Solving the "Blind men and the elephant problem": Additive deep learning of complex high dimensional models from partial faceted datasets

Yufei Wu [1,2], Pei-Hsun Wu [2,3], Allison Chambliss[4], Denis Wirtz[2,3], Sean X. Sun [1,2,5,*]

[1]Department of Mechanical Engineering, Johns Hopkins University, Baltimore, Maryland, USA

[2]Institute for NanoBioTechnology, Johns Hopkins University, Baltimore, Maryland, USA

[3]Department of Chemical and Biomolecular Engineering, Johns Hopkins University, Baltimore, MD, USA

[4]Department of Pathology & Laboratory Medicine, University of California Los Angeles, Los Angeles, California, USA

[5]Center for Cell Dynamics, Johns Hopkins School of Medicine, Baltimore, Maryland, USA

[*]To whom correspondence should be addressed; E-mail: ssun@jhu.edu.

Biological systems are complex networks involving tens of thousands of interacting molecular components, and measurable biological functions are emerging properties of these complex networks. Many quantitative studies in biology attempt to connect biological function with molecular components and genes, in the process developing mechanistic understanding. However, it is challenging to quantify the contribution of all components to the biological function simultaneously, especially at the single cell level. Instead, in typical experiments, only a subset of the variables (or facet) is measured. This makes it difficult to obtain a complete and unbiased

understanding of the network and how different components of the network cooperatively contribute to the biological function. In this paper, we explore a machine learning approach to combine different facets of data and obtain a complete picture of the biological system based on conditional distributions from faceted data subsets. Both a polynomial regression approach and a neural network approach are developed and examined with two set of concrete examples: A mechanical spring network system deforming under external forces and a small (8-dimensions) biological network including the cellular senescence marker P53. In the later example, single cell data is collected to validate the machine learning approach. We find that the full system is successfully reconstructed from faceted data in both examples. We further discuss the additive property of the model, where the model predictive accuracy increases with increasing number of simultaneously measured variables (dimension of subsets). Our model provides a systematic and novel approach to integrate different pieces of experimental information to reconstruct complex high dimensional systems, arriving at an unbiased and wholistic model of biological function.

Introduction

As told through centuries, the "Blind Men and the Elephant" is a fable of blind individuals attempting to comprehend the appearance and nature of an elephant by independent exploration (Fig. 1 (a)). Each individual has limited information and understanding, acquired through independent experience. However, by sharing, comparing, and synthe-

sizing their experiences, the group can gain a more comprehensive understanding of the elephant as a whole. Similarly, biological systems are complex networks with thousands of interacting molecular components [1, 2, 3]. Biological function and disfunction are often emergent properties of these complex networks. It can be challenging to quantify the contributions of all variables to the biological function simultaneously, making it difficult to obtain a full understanding of the system. More often, a subset of variables are measured and quantified, obtaining a projection (or facet) of the relationship between the biological output and the underlying variable. Therefore, just as in the "Blind men and the elephant" example, it is desirable to reconstruct the full relationship between the biological output and all the underlying variables from many sets of faceted data.

With advancements in machine learning (ML) and artificial intelligence (AI), there are now many methods that can predict outcomes from complex high dimensional data [4, 5, 6, 7]. However, in a typical biological experiment, the full space of underlying variables are almost never measured. Here we present a machine learning-based method to reconstruct the complete biological network from faceted data sets. The method allows for incremental improvement of the learned network, and is a systematic method of obtaining the global predictive model from multiple independent measurements and observations. When new hidden variables are discovered, new measurements can be added to the existing model to improve the model and predictions.

The basic biological unit is a single cell. Each cell is characterized by its proteome, genetic material, and other components such as lipids, small molecules, ions, and so on. Therefore, the underlying variable that describes the single cell, $\boldsymbol{x} = (x_1, x_2, x_3, \dots)$, is a high dimensional vector, where x_i is the quantity of the i-th component. The minimal number variables that define $\boldsymbol{x}$ is the proteome composition, or the number of expressed proteins in the cell, since given the same genetic sequence, the proteome composition

should determine the number of small molecule, lipid, ionic contents of the cell, as well as post-translationally modified forms of proteins. However, proteome composition itself probably does not fully specify biological function, since environmental chemical [8, 9], mechanical [10, 11], and electrical variables [12] also contribute. Therefore, $\boldsymbol{x}$ minimally will contain the expression levels of all genes and environmental variables.

If $\boldsymbol{x}$ is defined as the expression levels of genes, then the distribution of $\boldsymbol{x}$, $\rho(\boldsymbol{x})$, is often referred to a 'gene network' [13, 14]. In the context of gene regulatory networks, the discussions in our paper also applies (See Example 2: P53 network).

At the simplest level, a particular biological function/observable, F, is a function of the underlying variable: $F(\boldsymbol{x})$. For example, F could be the cell size, the cell cycle length, the growth rate, or the cell migrations speed, which should be measured at the **single cell level**. This is because much of recent work has demonstrated that there is additional complexity and phenotypic variation, even for isogenic cells [15, 16]. The reasons for this is complex, and could encompass epigenetic mechanisms and cellular memory [17, 18]. Therefore, $F(\boldsymbol{x})$ is a complex mapping from biological variables to biological function. It should be noted that recent advancements in AI and machine learning in fact has solved the high dimensional regression problem. If the data for $F(\boldsymbol{x})$ is available, then AI can now use neural networks or other types of methods that maps biological variables to biological function. The problem, therefore, is not the lack of methods to find $F(\boldsymbol{x})$. The problem is the lack of multi-dimensional methods that obtain data for all relevant $\boldsymbol{x}$, and measure F simultaneously at the single cell level.

Thus, the function $F(\boldsymbol{x})$ is difficult to learn in an unbiased way, and there are no systematic efforts to map F for major biological problems of interest. In most experiments, such as flow cytometry or Western blot experiments, only a few of the x_i out of thousands are quantified in a meaningful way. Moreover, it is typical that each researcher measures a

different subset of x_i's, and therefore is study a particular 'facet' of the problem, precisely the problem identified in the "blind men" story. The global picture is generally missing. There have been extensive studies in the ML field on system reconstruction from partial data sets based on eigenvectors of the system [19, 20]. However, it is desirable to have a method that can combine data from all individual facets, and progressively arrive at a global picture.

There are now increasing number of experimental methods to quantify cell components (e.g., RNAseq [21, 22], protein secretome [23] and morphological data [24, 25]) at the single cell level. For example, single cell RNAseq quantifies RNA at the genome-wide level. However, mRNA levels do not easily translate to proteomic composition [26, 27, 28], and no biological observable, F, is typically measured at the single cell level during sequencing. On the other hand, methods such as flow cytometry, Western blots, and immunohistochemistry allow one to examine a handful of proteins at a quantitative level, but it is generally difficult to examine biological function or observables at the single cell level. There are now highly accurate methods to measure cell size, cell contractility, and cell cycle at the single cell level. It remains to be seen if single cell methods can be combined with single cell measurements to produce truly predictive models of biological function.

In this paper, we first describe the general idea of faceted learning based on multiple data subsets of the same problem. We then illustrate the method using machine learning models based on polynomial regression and neural networks, respectively. Two concrete examples are discussed: A mechanical spring network system and a small biological network including the cellular senescence marker P53. Full system is successfully reconstructed from faceted data for both problems. Interestingly, we find that the mechanism regulating P53 level is the same for cells in different growth conditions. The only

difference is the underlying proteome distribution of network components. Our method separates the regulatory network that govern p53 level and the intrinsic distribution of the input variables. The polynomial regression model also allows us to explore mechanistic aspects of the network, whether components of the network act synergistically or antagonistically. We also discuss the additive property of faceted approach, where the model accuracy increases with increasing number of simultaneously measured variables (dimension of subsets). Our approach provides a novel method utilizing conditional distribution to integrate different pieces of information to reconstruct complex high dimensional biological systems.

Reconstructing the systems model from facets of probability distributions: Statement of the problem

We consider a system described by the function $y = F(\boldsymbol{x}; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a set of model parameters. For simplicity, we assume that y is an one-dimensional output and $\boldsymbol{x}$ is a d-dimensional input vector (e.g., for the system of a cell, cell volume is a function of protein content and kinase activity.) (Fig. 1 (b)). In experiments, we assume only p ($p < d$) variables of $\boldsymbol{x}$ and biological output y can be measured simultaneously. In general $p > 1$, which provides information about the correlation among different input variables ($\boldsymbol{x}$). It is also possible to perform multiple measurements to obtain different subsets of variables $(\boldsymbol{x}, y)$. Note that data-driven methods of manifold learning using principal component analysis (PCA) for learning models of $(\boldsymbol{x}, y)$ has been investigated extensively [29, 30]. Here we take these available methods as given.

Experimental measurements will generate probability distributions of $(\boldsymbol{x}, y)$. In the biological context, each instance of $(\boldsymbol{x}, y)$ arise from a single cell, and many cells are

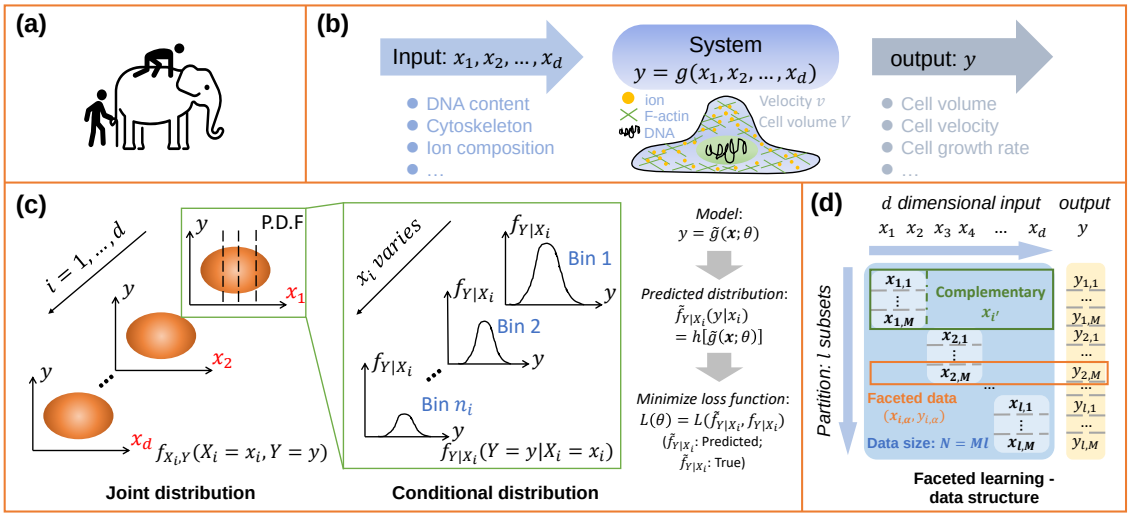

Figure 1: (a) Blind men and the elephant problem. Each observer measures a facet of the problem, and therefore receives a biased view. Combining data from all observers will generate a full model. (b) A biological function is a mapping from cell components to an observable, or output. (c) Biological network model reconstruction from mapping of data distribution functions. The original data is the joint probability distributions of partial input and output. We dissect the joint distributions into several consecutive conditional distributions and directly fit the conditional distribution to obtain model parameters. (d) Data structure in the faceted learning procedure. l faceted data sets are collected, each containing only partial dimensions of the input $\boldsymbol{x}$ and output y. Each data set contains M data points, with $N = M \times l$ total data points.

typically measured in a single experiment. Therefore, the mean biological output is

$$\langle F \rangle = \int d\boldsymbol{x} F(\boldsymbol{x}) \rho(\boldsymbol{x}) \tag{1}$$

We assume that it is possible to eventually measure the $d \times d$ covariance matrix of $\boldsymbol{x}$ and the mean value of the input variable $\boldsymbol{x}$, denoted by $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$, respectively. We denote all the d input variables as a universal set $U = \{x_1, x_2, ..., x_d\}$. Assume that each measurement includes p input variables, and we denote the simultaneously measured variables as S_i, which is a subset of U. There are in total $n_s = \binom{d}{p}$ different subsets ($i < n_s$) and i is the index of measurements. In principle, we can perform measurements

over all possible subsets. However, for simplicity, in the following discussion, we partition U into $l = d/p$ subsets and only use these l subsets for system reconstruction. The subsets are denoted by S_i $(i \leq l)$ and satisfy: $\cup_{i=1}^{l} S_i = U, S_i \cap S_j = \emptyset$. For each subset S_i, let $S_i' = U \backslash S_i$ be the complement. Assume we have l sets of experimental data covering the whole set as described above and each data set is composed of M data points: $(\boldsymbol{x}_{i,\alpha}, y_{i,\alpha})$ $(i \leq l, \alpha \leq M)$. Here $\boldsymbol{x}_i$ is a vector containing all variables in subset S_i, and the subscript α is the index of the data point. $y_{i,\alpha}$ is the output variable corresponding to $x_{i,\alpha}$. Similarly, we define $\boldsymbol{x_{i'}}$ as a vector containing all variables in the complementary set S_i'. **These data sets are l-facets of the full system** (Fig. 1(d)). We desire to approximate the full model of the system by $y = \tilde{F}(\boldsymbol{x})$ from these l sets of partial data and the measured statistical information of input variables.

We wish to reconstruct the full system model from the conditional probability distributions of output variables with fixed input variables. For each data set $(\boldsymbol{x_i}, y_i)$, we have the conditional distribution

$$f_i(y|\boldsymbol{x_i}) = \frac{\int \Pi(\boldsymbol{x}, y) d\boldsymbol{x_{i'}}}{\int \rho(\boldsymbol{x}) d\boldsymbol{x_{i'}}}. \tag{2}$$

Here f_i is the conditional probability of variable y given fixed $\boldsymbol{x_i}$, Π is the joint probability distribution of $\boldsymbol{x}, y$ of the full system and ρ is the joint probability distribution of only $\boldsymbol{x}$. $\Pi(\boldsymbol{x}, y)$ contains information for both the distribution of underlying variables $(\boldsymbol{x})$ and the dependence of y on $\boldsymbol{x}$. In principle, once the joint distribution of $\boldsymbol{x}, y$ is obtained, we know the mapping between $\boldsymbol{x}$ and y. However, Π is never explicitly measured in experiments. Only the facets, or $\rho_i(\boldsymbol{x}_i) = \int \rho(\boldsymbol{x}) d\boldsymbol{x_{i'}}$ and f_i are measured in experiments. By minimizing the difference between the predicted conditional distribution $(\hat{f}_i)$ and true distribution obtained from experimental data (f_i), we can obtain the best model parameters $\boldsymbol{\theta}$ (Fig.

1 (c)):

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \left(\sum_{i=1}^{l} \int [f_i(y|\boldsymbol{x_i}) - \hat{f}_i(y|\boldsymbol{x_i};\boldsymbol{\theta})]^2 d\boldsymbol{x_i} \right) \tag{3}$$

where f_i is the measured conditional distribution for the i-th partial (facet) data and $\hat{f}_i$ is the predicted distribution from our model. This represents the most unbiased model regression that includes all facets of the problem. One may also weigh the facets by their statistical confidence, or data quality, which is easily done in Eq. (3). In the following discussion, variables with hats imply predicted value based on assumed models.

Performing regression for the complete probability distribution function is sometimes not practical because the conditional distribution $f_i(y|\boldsymbol{x_i})$ is generally not analytic. We also would like to use deep learning and neural networks to parameterize the model. One possibility is to use the mean and the variance to approximate the distribution and minimize the differences in these two quantities with respect to model parameters, $\boldsymbol{\theta}$. This procedure is exact for systems with normally distributed data. The conditional expectation and variance are defined as: $L_i = \int y f_i(y|\boldsymbol{x_i}) dy$ and $V_i = \int (y - L_i)^2 f_i(y|\boldsymbol{x_i}) dy$. In practice, since we can not obtain analytical expression of the conditional distribution $f_i(y|\boldsymbol{x_i})$, we compute the predicted expectation and variance in terms of $\boldsymbol{x}$ based on the assumed model for output y ($\hat{y} = \hat{F}(\boldsymbol{x};\boldsymbol{\theta})$) and conditional distribution of $\boldsymbol{X_{i'}}$ when $\boldsymbol{X_i}$ is fixed ($\rho_i(\boldsymbol{x_{i'}}|\boldsymbol{x_i})$). Specifically, for each data set $(\boldsymbol{x_i}, y_i)$, we integrate the output function $F(\boldsymbol{x})$ over all the unknown variables $\boldsymbol{x_{i'}}$ with conditional probability distribution to get the conditional expectation and denote it by $\hat{L}_i(\boldsymbol{x_i})$. Moreover, we calculate the variance over all the unknown variables ($\boldsymbol{x_{i'}}$) while the known variables ($\boldsymbol{x_i}$) are fixed and denote it by $\hat{V}_i(\boldsymbol{x_i})$. The prediction accuracy can be improved by including higher order moments.

The conditional expectation and variance are related to faceted data as:

$$\hat{L}_i(\boldsymbol{x_i};\boldsymbol{\theta}) = \int \hat{F}(\boldsymbol{x};\boldsymbol{\theta})\rho_i(\boldsymbol{x_{i'}}|\boldsymbol{x_i})d\boldsymbol{x_{i'}} \tag{4}$$

$$\hat{V}_i(\boldsymbol{x_i};\boldsymbol{\theta}) = \int [\hat{F}(\boldsymbol{x};\boldsymbol{\theta}) - L_i(\boldsymbol{x_i})]^2\rho_i(\boldsymbol{x_{i'}}|\boldsymbol{x_i})d\boldsymbol{x_{i'}} \tag{5}$$

From the experimental data, we divide the independent variables $\boldsymbol{x_i}$ in each set of data into n_i consecutive bins and for each bin $[\boldsymbol{x}_{i,k}, \boldsymbol{x}_{i,k} + \boldsymbol{dx}](k \leq n_i)$, we calculate the mean value $L_i(\boldsymbol{x}_{i,k})$ and variance $V_i(\boldsymbol{x}_{i,k})$. The loss function is defined in the square error form as:

$$U = \sum_{i=1}^{l}\sum_{\alpha=1}^{M}[(L_i(\boldsymbol{x}_{i,\alpha}) - \hat{L}_i(\boldsymbol{x}_{i,\alpha}))^2 + (V_i(\boldsymbol{x}_{i,\alpha}) - \hat{V}_i(\boldsymbol{x}_{i,\alpha}))^2]. \tag{6}$$

The framework outlined above requires knowledge about the distribution of input variables $\boldsymbol{x}$. For many biological examples, the data is concentrated around the mean value and are close to the normal distribution. In our analysis, we first standardize the input and output data by: $\tilde{\boldsymbol{x}} = \boldsymbol{\Sigma}^{-1/2}\cdot(\boldsymbol{x}-\boldsymbol{\mu})$, where $\boldsymbol{\mu}$ is the mean value of the sample and $\boldsymbol{\Sigma}$ is the covariance matrix. After standardization, the mean value becomes zero and covariance matrix becomes the identity matrix. Therefore, the correlation between variables in $\rho(\boldsymbol{x})$ is removed in the transformed variables. For simplicity, in the following analysis, we assume that the variables are already standardized and follow the normal distribution $\boldsymbol{x} \sim N(0,1)$ and drop the tilde label if not specified. The underlying distribution is then

$$\hat{\rho}(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d}} e^{-\frac{1}{2}\tilde{\boldsymbol{x}}^T\boldsymbol{I}^{-1}\tilde{\boldsymbol{x}}} \tag{7}$$

where $\boldsymbol{I}$ is identity matrix after the standardization.

The Gaussian assumption for $\rho(\boldsymbol{x})$ is convenient for analytic manipulation, but in general the assumption is not valid. A more general approach is to use a Gaussian mixture model [31, 32], where we assume the probability distribution of $\boldsymbol{x}$ is the sum of

several Gaussians:

$$\hat{\rho}(\boldsymbol{x}) = \sum_{N} \frac{a_N}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_N)^T \boldsymbol{\Sigma}_N^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_N)} \tag{8}$$

where $(a_N, \boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$ are the weights and parameters of the N-th Gaussian. The Gaussian parameters can be optimized with respect to the measured faceted distributions. Specifically, for each measured facet $\boldsymbol{x_i}$, there is a marginal distribution $\rho(\boldsymbol{x_i})$. We use several Gaussian functions to fit $\rho(\boldsymbol{x_i})$ with parameters $(a_{N_i}, \boldsymbol{\mu}_{N_i}, \boldsymbol{\Sigma}_{N_i})$:

$$\hat{\rho}_i(\boldsymbol{x_i}) = \sum_{N_i} \frac{a_{N_i}}{\sqrt{(2\pi)^p |\Sigma|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_{N_i})^T \boldsymbol{\Sigma}_{N_i}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_{N_i})} \tag{9}$$

Since correlation information is removed in the normalized data, we can roughly assume that each measuring set is independent of others. We can then approximate the joint distribution of $\boldsymbol{x}$ as the product of the fitted marginal distributions of each faceted data set: $\hat{\rho}(\boldsymbol{x}) = \Pi_{i=1}^{l} \hat{\rho}_i(\boldsymbol{x_i})$.

Analytical case: Polynomial Models Based on Partial Data

For illustration purposes, we examine a polynomial model based on normally distributed data. The results are analytic, and therefore easily obtained. Also, due to concentrated property of many different kinds of data, we can sometimes approximate the output function using Taylor expansion up to the second order as:

$$\hat{F}(\boldsymbol{x}) = F_0 + \sum_{\alpha=1}^{d} F'_\alpha x_\alpha + \frac{1}{2} \sum_{\alpha=1}^{d} \sum_{\beta=1}^{d} F''_{\alpha\beta} x_\alpha x_\beta \tag{10}$$

From the Gaussian assumption, it is possible to compute the conditional mean value and variance explicitly. For each set of data, the conditional distribution of unknown variables when fixing the known variables also obeys normal distribution: $\rho(\boldsymbol{x_{i'}}|\boldsymbol{x_i}) = N(\bar{\boldsymbol{\mu}}^{(\boldsymbol{i})}, \bar{\Sigma}^{(i)})$,

where $\bar{\boldsymbol{\mu}}$ and $\bar{\boldsymbol{\Sigma}}$ are defined as follows: We first rearrange the d-dimensional column vector $\boldsymbol{x}$ as: $\boldsymbol{x} = (\boldsymbol{x}_{\boldsymbol{i'}}^{\boldsymbol{T}}, \boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{T}})^T$ and accordingly, $\boldsymbol{\Sigma}$ is arranged as follows ($\boldsymbol{\mu}$ is a null-vector):

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{i'i'} & \boldsymbol{\Sigma}_{i'i} \\ \boldsymbol{\Sigma}_{ii'} & \boldsymbol{\Sigma}_{ii} \end{pmatrix} \tag{11}$$

Then $\bar{\boldsymbol{\mu}}^{(\boldsymbol{i})}$ and $\bar{\boldsymbol{\Sigma}}^{(i)}$ can be expressed as:

$$\bar{\boldsymbol{\mu}}^{(\boldsymbol{i})} = \boldsymbol{\Sigma}_{i'i} \cdot {\boldsymbol{\Sigma}_{ii}}^{-1} \cdot \boldsymbol{x_i} \tag{12}$$

$$\bar{\boldsymbol{\Sigma}}^{(i)} = \boldsymbol{\Sigma}_{i'i'} - \boldsymbol{\Sigma}_{i'i} \cdot \boldsymbol{\Sigma}_{ii}^{-1} \cdot \boldsymbol{\Sigma}_{ii'} \tag{13}$$

Based on the conditional distribution, the mean output value when fixing $\boldsymbol{x_i}$ is calculated as:

$$\hat{L}_i(\boldsymbol{x_i}) = \int \hat{F}(\boldsymbol{x}) f_i(\boldsymbol{x_{i'}}|\boldsymbol{x_i}) d\boldsymbol{x_{i'}} \tag{14}$$

$$= \int (F_0 + \sum_{\alpha=1}^{d} F'_\alpha x_\alpha + \frac{1}{2}\sum_{\alpha=1}^{d}\sum_{\beta=1}^{d} F''_{\alpha\beta} x_\alpha x_\beta) \hat{f}_i(\boldsymbol{x_{i'}}|\boldsymbol{x_i}) d\boldsymbol{x_{i'}} \tag{15}$$

$$= F_0 + \sum_{\alpha=1}^{d} F'_\alpha M_\alpha^{(i)} + \frac{1}{2}\sum_{\alpha=1}^{d}\sum_{\beta=1}^{d} F''_{\alpha\beta}(C_{\alpha\beta}^{(i)} + M_\alpha^{(i)} M_\beta^{(i)}) \tag{16}$$

where the matrices $\boldsymbol{C}^{(i)}$ and $\boldsymbol{M}^{(i)}$ are as follows:

$$\boldsymbol{C}^{(i)} = \begin{pmatrix} \bar{\boldsymbol{\Sigma}}_{11}^{(i)} & \boldsymbol{0} & \bar{\boldsymbol{\Sigma}}_{12}^{(i)} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \bar{\boldsymbol{\Sigma}}_{21}^{(i)} & \boldsymbol{0} & \bar{\boldsymbol{\Sigma}}_{22}^{(i)} \end{pmatrix}, \boldsymbol{M}^{(k)} = \begin{pmatrix} \bar{\boldsymbol{\mu}}_{\boldsymbol{1}}^{(\boldsymbol{i})} \\ \boldsymbol{x_i} \\ \bar{\boldsymbol{\mu}}_{\boldsymbol{2}}^{(\boldsymbol{i})} \end{pmatrix}, \tag{17}$$

The positions of $\bar{\Sigma}_{11}^{(i)}, \bar{\Sigma}_{12}^{(i)}, \bar{\Sigma}_{21}^{(i)}, \bar{\Sigma}_{22}^{(i)}, \bar{\boldsymbol{\mu}}_{\boldsymbol{1}}^{(\boldsymbol{i})}, \bar{\boldsymbol{\mu}}_{\boldsymbol{2}}^{(\boldsymbol{i})}$ are determined by the indices of $\boldsymbol{x_{i'}}$ in the full vector $\boldsymbol{x}$. Similarly, the positions (columns and rows) of the inserted zeros in $C^{(i)}$ and $\boldsymbol{x_i}$ in $M^{(i)}$ correspond to the measured variable indices ($\boldsymbol{x_i}$). Furthermore, the variance of the predicted output value when fixing $\boldsymbol{x_i}$. We first calculate the first four moments of

the variable $\boldsymbol{x_{i'}}$:

$$E(x_\alpha) = M_\alpha^{(i)} \tag{18}$$

$$E(x_\alpha x_\beta) = C_{\alpha\beta}^{(i)} + M_\alpha^{(i)} M_\beta^{(i)} \tag{19}$$

$$E(x_\alpha x_\beta x_\gamma) = M_\alpha^{(i)} C_{\beta\gamma}^{(i)} + M_\beta^{(i)} C_{\alpha\gamma}^{(i)} + M_\gamma^{(i)} C_{\alpha\beta}^{(i)} + M_\alpha^{(i)} M_\beta^{(i)} M_\gamma^{(i)} \tag{20}$$

$$\begin{aligned} E(x_\alpha x_\beta x_\gamma x_\nu) = {} & C_{\alpha\beta}^{(i)} C_{\gamma\nu}^{(i)} + C_{\alpha\gamma}^{(i)} C_{\beta\nu}^{(i)} + C_{\alpha\nu}^{(i)} C_{\beta\gamma}^{(i)} \\ & + M_\alpha^{(i)} M_\beta^{(i)} C_{\gamma\nu}^{(i)} + M_\alpha^{(i)} M_\gamma^{(i)} C_{\beta\nu}^{(i)} + M_\alpha^{(i)} M_\nu^{(i)} C_{\beta\gamma}^{(i)} \\ & + M_\beta^{(i)} M_\gamma^{(i)} C_{\alpha\nu}^{(i)} + M_\beta^{(i)} M_\nu^{(i)} C_{\alpha\gamma}^{(i)} + M_\gamma^{(i)} M_\nu^{(i)} C_{\alpha\beta}^{(i)} + M_\alpha^{(i)} M_\beta^{(i)} M_\gamma^{(i)} M_\nu^{(i)} \end{aligned} \tag{21}$$

For convenience, the moments are denoted as: E_α, $E_{\alpha\beta}$, $E_{\alpha\beta\gamma}$ and $E_{\alpha\beta\gamma\nu}$. With these identities, the variance is:

$$\begin{aligned} \hat{V}_i(\boldsymbol{x_i}) = {} & F_0^2 + 2F_0[\sum_{\alpha=1}^{d}(F'_\alpha E_\alpha + \frac{1}{2}\sum_{\beta=1}^{d} F''_{\alpha\beta} E_{\alpha\beta})] \\ & + \sum_{\alpha=1}^{d}\sum_{\beta=1}^{d}[F'_\alpha F'_\beta E_{\alpha\beta} + \frac{1}{2}\sum_{\gamma=1}^{d}(F'_\beta F''_{\alpha\gamma} + F'_\alpha F''_{\beta\gamma}) E_{\alpha\beta\gamma} \\ & + \frac{1}{4}\sum_{\gamma=1}^{d}\sum_{\nu=1}^{d} F''_{\alpha\gamma} F''_{\beta\nu} E_{\alpha\beta\gamma\nu}] - \hat{L}_i(\boldsymbol{x_i})^2 \end{aligned} \tag{22}$$

Substituting Eqs. 16 and 22 into the loss function 6 and minimizing via simulated annealing method, we can obtain the optimal model parameters, which reconstructs the full system from partial experimental data.

Note that the polynomial model up to second order in the underlying variables represents a model with pair-wise interaction of biological components. The components can either enhance or suppress each others contribution to the biological function. This particular case can be considered as a representation of typical signaling network diagrams, although the interactions of the components are generally nonlinear. Pair-wise nonlinear interactions are generally not covered by the polynomial expansion.

Deep Learning Neural Network Models Based on Partial Data

Although the polynomial regression method can perform well around the mean, it is not suitable for complex models, especially in regions far from the mean. Neural networks and deep learning model have been proven effective for capturing general complex models. The basic idea is the same as polynomial regression except that the output function $\hat{F}(\boldsymbol{x})$ is approximated by an iterated function which depends on the structure of the neural network. In each layer, the node values are linearly mapped to the next layer and processed by activation function (Here we use ReLu as the activation function). (Fig. 2 (a)) We use the same loss function as Eq. 6. However, we cannot obtain analytic expressions for the conditional mean value and variance in the neural network model. Therefore, we use Monte Carlo sampling to compute these two quantities.

Our neural network has n_H layers and in the k^{th} layer, there are n_k nodes. For each hidden layer, the node values $\boldsymbol{z_k}$ are provided by the node values in the previous layer by:

$$\boldsymbol{z}_k = \boldsymbol{g}[\boldsymbol{W}_k \boldsymbol{z}_{k-1} + \boldsymbol{b}_k], \tag{23}$$

where $g(x)$ is the activation function (ReLu function), taking the form: $g(u) = max(0, u)$. The output layer node values are given by: $\boldsymbol{z}_k = \boldsymbol{W}_k \boldsymbol{z}_{k-1} + \boldsymbol{b}_k$. Therefore, the final output value will be several iterations of this linear transform and the model parameters are the coefficients $\boldsymbol{W}_k$ and $\boldsymbol{b}_k$ ($k \leq n_H + 1$). To obtain the conditional mean and variance value based on the neural network model, corresponding to each measuring set, we sample n_{sp} data from the fitted conditional distribution $\rho_i(\boldsymbol{x_{i'}}|\boldsymbol{x_i})$ when $\boldsymbol{x_i}$ is fixed. A nice property of the Gaussian model (Gaussian mixture model) is that the conditional probability density function is also Gaussian (Gaussian mixture model). For each $\boldsymbol{x}$, we can obtain the predicted value of y according to the neural network. From the samples, we can get the

conditional mean and variance values of y when $\boldsymbol{x_i}$ is fixed. Since the loss function (Eq. 6) cannot be expressed explicitly, gradient-based methods are not applicable. Therefore, we still use simulated annealing method to minimize the loss function with respect to model parameters $\boldsymbol{W}_k$ and $\boldsymbol{b}_k$.

As in the "blind men and elephant problem", each experiment generates partial knowledge of the problem. However, after combining the information fragments together, a more complete picture of the system is obtained. Similarly, With more and more facets collected, we are closer to the ground truth of the model. We also expect a difference in prediction when each measuring set has different number of variables or variable combinations (e.g., each measuring set contains only 2 or 3 variables) (Fig 2 (b)). When increasing the number of variables in facet, the prediction should become more accurate. The limit of this process is when all variables are measured and fitted simultaneously, which should give the most accurate prediction.

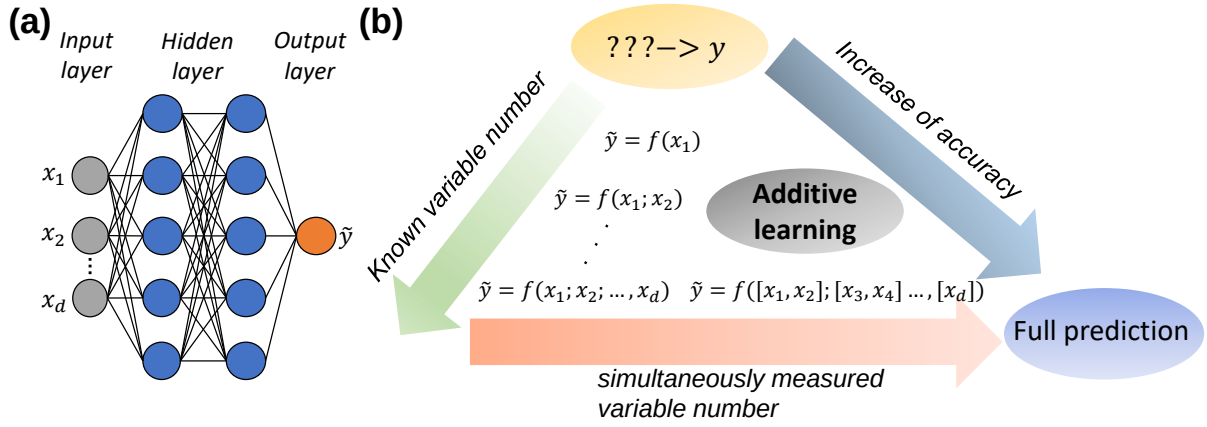

Figure 2: (a) Structure of the deep learning neural network model. (b) Illustration of the additive process in faceted learning. There are two dimensions in the "additive" notion: First, increase of known input variable number; second, increase of simultaneously measured variable number in one measurement. Both ways increase prediction accuracy.

Example 1: Spring network

As an example of a complex multi-dimensional system, we examine a networked system of springs, which can be thought of as a phenomenological example of a highly connected biological network. We implement our machine learning method on a two dimensional 8-node spring system. Therefore, system appears simple but because interactions between nodes are nonlinear, the response can be complex. Based on partial data measurements, we can reconstruct the complete force-deformation response function of this network.

Fig. 3 (a) shows the configuration of the spring network with forces exerted on all nodes. Nodes are connected by linear springs, whose stiffnesses are denoted by a 8×8 symmetric matrix $\boldsymbol{K}$ where K_{uv} is the stiffness of the spring between nodes u and v. The rest lengths are denoted by matrix $\boldsymbol{l}$ where $l_{uv} = \sqrt{|\boldsymbol{x}_u - \boldsymbol{x}_v|^2}$ is the length between nodes u and v. Nodes 1 and 5 are fixed to prevent overall translation and rotation. The spring system is subjected to random force $\boldsymbol{P}$ and has corresponding displacement matrix $\boldsymbol{\delta X}$. Both $\boldsymbol{P}$ and $\boldsymbol{\delta X}$ are 8×2 matrices, where the u^{th} row denotes the horizontal and vertical component of node u. Due to the constraints at nodes 1 and 5, the first and fifth rows of δX are fixed to be 0. We assume $\boldsymbol{P}$ is normally distributed: $P_{uv} \sim N(0, 0.02)$ and we want to predict the displacement matrix $\boldsymbol{\delta X} = \boldsymbol{h}(\boldsymbol{P})$ as a function of forces $\boldsymbol{P}$. In our calculation, the vertical displacement of node 2 (δX_{22}) is the output. The input vector is the twelve components of the forces exerted on the six free nodes, which is arranged as: $\boldsymbol{x} = (P_{21}, P_{31}, P_{41}, P_{61}, P_{71}, P_{81}, P_{22}, P_{32}, P_{42}, P_{62}, P_{72}, P_{82})$.

To implement the algorithm described above, we first generate training data with only partial information. We generate N_1 8×2 force matrices as the input of the training data and N_2 force matrices as testing data, in which every force component obeys a normal distribution: $N(0, 0.02)$. For each of the force matrix, we calculate the 8×2 deformation

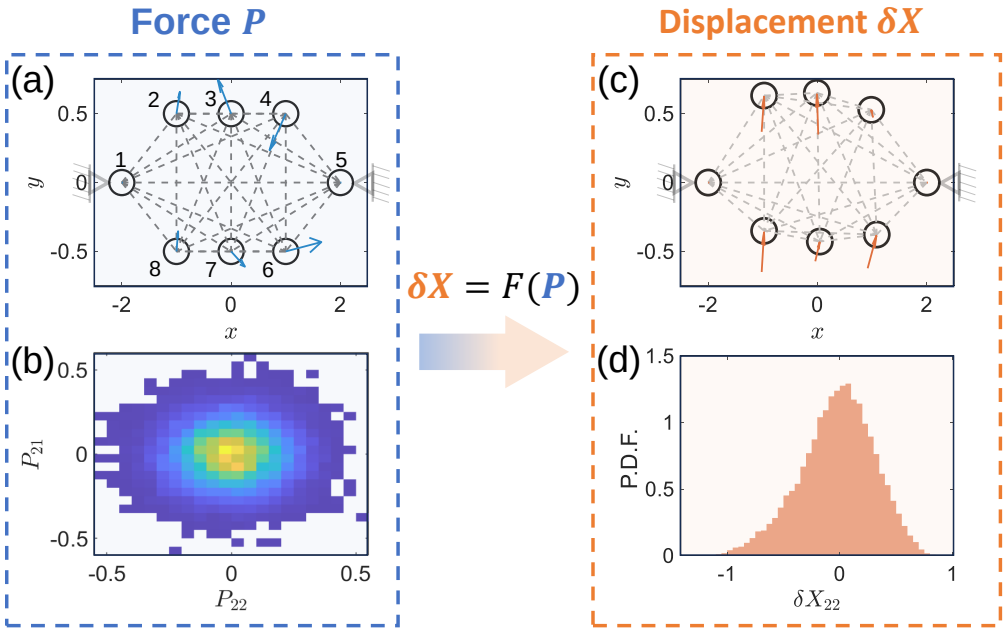

Figure 3: (a) Configuration of the 8-node spring network system. Random forces are exerted on each node, generating displacements. The applied forces follow the normal distribution $P \sim N(0, 0.02)$. Node 1 and 5 are fixed to prevent translation and rigid body rotation. The model input are forces on different nodes ($\boldsymbol{P}$) and the model output is the vertical displacement of node 2 (δX_{22}). (b) Joint probability distribution of vertical (P_{22}) and horizontal force (P_{21}) components at node 2. (c) Deformed configuration of the 8-node spring network system and the displacement of each node. (d) Probability distribution of the vertical displacement of node 2.

matrix $\boldsymbol{\delta X}$ by minimizing the total potential energy. The minimization is achieved by the gradient descent method and the initial displacements are randomly chosen, which is evenly distributed between (-0.05,0.05). The N_1 training data are evenly partitioned into 12 subgroups, which is equal to the dimension of the forces. For each subgroup i, we use one of the force components (P_i) together with the vertical displacement of node 2 (δX_{22}). We apply both the polynomial regression and neural network methods on these 12 data sets (Fig. 4). In the neural network implementation, the network has 2 hidden layers and each layer has 20 nodes. The activation function is the ReLu

function as described above. In both polynomial regression and neural network algorithms, the loss function is minimized by simulated annealing [33], where at each minimization step, all the parameters are perturbed randomly within the range of 0.05. The initial temperature T_0 is set to be 10^5 and at each step, the temperature is reduced to 95%. The minimization process is stopped when the maximum step ($i_{max} = 50000$) is reached. When approximating the conditional expectation and variance of output variable by Monte Carlo method, the sample size is set to be: $n_{sample} = 60000$.

Fig. 4 shows the predicted results of both polynomial regression (a-d) and neural network(e-h). Fig. 4(a and e) show the predicted and true δX_{22} when changing horizontal and vertical forces (P_{21} and P_{22}) applied on node 2 while other force components are zeros. For both polynomial regression and neural network approaches, the predicted surface fits well with the true surface. Fig. 4(b),(c),(f) and (g) show the predicted and true values of mean and variance of δX_{22} calculated in each bin of P_{22} (including both training and testing data). These are direct quantities that are minimized in the loss function. True and predicted displacements are evaluated for test data sets and plotted in Fig 4 (d) and (h). The scatter points are well aligned around diagonal, which implies accurate prediction.

Example 2: P53 network

In this section, we implement our algorithm on a small biological network involving expression of the senescence marker P53. The data is obtained using single cell proteomic method of [34]. We choose 8 molecules as inputs and the output is single cell expression level of P53 (Fig. 5(a)). The goal is to construct a predictive model of P53 expression as a function of 8 other single cell properties while only utilizing faceted information. Note that we measure the proteome level of 8 molecules for each single cell, therefore we have

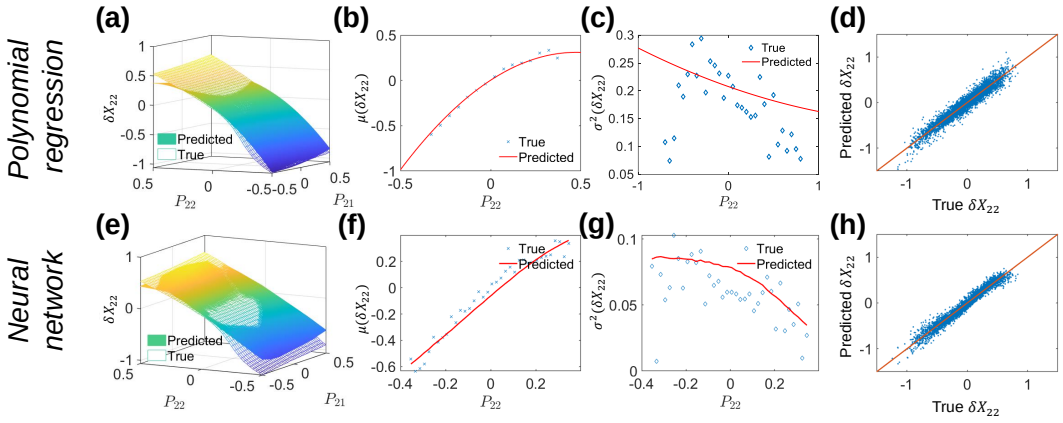

Figure 4: Polynomial and neural network model results for the spring network system. (a) Joint probability distribution of node 2 vertical displacement (δX_{22}) and node 2 vertical force component (P_{22}). (b) Projection result of mean vertical displacement dependent on vertical force on node 2. (c) variance of vertical displacement dependent on vertical force on node 2. (d) Comparison between true and predicted values of the testing data set. (e)-(h) Corresponding prediction results by neural network.

the full 8-dimensional data.

The data are obtained for cells in four conditions: control, quiescent, cells treated with 50μM Bleomycin and 250 nM Doxorubicin. The raw distributions of all variables are shown in Fig. 5(a). We standardized the data in each condition by the mean value and covariance matrix in the corresponding condition. We then remove outliers via GESD method [35]. The processed proteome expression data is bimodal, because cells are either in G1 or G2 phase of the cell cycle. For better accuracy, we use the Gaussian mixture model which consists of the sum of two Gaussian distributions, representing cells in G1 and G2, to fit the marginal distribution of each input variable. The joint distribution is approximated by the product of the 8 Gaussian mixture models (Fig. 5 (b)-(c)):

$$\hat{f}(x_1, x_2, ..., x_8) = \prod_{i=1}^{8} (\sum_{j=1}^{2} \pi_{ij} N(\mu_{ij}, \sigma_{ij})) \tag{24}$$

Similar to the spring system example, we first divide the data in each condition as

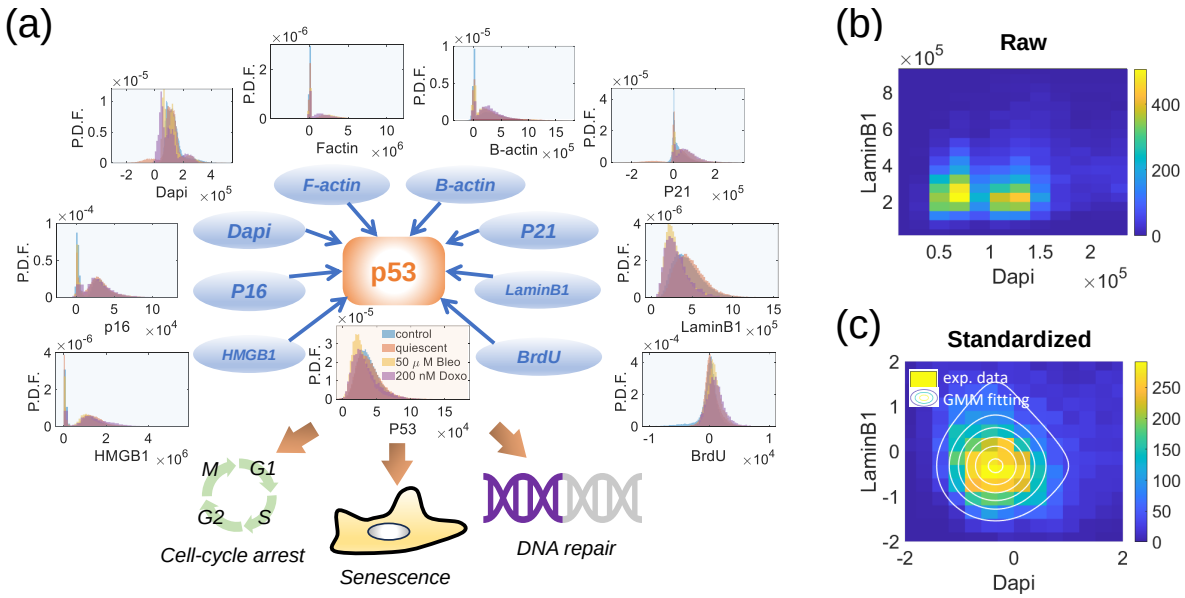

Figure 5: (a) The examined proteome of the P53 network. The input data are expressions of the 8 molecules measured in the single cell experiment and the output is the P53 expression. The probability distribution of all variables are show in 4 different cell conditions: 1. control; 2. quiescent; 3. treated with 50 μM Bleomycin; 4. treated with 200nM doxorubicin. Cells in different conditions show different proteome distributions because they have different cell cycle distributions. (b) Joint probability distribution of Dapi and LaminB1 for senescent cells treated with 50 μM Bleomycin. (c) Joint probability distribution of Dapi and LaminB1 for standardized data of senescent cells treated with 50 μM Bleomycin. The contour lines are from the Gaussian mixture model used to describe the probability distribution.

training (80%) and testing sets (20%). The training data are evenly partitioned into eight subgroups. In the i^{th} subgroup, only x_i and P53 intensity are used. In the neural network implementation, the network has 2 hidden layers and each layer has 20 nodes. The activation function is the ReLu function. In both polynomial regression and neural network algorithms, the loss function is minimized by simulated annealing methods, where at each minimization step, all the parameters are perturbed randomly within the range of ± 0.05. The initial temperature T_0 is set to be 10^5 and at each step, the temperature is

reduced to 95%. The minimization process is stopped when the maximum step ($i_{max} = 50000$) is reached. When approximating the conditional expectation and variance of output variable by monte carlo method, the sample size is set to be: $n_{sample} = 60000$.

Fig. 6 shows the predicted results for both polynomial regression (a-d) and neural network (e-h) for cells in the control condition. Fig. 6 (a)(e) show predicted P53 when Dapi and LaminB1 content change while other are fixed to zero. All data are standardized as described in previous section. Plots of mean and variance values vs. LaminB1 are shown in (Fig. 6 (b)(c)(f)(g)). True and predicted P53 content evaluated at both the testing data sets are plotted in Fig 6(d)(h). The scatter points are well aligned around $y = x$.

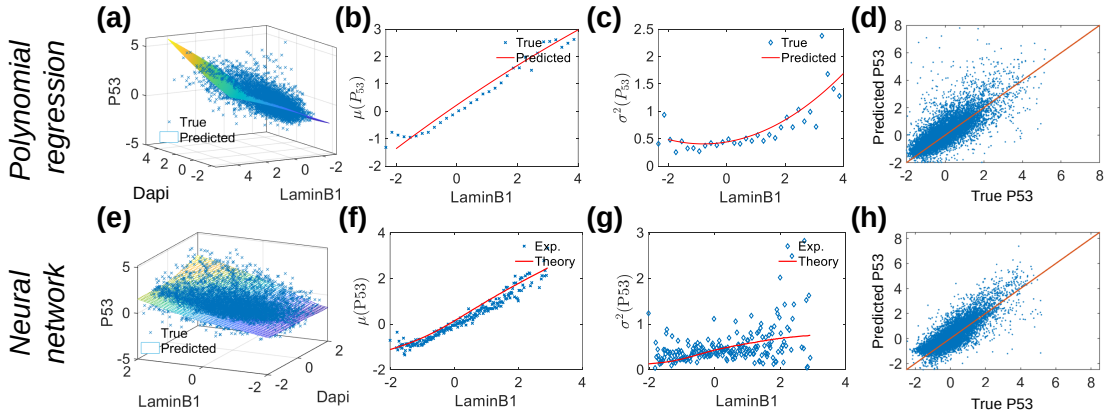

Figure 6: Polynomial and neural network model results of P53 network. (a) Surface plot of P53 content dependent on Dapi and LaminB1 content. (b) Projection result of mean value of P53 content dependent on LaminB1 content. (c) variance of P53 content dependent on LaminB1 content. (d) Comparison between true and predicted values on the testing set. (e)-(h) Corresponding prediction results by neural network.

It is also of great interest to examine our model predictions for different cell culture conditions. Quiescent and senescent cells generally have different cell cycle distributions, leading to different G1/G2 cell proportions (Fig. 5(a)). However, the mapping between the standardized input variables and P53 are the same across different cell conditions

(Fig. 7). Here we examine the model trained by data in control condition, and utilize the trained model to predict P53 content in quiescent condition (Fig. 7(b)) and senescent conditions (Fig. 7(c)&(d)). We also show the results of full neural network trained by data in the same condition. Note, in our method, the standardization procedure removes the correlation among the independent variables and the function F we learn only describes the mapping between the processed uncorrelated data, and doesn't include mutual information among the independent variables. In reality, the true function (mapping F_{true}) should combine both the intrinsic function of uncorrelated data (F) and the correlation information (Σ).

Our model also provides information on which variables contribute most to P53 content and can also illustrate the synergistic and antagonistic effects of several molecules on P53. This can be analyzed via the polynomial model. The linear coefficients F_i' mean the effect of single molecule on P53 while the quadratic coefficients F_{ij}'' represent the synergistic/antagonistic effects. For cells in the control condition, for example, LaminB1 and HMGB1 contribute most to P53 content and we can see clearly synergistic effects of HMGB1 and B-actin on P53, and antagonistic effects of HMGB1 and F-actin (Fig. 8) on P53. We can also apply the method on other variables, which finally leads to the complete network structure reconstruction with both first order (correlation) and higher order information (synergistic/antagonistic effects).

Additive property of the faceted learning

As mentioned before, the faceted learning has an additive process, during which the prediction accuracy is increased with increasing number of simultaneously used variable in one set of measurement.

To examine this, we increase the number of variables in each measuring set (e.g., from

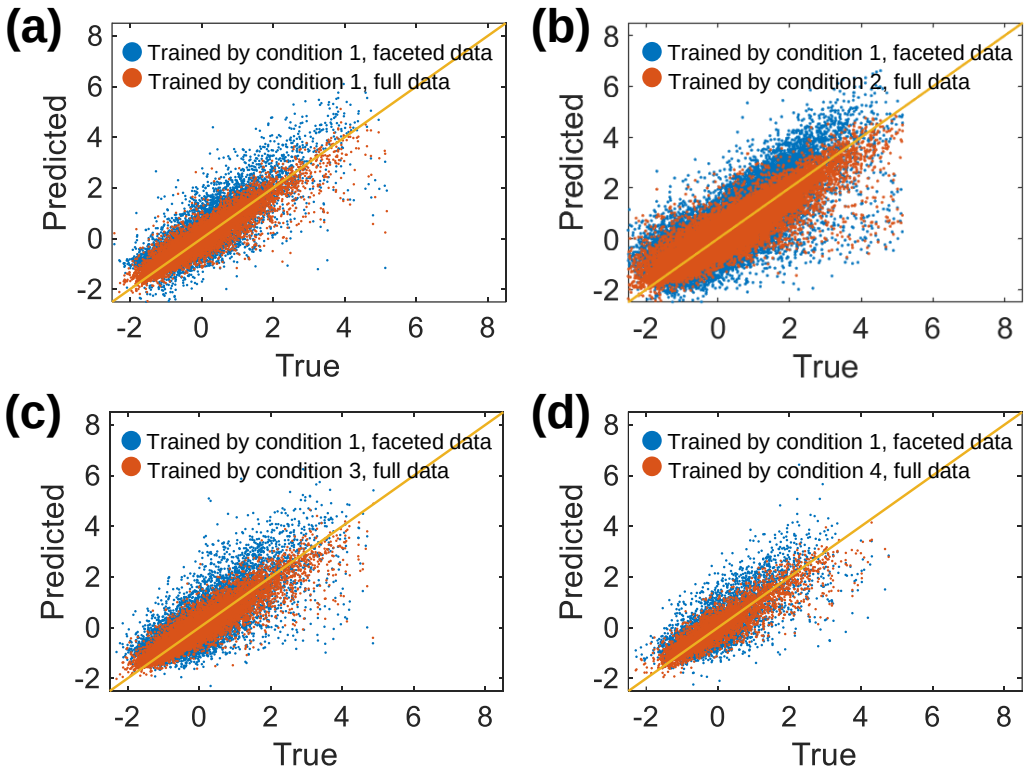

Figure 7: Testing of the model trained by data in control condition on other cell conditions. (a) Test on control condition (condition 1). (b) Test on quiescent cell data (condition 2). (c) Test on data from cells treated with 50 μM Bleomycin (condition 3). (d) Test on data from cells treated with 200 nM Doxorubycin (condition 4). In all the results, the model trained by data in control condition provides satisfactory accuracy compared to the full neural network and this means that the intrinsic mapping between standardized input and standardized P53 content remains invariant across different cell conditions. The only difference among different conditions is the probability distribution.

measuring one force component to measuring two force components simultaneously), the prediction becomes more accurate (Fig. 9). The limit of this additive process is measuring all the variables simultaneously, which is the typical regression problem.

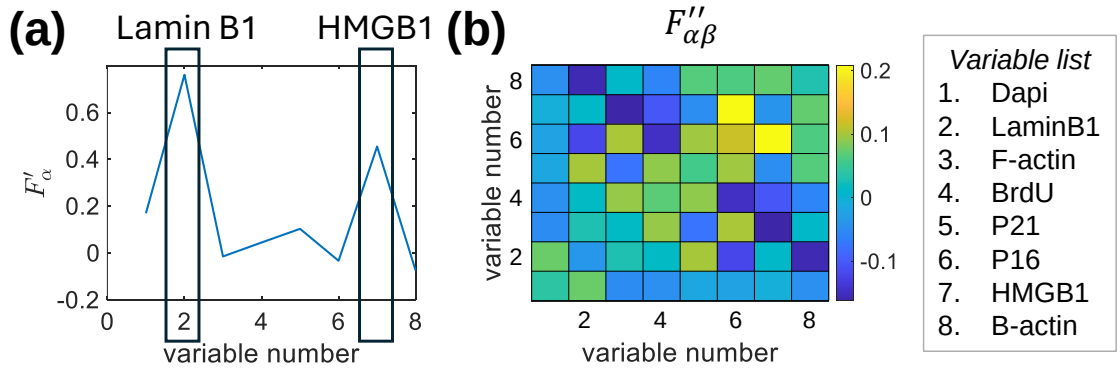

Figure 8: Linear and quadratic coefficients of the polynomial regression model of the P53 data. (a) Linear coefficients. LaminB1 and HMGB1 contribute most to P53 content. (b) Quadratic coefficients. There is obvious synergistic effects of HMGB1 and P16 and antagonistic effects of HMGB1 and F-actin on P53.

Discussion and Conclusion

In this work, we develop a method that reconstructs the complete picture of a system from partial data sets. Each data set only contains part of the input variables and the output variable. This is the essence of the Blind men and elephant problem, where each person only know partial information about the elephant. However, exchanging information among each other helps better understand the system. In general, we abstract the system information from the conditional distribution of the output variable when partial input variables are fixed. By assuming some models for the system equation, we fit the true distribution with model parameters. Both polynomial regression and neural network methods are applied and compared. For normally distributed input variable, we can well approximate the output distribution by only mean value and variance value. By minimizing the loss function that contains the mean squared errors of both mean and variance of output values, we can obtain the predictive model that reconstructs the

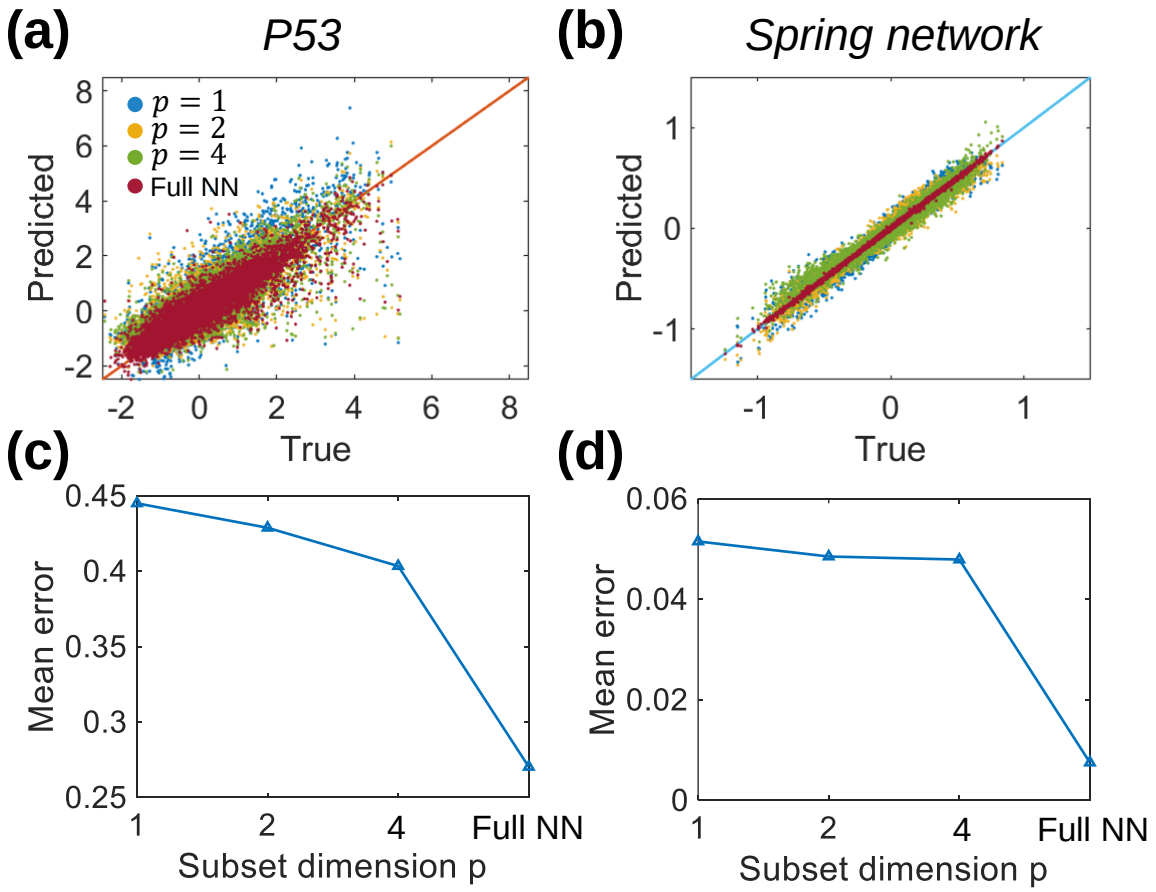

Figure 9: Additive property: Increase of simultaneously measured variable number improves the prediction accuracy. (a)&(c) P53 network data. (b)&(d) Spring network data.

complete system.

It is possible to use the toolbox developed in ML to optimize data regression. It is also possible to minimize a different set of objective functions for the ML training process. These improvements can be made depending on the specifics of the problems at hand. Other methods from network reconstruction also can be applied. One possible problem is the uniqueness of the model from faceted data. We have not explored this angle in this paper, but it is likely that multiple networks can produce the same set of data, as others

have noted [36, 37].

We implement our method on both a mechanical system (spring network) and a small biological network (P53 network). Both polynomial and neural network methods are examined. 2D and 1D projection results are compared between true data and prediction. Finally, we examine the additive property of the learning process. By increasing the number of known variables and number of simultaneously measured variables, the prediction accuracy is gradually increased.

Our proposed method can be applied to high dimensional data, including single cell proteomics data. The resulting model function $y = F(\boldsymbol{x})$ represents the genome-wide unbiased model of a particular biological function. As long as measurements can be made for the output y and underlying variable $\boldsymbol{x}_i$, the model can be systematically improved. Since real biological functions are complex emergent properties of a highly connected network, our method represents a systematic and unbiased way of reconstructing the network. Moreover, our approach also allows us to examine cells which are rare in the population of cells, and look for how these cells generate biological function. Since cell heterogeneity and entropy is increased in diseased context such as cancer [38], our approach can reveal how the network is perturbed in these diseased context. With increasing quality of single cell data sets, the predictions will be more accurate and useful. What is clear presently, however, is that there is a lack of single-cell high dimensional data or concerted efforts to obtain faceted data that connect biological function with the underlying proteome. If these data sets are available, then our procedure proposed here, combined with machine learning and AI methods, can be implemented in a straightforward manner, and truly predictive models can be obtained. New technological innovations for single cell measurements and systematic data gathering efforts are needed to achieve this next level era of quantitative biology.

References

[1] Alm E, Arkin AP. Biological networks. *Curr. Opin. Struct. Biol.* 2003; 13(2):193-202.

[2] Wuchty S, Ravasz E, Barabsi AL. The architecture of biological networks. *Complex systems science in biomedicine.* 2006:165-81.

[3] Saint-Antoine MM, Singh A. Network inference in systems biology: recent developments, challenges, and applications. *Curr Opin Biotechnol* 2020; 63:89-98.

[4] Jordan MI, Mitchell TM. Machine learning: Trends, perspectives, and prospects. *Science.* 2015; 349(6245):255-60.

[5] Zitnik M, Nguyen F, Wang B, Leskovec J, Goldenberg A, Hoffman MM. Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities. *Inf. Fusion.* 2019; 50:71-91.

[6] Mahesh B. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).* 2020; 9(1):381-6.

[7] Greener JG, Kandathil SM, Moffat L, Jones DT. A guide to machine learning for biologists. *Nat. Rev. Mol. Cell Biol.* 2022; 23(1):40-55.

[8] Roussos ET, Condeelis JS, Patsialou A. Chemotaxis in cancer. *Nat. Rev. Cancer.* 2011; 11(8):573-87.

[9] Li Y, Zhou X, Sun SX. Hydrogen, bicarbonate, and their associated exchangers in cell volume regulation. *Front. cell dev. biol.* 2021; 9:683686.

[10] Sunyer R, Trepat X. Durotaxis. *Curr. Biol.* 2020; 30(9):R383-7.

[11] Jiang H, Sun SX. Cellular pressure and volume regulation and implications for cell mechanics. *Biophys. J.* 2013; 105(3):609-19.

[12] Shim G, Breinyn IB, Martnez-Calvo A, Rao S, Cohen DJ. Bioelectric stimulation controls tissue shape and size. *Nat. Commun.* 2024; 15(1):2938.

[13] Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Favera RD, Califano A. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinforma.* 2006; 7:1-15.

[14] Wang YR, Huang H. Review on statistical methods for gene network reconstruction using expression data. *J. Theor. Biol.* 2014; 362:53-61.

[15] Wong AH, Gottesman II, Petronis A. Phenotypic differences in genetically identical organisms: the epigenetic perspective. *Hum. Mol. Genet.* 2005; 14:R11-8.

[16] Usaj MM, Yeung CH, Friesen H, Boone C, Andrews BJ. Single-cell image analysis to explore cell-to-cell heterogeneity in isogenic populations. *Cell Syst.* 2021; 12(6):608-21.

[17] Hall SE, Beverly M, Russ C, Nusbaum C, Sengupta P. A cellular memory of developmental history generates phenotypic diversity in C. elegans. *Curr. Biol.* 2010; 20(2):149-55.

[18] Peaston AE, Whitelaw E. Epigenetics and phenotypic variation in mammals. *Mamm. Genome* 2006; 17:365-74.

[19] Everson R, Sirovich L. KarhunenLoeve procedure for gappy data. *JOSA A.* 1995 Aug 1;12(8):1657-64.

[20] Koronaki ED, Evangelou N, Psarellis YM, Boudouvis AG, Kevrekidis IG. From partial data to out-of-sample parameter and observation estimation with diffusion maps and geometric harmonics. *Comput. Chem. Eng.* 2023; 178:108357.

[21] Luecken MD, Theis FJ. Current best practices in single cell RNAseq analysis: a tutorial. *Mol. Syst. Biol.* 2019; 15(6):e8746.

[22] Shaffer SM, Emert BL, Hueros RA, Cote C, Harmange G, Schaff DL, Sizemore AE, Gupte R, Torre E, Singh A, Bassett DS. Memory sequencing reveals heritable single-cell gene expression programs associated with distinct cellular behaviors. *Cell.* 2020; 182(4):947-59.

[23] Lu Y, Chen JJ, Mu L, Xue Q, Wu Y, Wu PH, Li J, Vortmeyer AO, Miller-Jensen K, Wirtz D, Fan R. High-throughput secretomic analysis of single cells to assess functional cellular heterogeneity. *Anal. Chem.* 2013 Feb 19;85(4):2548-56.

[24] Wu PH, Gilkes DM, Phillip JM, Narkar A, Cheng TW, Marchand J, Lee MH, Li R, Wirtz D. Single-cell morphology encodes metastatic potential. *Sci. Adv.* 2020; 6(4):eaaw6938.

[25] Chambliss AB, Wu PH, Chen WC, Sun SX, Wirtz D. Simultaneously defining cell phenotypes, cell cycle, and chromatin modifications at single-cell resolution. *FASEB J.* 2013; 27(7):2667.

[26] McManus J, Cheng Z, Vogel C. Next-generation analysis of gene expression regulationcomparing the roles of synthesis and degradation. *Molecular BioSystems.* 2015; 11(10):2680-9.

[27] Liu Y, Beyer A, Aebersold R. On the dependency of cellular protein levels on mRNA abundance. *Cell.* 2016; 165(3):535-50.

[28] Wu Y, Pegoraro AF, Weitz DA, Janmey P, Sun SX. The correlation between cell and nucleus size is explained by an eukaryotic cell growth model. *PLoS Comput. Biol.* 2022; 18(2):e1009400.

[29] Izenman AJ. Introduction to manifold learning. *Wiley Interdisciplinary Reviews: Computational Statistics.* 2012; 4(5):439-46.

[30] Ma Y, Fu Y. Manifold learning theory and applications. *Boca Raton: CRC press;* 2012.

[31] Reynolds DA. Gaussian mixture models. *Encyclopedia of biometrics.* 2009; 741(659-663).

[32] Rasmussen C. The infinite Gaussian mixture model. *Advances in neural information processing systems.* 1999;12.

[33] Bertsimas D, Tsitsiklis J. Simulated annealing. *Stat. Sci.* 1993; 8(1):10-5.

[34] Lin JR, Fallahi-Sichani M, Chen JY, Sorger PK. Cyclic immunofluorescence (CycIF), a highly multiplexed method for single-cell imaging. *Current protocols in chemical biology.* 2016; 8(4):251-64.

[35] Rosner B. Percentage points for a generalized ESD many-outlier procedure. *Technometrics.* 1983; 25(2):165-72.

[36] Angulo MT, Moreno JA, Lippner G, Barabsi AL, Liu YY. Fundamental limitations of network reconstruction from temporal data. *J. R. Soc. Interface.* 2017; 14(127):20160966.

[37] Henderson J, Michailidis G. Network reconstruction using nonparametric additive ODE models. *PloS one.* 2014; 9(4):e94003.

[38] Feinberg AP, Levchenko A. Epigenetics as a mediator of plasticity in cancer. *Science.* 2023; 379(6632):eaaw3835.