Jangir Selimkhanov, Brooks Taylor, Jason Yao, Anna Pilko, John Albeck, Alexander Hoffmann, Lev Tsimring, and Roy Wollman. Accurate information transmission through dynamic biochemical signaling networks. Science, 346(6215):1370–1373, 2014.

[48] Garrett D Potter, Tommy A Byrd, Andrew Mugler, and Bo Sun. Dynamic sampling and information encoding in biochemical networks. Biophys. J., 112(4):795–804, 2017.

[49] Ryan Suderman, John A Bachman, Adam Smith, Peter K Sorger, and Eric J Deeds. Fundamental trade-offs between information flow in single cells and cellular populations. Proc. Natl. Acad. Sci., 114(22):5755–5760, 2017.

[50] Amiran Keshelava, Gonzalo P Solis, Micha Hersch, Alexey Koval, Mikhail Kryuchkov, Sven Bergmann, and Vladimir L Katanaev. High capacity in g protein-coupled receptor signaling. Nat. Commun., 9(1):1–8, 2018.

[51] Claude E Shannon. A mathematical theory of communication, bell systems tech. J, 27:379–423, 1948.

[52] Yuansheng Cao, Hongli Wang, Qi Ouyang, and Yuhai Tu. The free-energy cost of accurate biochemical oscillations. Nat. Phys., 11(9):772–778, 2015.

[53] Yoshihiko Hasegawa. Optimal temporal patterns for dynamical cellular signaling. New J. Phys., 18(11):113031, 2016.

[54] Pankaj Mehta, Alex H Lang, and David J Schwab. Landauer in the age of synthetic biology: energy consumption and information processing in biochemical networks. J. Stat. Phys., 162(5):1153–1166, 2016.

[55] Thomas E Ouldridge, Christopher C Govern, and Pieter Rein ten Wolde. Thermodynamics of computational copying in biochemical systems. Phys. Rev. X, 7(2):021004, 2017.

[56] Nick Lane and William F Martin. The origin of membrane bioenergetics. Cell, 151(7):1406–1416, 2012.

[57] ER Stadtman and PB Chock. Superiority of interconvertible enzyme cascades in metabolic regulation: analysis of monocyclic systems. Proceedings of the National Academy of Sciences, 74(7):2761–2765, 1977.

[58] Albert Goldbeter and Daniel E Koshland. An amplified sensitivity arising from covalent modification in biological systems. Proceedings of the National Academy of Sciences, 78(11):6840–6844, 1981.

[59] Peter B Detwiler, Sharad Ramanathan, Anirvan Sengupta, and Boris I Shraiman. Engineering aspects of enzymatic signal transduction: photoreceptors in the retina. Biophysical Journal, 79(6):2801–2817, 2000.

[60] Reinhart Heinrich, Benjamin G Neel, and Tom A Rapoport. Mathematical models of protein kinase signal transduction. Molecular cell, 9(5):957–970, 2002.

[61] Michael Hinczewski and D Thirumalai. Cellular signaling networks function as generalized wiener-kolmogorov filters to suppress noise. Physical Review X, 4(4):041017, 2014.

[62] Michael Hinczewski and D. Thirumalai. Noise control in gene regulatory networks with negative feedback. J. Phys. Chem. B, advanced online publication, 2016.

[63] Michael Lynch and Georgi K Marinov. The bioenergetic costs of a gene. Proc. Natl. Acad. Sci., 112(51):15690–15695, 2015.

[64] E. Ilker and M. Hinczewski. Modeling the growth of organisms validates a general relation between metabolic costs and natural selection. Phys. Rev. Lett., 122:238101, 2019.

[65] Oliver E Sturm, Richard Orton, Joan Grindlay, Marc Birtwistle, Vladislav Vyshemirsky, David Gilbert, Muffy Calder, Andrew Pitt, Boris Kholodenko, and Walter Kolch. The mammalian mapk/erk pathway exhibits properties of a negative feedback amplifier. Sci. Signal., 3(153):ra90–ra90, 2010.

[66] Manju Saxena, Scott Williams, Kjetil Taskén, and Tomas Mustelin. Crosstalk between camp-dependent kinase and map kinase through a protein tyrosine phosphatase. Nat. Cell Biol., 1(5):305, 1999.

[67] Carlos Salazar and Thomas Höfer. Multisite protein phosphorylation–from molecular mechanisms to kinetic models. FEBS J., 276(12):3177–3198, 2009.

[68] Daniel T Gillespie. Exact stochastic simulation of coupled chemical reactions. The journal of physical chemistry, 81(25):2340–2361, 1977.

[69] Arren Bar-Even, Elad Noor, Yonatan Savir, Wolfram Liebermeister, Dan Davidi, Dan S Tawfik, and Ron Milo. The moderately efficient enzyme: evolutionary and physicochemical trends shaping enzyme parameters. Biochemistry, 50(21):4402–4410, 2011.

[70] Wolfram Liebermeister and Edda Klipp. Biochemical networks with uncertain parameters. IEE Proc.-Syst. Biol., 152(3):97–107, 2005.

[71] Mingcong Wang, Christina J Herrmann, Milan Simonovic, Damian Szklarczyk, and Christian von Mering. Version 4.0 of paxdb: protein abundance data, integrated across model organisms, tissues, and cell-lines. Proteomics, 15(18):3163–3168, 2015.

[72] UniProt Consortium. UniProt: a worldwide hub of protein knowledge. Nucleic Acids Res., 47(D1):D506–D515, 2018.

[73] Ulrike Wittig, Renate Kania, Martin Golebiewski, Maja Rey, Lei Shi, Lenneke Jong, Enkhjargal Algaa, Andreas Weidemann, Heidrun Sauer-Danzwith, Saqib Mir, et al. SABIO-RK—database for biochemical reaction kinetics. Nucleic Acids Res., 40(D1):D790–D796, 2011.

[74] Zhike Zi, Wolfram Liebermeister, and Edda Klipp. A quantitative study of the Hog1 MAPK response to fluctuating osmotic stress in Saccharomyces cerevisiae. PLOS ONE, 5(3):e9522, 2010.

[75] Pascal Hersen, Megan N. McClean, L. Mahadevan, and Sharad Ramanathan. Signal processing by the hog map kinase pathway. Proceedings of the National Academy of Sciences, 105(20):7165–7170, 2008.

[76] N. Wiener. Extrapolation, Interpolation and Smoothing of Stationary Times Series. Wiley, New York, 1949.

[77] A. N. Kolmogorov. Interpolation and extrapolation of stationary random sequences. Izv. Akad. Nauk SSSR., Ser. Mat., 5:3–14, 1941.

[78] H. W. Bode and C. E. Shannon. A simplified derivation of linear least square smoothing and prediction theory. Proc. Inst. Radio. Engin., 38(4):417–425, 1950.

[79] Nils B Becker, Andrew Mugler, and Pieter Rein ten Wolde. Optimal prediction by cellular signaling networks. Phys. Rev. Lett., 115(25), 258103, 2015.

[80] Christoph Zechner, Georg Seelig, Marc Rullan, and Mustafa Khammash. Molecular circuits for dynamic noise filtering. Proc. Natl. Acad. Sci. USA, 113(17):4729–4734, 2016.

[81] Himadri S Samanta, Michael Hinczewski, and DJPRE Thirumalai. Optimal information transfer in enzymatic networks: A field theoretic formulation. Phys. Rev. E, 96(1):012406, 2017.

[82] David Hathcock, James Sheehy, Casey Weisenberger, Efe Ilker, and Michael Hinczewski. Noise filtering and prediction in biological signaling networks. IEEE Trans. Mol. Biol. Multi-Scale Commun., 2(1):16–30, 2016.

[83] Brian Charlesworth. Effective population size and patterns of molecular evolution and variation. Nature Rev. Genet., 10(3):195, 2009.

[84] Jerome T. Mettetal, Dale Muzzey, Carlos Gómez-Uribe, and Alexander van Oudenaarden. The frequency dependence of osmo-adaptation in saccharomyces cerevisiae. Science, 319(5862):482–484, 2008.

[85] John H Gillespie. Population genetics: a concise guide. JHU Press, 2010.

[86] Motoo Kimura. On the probability of fixation of mutant genes in a population. Genetics, 47(6):713–719, 1962.

[87] Leslie E Orgel and Francis HC Crick. Selfish dna: the ultimate parasite. Nature, 284(5757):604, 1980.

[88] Andreas Wagner. Energy constraints on the evolution of gene expression. Mol. Biol. Evol., 22(6):1365–1374, 2005.

[89] Gita Mahmoudabadi, Ron Milo, and Rob Phillips. Energetic cost of building a virus. Proc. Natl. Acad. Sci., 114:E4324–E4333, 2017.

[90] Isheng J Tsai, Douda Bensasson, Austin Burt, and Vassiliki Koufopanou. Population genomics of the wild yeast saccharomyces paradoxus: quantifying the life cycle. Proc. Natl. Acad. Sci., 105(12):4957–4962, 2008.

[91] Ron Milo. What is the total number of protein molecules per cell volume? a call to rethink some published values. Bioessays, 35(12):1050–1055, 2013.

[92] Birgit Schoeberl, Claudia Eichler-Jonsson, Ernst Dieter Gilles, and Gertraud Müller. Computational modeling of the dynamics of the map kinase cascade activated by surface and internalized EGF receptors. Nat. Biotech., 20(4):370–375, 2002.

[93] Christopher P Mattison and Irene M Ota. Two protein tyrosine phosphatases, Ptp2 and Ptp3, modulate the subcellular localization of the Hog1 MAP kinase in yeast. Genes Dev., 14(10):1229–1235, 2000.

[94] Sina Ghaemmaghami, Won-Ki Huh, Kiowa Bower, Russell W Howson, Archana Belle, Noah Dephoure, Erin K O'Shea, and Jonathan S Weissman. Global analysis of protein expression in yeast. Nature, 425(6959):737–741, 2003.

[95] Erik Gawehn, Jan A Hiss, and Gisbert Schneider. Deep learning in drug discovery. Molecular informatics, 35(1):3–14, 2016.

[96] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. Science advances, 4(7):eaap7885, 2018.

[97] Keegan E Hines. A primer on bayesian inference for biophysical systems. Biophysical journal, 108(9):2103–2113, 2015.

[98] Christopher P Calderon and Kerry Bloom. Inferring latent states and refining force estimates via hierarchical dirichlet process modeling in single particle tracking experiments. PloS one, 10(9):e0137633, 2015.

[99] Keegan E Hines, John R Bankston, and Richard W Aldrich. Analyzing single-molecule time series via nonparametric bayesian inference. Biophysical journal, 108(3):540–556, 2015.

[100] Ioannis Sgouralis and Steve Pressé. An introduction to infinite hmms for single-molecule data analysis. Biophysical journal, 112(10):2021–2029, 2017.

[101] Ioannis Sgouralis and Steve Pressé. Icon: an adaptation of infinite hmms for time traces with drift. Biophysical journal, 112(10):2117–2126, 2017.

[102] Ioannis Sgouralis, Miles Whitmore, Lisa Lapidus, Matthew J Comstock, and Steve Pressé. Single molecule force spectroscopy at high data acquisition: A bayesian nonparametric analysis. The Journal of chemical physics, 148(12):123320, 2018.

[103] Ioannis Sgouralis, Shreya Madaan, Franky Djutanta, Rachael Kha, Rizal F Hariadi, and Steve Pressé. A bayesian nonparametric approach to single molecule forster resonance energy transfer. The Journal of Physical Chemistry B, 123(3):675–688, 2018.

[104] Sina Jazani, Ioannis Sgouralis, Omer M Shafraz, Marcia Levitus, Sanjeevi Sivasankar, and Steve Pressé. An alternative framework for fluorescence correlation spectroscopy. Nature communications, 10(1):1–10, 2019.

[105] Meysam Tavakoli, Sina Jazani, Ioannis Sgouralis, Omer M Shafraz, Sanjeevi Sivasankar, Bryan Donaphon, Marcia Levitus, and Steve Pressé. Pitching single-focus confocal data analysis one photon at a time with bayesian nonparametrics. Physical Review X, 10(1):011021, 2020.

[106] Olga K Dudko, Gerhard Hummer, and Attila Szabo. Theory, analysis, and interpretation of single-molecule force spectroscopy experiments. Proceedings of the National Academy of Sciences, 105(41):15755–15760, 2008.

[107] Richard Jozsa. Fidelity for mixed quantum states. Journal of modern optics, 41(12):2315–2323, 1994.

[108] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[109] Keir C Neuman and Attila Nagy. Single-molecule force spectroscopy: optical tweezers, magnetic tweezers and atomic force microscopy. Nature methods, 5(6):491–505, 2008.