Data-Guided Controllability: Learning from the Human Genome

G. Patterson¹, A. Surana², M. Mesbahi³, S. Smale⁴, A. Bloch⁵, I. Rajapakse¹,⁵

¹Department of Computational Medicine & Bioinformatics, Medical School, University of Michigan
²United Technologies Research Center
³Department of Aeronautics and Astronautics at the University of Washington
⁴Department of Mathematics, City University of Hong Kong and University of California, Berkeley
⁵Department of Mathematics, University of Michigan

Abstract

We develop a framework for the control of dynamical systems that we know very little about. From limited, high-dimensional data observations, we approximate the natural dynamics of the system and investigate the controllability of the identified equations. The linear discrete-time-invariant model identification is done using dynamic mode decomposition (DMD), which is motivated by Koopman operator theory and can capture, from data, underlying nonlinear coherent structures. Viewing the identified linear model as a network, we propose three “control evaluations” to recommend intelligent control placements for steering the system to a specific target. The motivating application for this work is the control of genetic networks, in the context of cell reprogramming. We show that our targeted control evaluations identify known cell reprogramming strategies better than existing, general control metrics. Our framework is a first step toward using mathematics, guided by data, to control the genome.

When the governing equations of a dynamical system are known or can be well approximated, there are effective methods to study the controllability of that system. However, data limitations in complex dynamical systems that are difficult or resource-intensive to observe make it challenging to generate adequate governing equations. Biological systems in particular have limited data, thus it has been difficult to acquire insight into controllability.

The motivating application and focus for this work is the control of genetic networks, specifically in the context of cell reprogramming. While it is generally accepted that the human genome is a dynamical system [1, 2], the dynamic properties of the genome are little understood, despite increasingly advanced technologies and assays (e.g. RNA-seq, ChIP-seq, Hi-C). Nevertheless, Weintraub et al. demonstrated control over that system with the reprogramming of fibroblasts (FIB) into muscle cells via the introduction of a single transcription factor, MyoD [3]. In addition, Yamanaka et al. (2007) successfully reprogrammed a human FIB into an induced pluripotent stem cell (iPSC). With information regarding merely the initial state (FIB) and the desired final state (embryonic stem cell, ESC), Yamanaka and colleagues predicted and determined experimentally that the system can be driven from one cell type to another using just the four transcription factors OCT4, SOX2, KLF4, and MYC (all four abbreviated OSKM).

In the same spirit, our aim is to formalize and validate mathematically the findings of Yamanaka. Using function (RNA-seq) and form (Hi-C) data, as well as binding location data for 222 transcription factors (TFs), our goal is to recommend mathematically which TFs are the best candidates to reprogram FIB into another cell type. More generally, we present methodology for making inferences about the controllability of a dynamical system with respect to a specific target, and when the precise dynamic equations are unknown and only minimal data is available. We present the proposed framework in a general context, followed by the application to controlling genetic networks.

We show that ...

1 General Framework

Assume a complex dynamical system is settled into a basin of attraction, that is incompletely described by $M$ data snapshots $x_i \in \mathbb{R}^N$, $i = 1, \cdots, M$, which are collected into the matrix $X_0$.

For simplicity we will assume that the snapshots have been uniformly sampled every $\Delta t$ time units; however, the methodology can be adapted if this was not the case. The state dimension

1
Figure 1: Two- and three-dimensional (left and right, resp.) depictions of an energy landscape for a hypothetical dynamical system. In black, hypothetical observations of the system at different times. Our methods are based on the assumption that data \( \mathbf{X}_0 \) is available at several time points describing some initial conditions or basin of attraction, and that at least one data point \( \mathbf{x}_1 \) is available describing some desired terminal condition or basin of attraction.

\( N \) is assumed to be high and the number of snapshots \( M \) to be low; thus \( 1 < M \ll N \). Our goal is to steer the system to another basin of attraction, represented by the single data snapshot \( \mathbf{x}_1 \),

\[
\mathbf{X}_0 = \begin{pmatrix} \mathbf{x}_0^1 & \cdots & \mathbf{x}_0^M \end{pmatrix} \in \mathbb{R}^{N \times M}, \quad \mathbf{x}_1 \in \mathbb{R}^N
\]  

A low-dimensional depiction of the basins of attraction and the small sample of data points representing our system is given in Figure 1.

We develop a framework for identifying good candidates for control to a specific target, based on very limited data. The problem, which we dub data-guided control, consists of two parts: (i) model identification and (ii) control thereof. Separately, both questions have been addressed in depth, for both linear and nonlinear dynamical systems. As a first approach, in this paper we identify and analyze linear discrete-time-invariant dynamical systems of the form

\[
\mathbf{x}[k+1] = A \mathbf{x}[k] + B \mathbf{u}[k]
\]

Our data snapshots \( \mathbf{X}_0 \) and \( \mathbf{x}_1 \) correspond to the \( N \)-dimensional state variable \( \mathbf{x} = [x_1, \ldots, x_N]^T \in \mathbb{R}^N \), and the unknown constant matrix \( A \in \mathbb{R}^{N \times N} \) encodes interactions between the state variables over time, so that entry \( A_{ij} \) in the \( i \)th row and \( j \)th column describes how \( x_j \) contributes to the dynamics of \( x_i \). We will think of the state variables \( x_i \) as nodes in a network, and so \( A \) describes directed and weighted edges between nodes.

The control function \( \mathbf{u} = [u_1, \ldots, u_N]^T \in \mathbb{R}^{N_d} \) can perturb the dynamics, where \( N_d \leq N \) is the dimension of the control signal. The unknown constant matrix \( B \in \mathbb{R}^{N \times N_d} \) encodes where the control terms influence the network, so that entry \( B_{ij} \) describes how \( u_j \) contributes to the dynamics of \( x_i \).

For (i), a variety of model identification methods exist in literature, each having its own pros/cons and being especially useful for certain classes (e.g. linear, hybrid, nonlinear, etc.) of systems [4, 5, 6]. In particular for linear systems, several approaches exist including prediction error methods; expectation maximization based iterative approaches, and subspace methods [4]. In high dimensional problems one cannot in practice apply such standard identification algorithms directly, and a dimensionality reduction step is a must. In this direction several techniques such as Balanced Proper Orthogonal Decomposition (BPOD) [7], Eigensystem Realization Algorithm

2
(ERA) [8], Observer Kalman Filter Identification [9], and more recently Dynamic Mode Decomposition (DMD) and its variants [10, 11] have been developed. We intend to use DMD in our application for several reasons: first, DMD can be applied to arbitrary time histories compared to impulse response data for BPOD/ERA type approaches, see [11] for a detailed discussion. Secondly, substantial success has been demonstrated [12] in applying DMD to high dimensional problems such as in fluid dynamics which have been otherwise difficult to analyze and develop controllers. Finally DMD has a strong connection with Koopman operator theory [12, 13, 14] and has proven useful for systems with nonlinear dynamics [15] (further discussed later in the paper).

Relevant to cell reprogramming, several network identification methods have been proposed specific to genetic regulatory networks. In [16] extensive TF interaction networks are constructed for 475 TFs in 41 different cell types. From a dynamical systems point of view, the work reported in [16] is capable of providing the zero/non-zero pattern of $A$ but does not assign weights to edges. In [17] the authors present a scalable, multiple linear regression based methodology to identify a linear first-order model of regulatory interactions, using data from systematic transcriptional perturbations. In [18] the authors evaluate the performance of several “reverse-engineering” methods for gene network identification, including those based on clustering algorithms, Bayesian networks, information-theoretic approaches, and ordinary differential equations. As previously mentioned, we choose DMD for its scalability and connections to nonlinear systems. Moreover, to our knowledge DMD has not been previously used in the identification of genetic networks. In addition, we propose an alternative identification method which specifically accounts for TF interactions, similar to [16]. The details of both identification methods are given in Section 2.

As for (ii), the control of dynamical systems is a classic problem. Especially in the linear setting, analytic results exist that effectively characterize the controllability properties of a given system [19, 20]. The field of network control has brought renewed interest to linear systems theory in the context of large, complex networks, including applications to genetic networks [2, 21, 22, 23]. Questions that have received particular attention recently investigate which subsets of the state variables need to be directly controlled in order to gain complete control of the network [24, 25, 26], part of the network [27], or in order to maximize some controllability or observability metric [28, 29, 30, 31]. These questions are related to the design of the $B$ matrix in (2), which defines exactly how and, more importantly, where the controls directly influence the network.

The control problem we address in this work is similar, with an important distinction that we have a specific target $x_1$ in mind. For a choice of $B$, we propose three control evaluations which, based on existing controllability results and metrics, assess the capacity of the corresponding system (2) to be driven from the initial state $x_0$ to the target state $x_1$. Another key difference in our work from existing research is that in most linear and network control references the matrices $A$ and sometimes $B$ are assumed to be known, with the notable exception [11]. For us, the state matrix $A$ will come from the identified (uncontrolled) model, and the choice of the input matrix $B$ is what we hope to evaluate. For the genetic application, the choices for the input matrix $B$ will come from the TF binding data. This coupling of model identification with control techniques is, to our knowledge, new, as well as the target-specific controllability evaluations.

The remainder of this work is organized as follows: Sections 2 and 3 present the general framework for model identification and the control evaluations, respectively. Section 4 applies the framework to the cellular reprogramming, and our findings are discussed in Section 5.

## 2 The “A” Matrix

The uncontrolled dynamics ($u \equiv 0$ in (2)) are given by

$$x[k+1] = Ax[k]$$

where $A \in \mathbb{R}^{N \times N}$ is an unknown constant matrix. About the initial basin of attraction, the data snapshots $X_0$ can be used to approximate $A$ using a technique known as dynamic mode
decomposition, presented next. In section 2.3 we also present an alternative computation for the identification of $A$ with imposed structural constraints, specifically catered to genetic networks.

2.1 Dynamic Mode Decomposition

We assume the data points $X_0$ can be described by the discrete-time linear difference equation (3). Letting $X = X_0(:, 1:M - 1)$ and $Y = X_0(:, 2:M)$ (MATLAB notation: the first and last $M - 1$ columns of $X_0$, respectively), we can rewrite (3) using all data snapshots as

$$Y = AX,$$

which implies

$$A \approx \tilde{A} := YX^\dagger$$

where $\dagger$ represents the Moore-Penrose pseudoinverse. It can be shown $\tilde{A}$ is the best approximation of $A$ (with respect to the Frobenius norm), with exact equality when $X$ and $Y$ are linearly consistent (that is, the null space of $Y$ contains the null space of $X$) [10]. The pseudoinverse is easily calculated, $X^\dagger = V\Sigma^{-1}U^T$, using the singular value decomposition (SVD) of $X = UV\Sigma$, where $U$ and $V$ are orthogonal matrices, $\Sigma$ is a diagonal matrix, and $^T$ denotes the complex conjugate transpose.

Dynamic mode decomposition (DMD) computes the eigendecomposition of the linear operator $\tilde{A}$, which allows for a lower dimension representation and analysis of the dynamics. The columns of the matrix $\Phi \in \mathbb{C}^{N \times (M-1)}$ are the so-called dynamic modes: the eigenvectors of $\tilde{A}$, with corresponding eigenvalues along the diagonal of the matrix $A \in \mathbb{C}^{(M-1) \times (M-1)}$. That is,

$$A\Phi = \Phi \Lambda.$$ 

For high-dimensional $\tilde{A}$, the direct computation of $(\Phi, \Lambda)$ can be expensive, but an alternative algorithm provides efficiency:

Algorithm 1 (Exact DMD, J. Tu, 2013, [10])

1. Compute the SVD of $X = UV\Sigma^T$ and let $\tilde{A} := YV\Sigma^{-1}U^T$.

2. Let $\hat{A} := U^T\tilde{A}U$.

3. Compute the eigendecomposition of the much lower dimensional matrix $\hat{A} \in \mathbb{R}^{(M-1) \times (M-1)}$, writing:

$$\hat{A}w_i = \lambda_i w_i$$

Each nonzero eigenvalue $\lambda_i$ is also an eigenvalue of $\tilde{A}$.

4. The DMD mode $\phi_i$ (and eigenvector of $\tilde{A}$) corresponding to $\lambda_i$ is then given by

$$\phi_i = \frac{1}{\lambda_i}YV\Sigma^{-1}w_i$$

Each DMD mode represents some oscillatory and/or exponential coherent structure underlying the full dynamics. Since we are working in discrete time, the complex DMD eigenvalues with modulus less than one correspond to stable DMD modes, and similarly eigenvalues with modulus greater than one correspond to unstable modes. The periods and frequencies of each oscillatory mode ($\text{Im}(\log(\lambda_i)) \neq 0$) are computed by

$$\text{period}_i = \frac{2\pi \Delta t}{\text{Im}(\log(\lambda_i))}, \quad \text{frequency}_i = \frac{1}{\text{period}_i}$$

where $\Delta t$ is the sampling period. It is common practice to identify DMD modes of interest according to their norms, under the assumption that modes with larger norms are dynamically more important. The modes, being eigenvectors, allow for arbitrary scaling and so a variety of scaling methods have been developed, which are summarized in [10]. Our modes are scaled in Step 4 of Algorithm 1 according to the “exact DMD” method [10].
2.2 DMD and Koopman Operator

The DMD modes have been shown, under certain assumptions, to capture nonlinear properties related to the Koopman operator: a linear but infinite dimensional operator that governs the time evolution of observables or outputs defined on the state space of a (nonlinear) dynamical system [13, 14, 15]. The Koopman operator being linear (albeit infinite-dimensional) admits eigenvalues and eigenfunctions, and enables one to express the time evolution of system outputs as a linear superposition of Koopman modes. When the snapshot data is linearly consistent and the so-called Koopman eigenfunctions are in the span of the system observables, it can be shown that the Koopman eigenvalues and modes are equivalent to their DMD analog [10]. Included in the Appendix is a more in-depth description of the Koopman operator and its relationship to DMD.

DMD provides an efficient and meaningful model identification strategy that has connections to the underlying nonlinear complex dynamics.

2.3 Including Structural Constraints

When additional physical constraints are known they can be used to impose structure on the A matrix, e.g. the lack of a transcriptional interaction from gene $x_j$ to gene $x_i$ could imply $A_{ij} = 0$. Let $\mathcal{I}_A$ denote the set of index pairs $(i, j)$ that represent known non-interactions:

$$\mathcal{I}_A := \{(i, j); \text{node } x_i \text{ can not be influenced by node } x_j\}$$  \hspace{1cm} (8)

That is, $\mathcal{I}_A$ imposes a structural pattern on the A matrix, with $A_{ij} = 0$ for all $(i, j) \in \mathcal{I}_A$. The following optimization problem now summarizes the identification of the best fit linear operator satisfying the structural constraints:

$$\min_A \|Y - AX\|_F \text{ subject to } A_{ij} = 0, \text{ for all } (i, j) \in \mathcal{I}_A$$  \hspace{1cm} (9)

where $\| \cdot \|_F$ denotes the Frobenius norm. As mentioned above, DMD solves (9) when there are no structural constraints ($\mathcal{I}_A = \emptyset$). Otherwise, when $\mathcal{I}_A \neq \emptyset$, there is not necessarily an analytic solution and we must turn to numeric methods. The optimization problem (9) can be solved using the CVX package for MATLAB [32, 33] in just a few lines of code:

```matlab
cvx_begin
variable A(N,N)
minimize( norm(Y - A*X,'fro') )
subject to
    A(sub2ind([N,N],idx_i,idx_j)) == 0;

cvx_end
```

where in MATLAB the command `sub2ind([N,N],idx_i,idx_j)` appropriately indexes all $(i, j) \in \mathcal{I}_A$. Note that in this case we do not obtain modes or eigenvalues so we are somewhat more limited in the types of analysis we can do. Nevertheless, for the forthcoming controllability evaluations just having a definition of the matrix A will suffice.

3 The “B” Matrix

The heart of our framework is in the definition of the input matrix B and the controllability analysis of the corresponding influenced dynamic system. Given $(A, x_0, x_1)$, we seek to evaluate which choices of the input matrix B enable efficient steering of (2) from $x_0$ to $x_1$. Indeed, with the state matrix A computed above, the choice of B completes the dynamic model and allows for initial controllability analysis. Our data-guided control recommendations will therefore be based directly on which choices of the input matrix B yield the most desirable control properties.
with respect to a choice of $A$, $x_1$ (the target data), and $x_0$ (one of the initial data snapshots $x_0^j$).

For any $B$ matrix, we propose three controllability evaluations $\mu_i(B)$, $i = 1, 2, 3$, which are intended to quantify the capacity of the system (2) to be driven from $x_0$ to $x_1$. We refer to the three evaluations as

- $\mu_1(B; A, x_0, x_1)$ - “minimum-distance evaluation”
- $\mu_2(B; A, x_0, x_1)$ - “constant-control minimum-distance evaluation”
- $\mu_3(B; A, x_0, x_1)$ - “minimum-energy evaluation”

For comparison, we’ll also compute a well-known control metric, which we denote $\mu_0(B; A)$, that does not take into account the initial or target points but rather measures the overall average controllability of the system (2):

- $\mu_0(B; A)$ - “average controllability”

When the choice of $A$, $x_0$, and $x_1$ are clear from context, we will drop them from the notation and instead write simply $\mu_0(B)$, $\mu_1(B)$, $\mu_2(B)$, $\mu_3(B)$.

Since the input matrix $B$ encodes where the external signal can influence the network, the controllability evaluations will be used to compare different placements of controls and subsequently identify those which yield the most desirable controllability properties. We will specifically consider $B$ matrices of the form

$$B = \left[ \begin{array}{cccc} e_{i_1} & e_{i_2} & \cdots & e_{i_{N_d}} \end{array} \right]$$  \hspace{1cm} (10)$$

where $e_i$ is the $i$th column of the $N \times N$ identity matrix, for each $i \in \{i_1 < i_2 < \cdots < i_{N_d}\}$, for some $N_d \leq N$. In words, we assume we have $N_d$ control terms that each directly influence one node in the network. Certainly the reality is that $u_i$ could directly affect multiple nodes (and we do note that the controllability evaluations can be computed for any general $B$); however, this would require us to presume the relative influence of $u_i$ on each node it affects. For example, if we set $B = \left[ \begin{array}{c} e_1 + e_2 \end{array} \right]$ we are assuming $u_1$ influences nodes 1 and 2 the same amount. By instead considering $B$ of the form (10) allows for clean analysis of where to control the network.

Each of the control evaluations $\mu_i$ are based on standard linear systems results which we recall briefly in section 3.1, followed by the explicit definition of each evaluation in sections 3.2-3.5.

### 3.1 Controllability Gramian

An explicit solution to system (2) is given by

$$x[K] = A^K x_0 + \sum_{k=0}^{K-1} A^{K-1-k} Bu[k]$$  \hspace{1cm} (11)$$

where $x[0] = x_0$ is the initial state (for us, $x_0$ will be any one of the $M$ initial snapshots $x_0^j$).

The choice of control $u[k]$ for each time $k = 0, \cdots, K-1$ determines $x[K]$ for any $K > 0$. The following theorem characterizes for which states $x_1$ there exists a control $u[\cdot]$ which steers the system from $x_0$ to $x_1$:

**Theorem 1** ([19]) There exists a control $u[\cdot]$ which drives the state of system (2) from the value $x_0$ at $k = 0$ to the value $x_1$ at $k = K > 0$ if and only if

$$x_1 - A^K x_0 \in \text{Range}(W_c[K])$$  \hspace{1cm} (12)$$

where $W_c[\cdot]$ is the controllability gramian given by

$$W_c[K] = \sum_{k=0}^{K-1} A^k BB^T (A^k)^T$$  \hspace{1cm} (13)$$
An important remark is that when
\[
\text{rank}(W_c[K]) = N
\] (14)
then \(\text{Range}(W_c[K])\) is the entire state space, and thus (12) is satisfied trivially and a control exists to drive the system from any \(x_0\) to any \(x_1\). In this case the system is said to be completely controllable. If condition (14) is not satisfied, the system is not completely controllable, but it is possible that condition (12) is still satisfied. To check, we simply test if
\[
\text{rank}(W_c[K]) = \text{rank}(\left[ W_c[K], x_1 - A^K x_0 \right])
\] (15)
Once it has been established that a control \(u[\cdot]\) exists that drives the system from \(x_0\) to \(x_1\), the energy-minimal control \(u[\cdot]\) exists and is given by the following theorem:

**Theorem 2** If \(u[\cdot]\) is any control of the form
\[
u_{\ast}[k] = -B^T (A^T)^{K-1-k} \xi
\]
where \(\xi\) satisfies
\[
W_c[K] \xi = A^K x_0 - x_1
\]
then the control \(u_{\ast}[\cdot]\) drives the system from \(x_0\) at \(k = 0\) to \(x_1\) at \(k = K > 0\), and if \(u_1[\cdot]\) is any other control which drives the system from \(x_0\) at \(k = 0\) to \(x_1\) and \(k = K\) then
\[
\sum_{k=0}^{K-1} u_1^T[k] u_1[k] \geq \sum_{k=0}^{K-1} u_{\ast}^T[k] u_{\ast}[k]
\]
We say that \(u_{\ast}[\cdot]\) is the energy-minimal control which takes the system from \(x_0\) to \(x_1\) in \(K > 0\) time steps. Moreover, the minimal energy \(e_{\ast}\) is given by
\[
e_{\ast} = \sum_{k=0}^{K-1} u_{\ast}^T[k] u_{\ast}[k] = \xi^T (A^K x_0 - x_1)
\] (16)

The three evaluations \(\mu_1, \mu_2, \mu_3\) will be based on Equations (13), (11), and (16), respectively.

### 3.2 Average Controllability

The first evaluation \(\mu_0\) is given by
\[
\mu_0(B) := \text{tr}(W_c[K, B]).
\] (17)
Here \(W_c[K, B]\) denotes the explicit dependence of the controllability gramian \(W_c\) on \(B\). Different fixed values for the final time \(K\) can be considered. The trace of the gramian is a well-known control metric that measures the “average controllability” of a linear control system [30]. The rank condition (14) provides a “yes-no” answer to the question of complete controllability, but other metrics (e.g. \(\text{tr}(W_c)\), \(\text{tr}(W_c^{-1})\), \(\log(\det(W_c))\) [30]) have been shown to more precisely answer how controllable a system is [30]. One advantage of computing \(\text{tr}(W_c)\) compared to the other established metrics is that Equation (14) does not need to be satisfied, allowing for analysis on a larger set of possible \(B\) matrices.

Under the assumptions (10), computations can be expedited to quickly compute \(\mu_0\) for a
variety of $B$:

$$
\mu_0(B) = \text{tr}\left( \sum_{k=0}^{K-1} A^k B B^T (A^k)^T \right) \\
= \text{tr}\left( \sum_{k=0}^{K-1} B B^T (A^k)^T A^k \right) \\
= \text{tr}\left( B B^T \sum_{k=0}^{K-1} (A^k)^T A^k \right) \\
= \sum_{j=1}^{N_d} \left( \sum_{k=0}^{K-1} (A^k)^T A^k \right)_{i,j,i,j}
$$

That is, $\sum_{k=0}^{K-1} (A^k)^T A^k$ can be computed once, and the diagonal entries of the resulting matrix are summed according to each $B$. A higher value of $\mu_0$ indicates more controllability.

An important remark is that, unlike the forthcoming evaluations, $\mu_0$ does not depend on the target $x_1$ or initial condition $x_0$.

### 3.3 Minimum-Distance Evaluation

The first target-dependent evaluation $\mu_1(B)$ we propose measures how close we can get to the target for a given control placement:

$$
\mu_1(B) := \min_u \left\| x_1 - \left( A^K x_0 + \sum_{k=0}^{K-1} A^{K-1-k} B u[k] \right) \right\|_2 \\
\text{When (12) is satisfied, we know } \mu_1(B) = 0 \text{ since we can get exactly to the target, which will not be the case for most of the } B \text{ matrices we consider in the genetic application. An analytic solution to the minimization problem (19) exists based on the pseudoinverse: From (11) and using the endpoint condition } x[K] = x_1 \text{ for some fixed time } K > 0, \text{ we have}
$$

$$
x_1 = A^K x_0 + \sum_{k=0}^{K-1} A^{K-1-k} B u[k] \\
x_1 - A^K x_0 = \sum_{k=0}^{K-1} A^{K-1-k} B u[k]
$$

Letting $C := [A^{K-1} B, A^{K-2} B, \ldots, A B, B]$ and $\bar{u} := [u[0]^T, u[1]^T, \ldots, u[K-2]^T, u[K-1]^T]^T$, we have

$$
x_1 - A^K x_0 = C \bar{u} \\
C^\dagger (x_1 - A^K x_0) = \bar{u}
$$

So, $\mu_1(B)$ is rewritten:

$$
\mu_1(B) := \min_u \left\| x_1 - A^K x_0 - C C^\dagger (x_1 - A^K x_0) \right\|_2 \\
\text{(19)}
$$

### 3.4 Constant-Control Minimum-Distance Evaluation

The second proposed control evaluation $\mu_2$ measures how close to $x_1$ we can steer the system using a constant control $u[k] \equiv \bar{u}$. For FIB to ESC reprogramming, this is consistent with the constant flooding of the four OSKM factors. Under this assumption, (2) becomes

$$
x[k+1] = A x[k] + B u. \\
\text{(20)}
$$
Then, from (11) and using the endpoint condition $x[K] = x_1$ for some time $K > 0$, we have

$$x_1 = A^K x_0 + \sum_{k=0}^{K-1} A^k B \hat{u}$$

$$x_1 - A^K x_0 = \left( \sum_{k=0}^{K-1} A^k \right) B \hat{u}$$

In general a $\hat{u}$ that satisfies the above equality may not exist, but it is known that the Moore-Penrose pseudoinverse solves the least-squares problem:

$$\min_{\hat{u}} \| x_1 - A^K x_0 - \left( \sum_{k=0}^{K-1} A^k \right) B \hat{u} \|_2$$

(21)

where the solution is given by

$$\hat{u}_* := \left( \sum_{k=0}^{K-1} A^k \right)^\dagger \left( x_1 - A^K x_0 \right)$$

(22)

Define $\mu_2(B)$ as the corresponding minimal distance:

$$\mu_2(B) := \| x_1 - A^K x_0 - \left( \sum_{k=0}^{K-1} A^k \right) B \hat{u}_* \|_2$$

(23)

This evaluation clearly depends on the initial and target data, $x_0$ and $x_1$, and utilizes a simple constant control regime.

### 3.5 Minimum-Energy Evaluation

When it exists, the energy-minimal control from $x_0$ to $x_1$ in $K$ time units is given in Equation (16) of Theorem 2, and is dependent on $W_c$ and therefore on $B$. So, for our final controllability evaluation, we define $\mu_3$ by the minimal energy:

$$\mu_3(B) := e_*(B) = e_*(x_0, x_1, W_c[B])$$

(24)

with $\mu_3(B)$ undefined when no such control exists.

### 4 Application to Cell Reprogramming

Indika: Can you write a transition to the biology here?

Following the methodology of sections 2 and 3, we compute $\mu_0$, $\mu_1$, $\mu_2$, and $\mu_3$ for tuples $(B; A, x_0, x_1)$, all derived from genomic form and function data. Specifically, two data types binned at two different resolutions are used to construct triples of the form $(A, x_0, x_1)$, where $A$ is computed from data using one of the methods outline in section 2, $x_i$ is one of the initial $M = 8$ data snapshots representing FIB, and $x_1$ is the target snapshot representing ESC. For each triple, binding data for 222 transcription factors (TFs) is used to construct a set $B$ matrices, which are evaluated with $\mu_0$, $\mu_1$, $\mu_2$, and $\mu_3$.

In section 4.1 we describe the state data, and in section 4.2 we describe the TF binding data. In section 4.3 we briefly analyze the results of DMD applied to the data, and in section 4.4 we present the results of the control evaluation computations. Similarly, in section 4.5 we briefly analyze the results of the constrained $A$ identification, and in 4.6 we give the associated results.
4.1 State Data

Using fibroblast (FIB) and embryonic stem cell (ESC) data, we apply the methods from sections 2 and 3 to analyze FIB-to-ESC reprogramming. The data used for this application comes from [1]. We provide a summary of the data types and relevant information here and in the appendix, and refer the reader to [1] for a more detailed description and technical protocols.

Two biological data types (RNA-seq and Hi-C) were collected over eight time points for proliferating human primary fibroblasts (FIB), and at one time point for embryonic stem cells (ESC):

**RNA-seq** is next generation genome-wide sequencing that allows one to determine the quantity of specific RNA within a cell at a given time. Sequenced RNA is mapped back to the genome to determine where it was transcribed from, and each mapped sequence is referred to as a read. Reads are binned either by gene or by mega-base-pair (MB), and the units are normalized to *reads per kilo-base-pair per million* (RPKM). We will refer to the normalized data binned by gene and by MB as *gene-resolution* and *MB-resolution*, respectively.

**Hi-C**, or genome-wide chromosome conformation capture, is a population level genome-wide analysis that shows where specific portions of the genome interact in physical space with other portions. This is done by cross-linking DNA, ligation the DNA using restriction enzyme cutting sites, then sequencing and mapping the DNA stands. An observed interaction between two basepairs is called a read. Hi-C data is most easily visualized as a two-dimensional, binned, contact matrix $H$, where entry $H_{ij}$ represents the number of observed reads between bins $i$ and $j$. In future work we will consider the tensorial dynamics of the Hi-C matrix itself, but instead here we consider a vectorization of the two-dimensional data. The vectorization method is different for gene-resolution versus MB-resolution, but both rely on spectral graph theory. A Hi-C matrix can be thought of as an adjacency matrix, which allows the construction of a corresponding Laplacian matrix. The second smallest eigenvalue $\lambda_2$ of a graph’s Laplacian is known as the *Fiedler number* and measures the strength of the connectedness of the graph, with $\lambda_2 = 0$ for disconnected graphs and $\lambda_2 > 0$ for connected graphs. The corresponding eigenvector, known as the *Fiedler vector*, partitions the graph into strongly connected components. In particular for MB-resolution Hi-C data per chromosome, it has been shown that the sign of the Fiedler vector characterizes euchromatin and heterochromatin regions of the genome [1]. To vectorize MB-resolution, we bin Hi-C reads by MB blocks and compute the Fiedler vector of the corresponding Laplacian – effectively quantifying the chromatin accessibility of each MB bin. To vectorize gene-resolution, we bin Hi-C reads by HindIII cut sites for each gene and compute the Fiedler number of the corresponding Laplacian – effectively quantifying the self-connectedness or compactness of each gene. More details are given in the Appendix 0.1, and the same methods are also used in [1].

Following notation from section 2, the FIB and ESC data correspond to the initial and final state snapshots, $X_0$ and $x_1$, respectively. To notationally distinguish the different data resolutions and types, we will use uppercase subscripts $R$ and $S$ to denote MB-resolution functional (RNA-seq) and structural (Hi-C Fiedler vector) data, respectively, and lowercase subscripts $r$ and $s$ to denote gene-resolution functional (RNA-seq) and structural (Hi-C Fiedler number) data, respectively. That is, we consider four pairs of initial and final data, or datasets:

\[
(X_{0R}, x_{1R}) \\
(X_{0S}, x_{1S}) \\
(X_{0r}, x_{1r}) \\
(X_{0s}, x_{1s})
\]  

(25)

This notation is summarized in Table 1 for convenience, including the corresponding state dimensions $N$ for each dataset. Note a slight abuse of notation since $N$ has different values depending on the dataset, but the correct meaning should be clear from context.

For gene-resolution, although the data was obtained for over 22,000 genes, the RNA-seq observations for many of the genes were identically zero over the eight FIB time points and as
Subscript | Resolution | Data Type | N   
---|---|---|---
$R$ | MB | RNA-seq | 2,731  
$s$ | MB | Fiedler vector | 2,731  
$r$ | Gene | RNA-seq | 2,605  
$s$ | Gene | Fiedler number | 2,605  

Table 1: Summary of data notation.

Figure 2: (left) Gene-resolution FIB data at time point 1. The numbered ring represents the different chromosomes, with chromosome lengths scaled according to the number of genes considered on that chromosome. The outer blue ring plots RNA-seq data per gene, displayed with a logarithmic scale. The inner red ring plots gene Fiedler number. (right) MB-resolution FIB data at time point 1. Again the numbered ring represents the different chromosomes, here with chromosome lengths scaled according to the number of MB bins considered on that chromosome (roughly proportional to actual chromosome lengths minus centromeres). The outer blue ring plots RNA-seq data per MB bin. The inner red ring plots the MB Fiedler vector for each chromosome.

well as the single ESC time point, and many more were very close to zero. We only considered genes whose expression level went above 50 RPKM for at least one FIB time point or for the ESC time point, leaving us with $N = 2,605$ genes, in order to reduce the gene-resolution dimension $N$ without discarding important dynamic data.

For MB-resolution, starting from the p-arm of each chromosome, 1 MB bins were used for binning RNA-seq and Hi-C reads at MB-resolution into a total of 2,894 bins. Several MB bins per chromosome had zero Hi-C reads, typically in line with the chromosome centromeres. So that the Hi-C matrices represent connected graphs, any MB bin that had zero Hi-C reads (in any of the FIB time points or the ESC time point) were discarded, leaving $N = 2,731$ MB bins. Note that expression data was also typically close to zero for these bins.

Figure 2 visualizes FIB time point one data $x^1$ for all four datasets ($R, S, r, s$). The left circular plot contains gene-resolution data $x^1_{Rr}, x^1_{Rs}$, and the right contains MB-resolution $x^1_{Br}, x^1_{Bs}$. For reference, the numbered light/dark gray ring denotes the 22 autosomes (note that X and Y chromosomes were not included in our analysis).
4.2 TF Binding Data

A transcription factor (TF) is a protein that recognizes and binds to specific sequences of DNA or genes. A given TF can typically bind to many places throughout the genome. If a TF binds at or near a gene, the action of binding can serve to activate or repress the expression of the target gene. Because TFs recognize sequences in the linear genome, their potential binding locations are cell-type invariant; however, cell-type specific chromatin compactness and other genomic characteristics affect which binding locations are accessible per cell type [16].

TFs are products of natural gene expression, but can also be introduced exogenously, acting as a mechanism for controlling the genome. We will use binding data to construct our $B$ matrices for evaluation.

Potential cell-type invariant binding location data for 222 TFs was obtained using the FIMO tool from the MEME Suite package [34] on consensus sequence data from TRANSFAC [35, 16]. We derive FIB specific binding profiles for the TFs by eliminating those cell-type invariant binding sites located in heterochromatin portions of DNA, according to the MB-resolution FIB Hi-C Fiedler vector data.

For example, at gene-resolution $N = 2,605$, so let $b_{TF} \in [0,1]^{2,605}$ be a boolean vector of length $N$ with one entries corresponding to genes that contain identified binding sites for a specific TF. If our initial conditions are $x_{1r}^0$ (gene-expression FIB data at time point 1), it is known that some genes are heterochromatin and inaccessible to input, so some one entries of $b$ must be set to zero.

The total number of cell-type-invariant binding nodes $N_d$ for all 222 TFs is plotted in Figure 3 for MB and gene-resolution. The left plot shows, as one would expect, that generally those TFs that bind to a greater number of MB bins also bind to a greater number of genes. The right plot shows, for example, where in the genome the MYC TF (produced by the MYC gene) can bind.

4.3 DMD output

As described in section 2, we identify an $A$ matrix for each initial dataset $X_{0R}, X_{0S}, X_{0r}, X_{0s}$ using DMD, which gives

$$(A_R, \Phi_R, \Lambda_R), \ (A_S, \Phi_S, \Lambda_S), \ (A_r, \Phi_r, \Lambda_r), \ (A_s, \Phi_s, \Lambda_s),$$

respectively. Note that each dataset was verified to be linearly consistent (rank($X_0$) = 8), so in each case we have exact equality in (5). (Numerically we can check $\text{norm}(Y-A*X,'fro')$ for each dataset, yielding $S : 8.0681e-14, R : 9.739e-09, s : 1.0504e-13$ and $r : 2.0972e-10$). Linear
consistency is also the necessary condition for equivalence of Koopman and DMD modes, which establishes that DMD could indeed be capturing some underlying nonlinearity of the dynamics.

The DMD eigenvalues describe the stability and period of each corresponding DMD mode, and are plotted for the four datasets in Figure 4. As a sanity check we are glad to see some modes with periods between 16-30 hours, roughly in line with the 24 hour circadian clock. Furthermore, almost all eigenvalues have modulus less than 1, indicating stable decaying modes, consistent with known decay of the experimental cells.

It is interesting to notice at both resolutions, there is some apparent alignment of the structural and functional eigenvalues, i.e., $\Lambda_S$ with $\Lambda_R$ and $\Lambda_s$ with $\Lambda_r$. Although it is not within the scope of this paper to elucidate the relationship between form and function, an interesting direction for future work would be to further explore this.

We compute the angle between each mode of each dataset and the corresponding target difference vector $x_1 - x_0$. A small angle means alignment between the DMD mode and the target, whereas a 90° angle means the target is orthogonal to the DMD mode. For $S$, $s$, and $R$, we see the target is generally perpendicular to the dynamic modes, and for $r$ we see angles ranging from 20° to 70°. We interpret this to mean ESC is not in the natural direction of FIB proliferation. The calculation:

$$\theta_{ij} = \arccos\left(\frac{\text{abs}(\phi_j)}{\|\text{abs}(\phi_j)\|} \cdot \frac{x_1 - x_0}{\|x_1 - x_0\|}\right)$$

(26)
4.4 Evaluation Results

4.4.1 $\mu_1$

We first computed $\mu_1$ for 222 TFs with four different datasets and eight accessibility conditions. Figure 5 plots the average $\mu_1$ scores over the eight FIB time points, denoted $\mu_1$, for all 222 TFs, with respect to the control dimension coverage $N_d/N$, where $N_d$ is the average number of driver nodes $N_d$ over the eight FIB time points, for a given TF, and $N$ is the state dimension.

In all four datasets, there is a strong linear relationship between $N_d/N$ and $\mu_1$. This tells us that $\mu_1$ strongly depends on the number of nodes controlled by a TF. That is, TFs which influence more places in the genome yield (as we expect) higher average control scores. The dependence is so strong; however, that it doesn’t greatly distinguish between two TFs with similar $N_d$. Moreover, recall that $\mu_1$ is not target dependent.

Average controllability is not a suitable metric for distinguishing sets of driver nodes, or in our case TFs. It simply identifies $B$ with the largest $N_d$.

4.4.2 $\mu_2$

We computed $\mu_2$ for 222 TFs on the four datasets and eight accessibility conditions. Similar to Figure 5, Figure 6 plots the average $\mu_2$ scores over the eight FIB time points, denoted $\mu_2$, for all 222 TFs. A more interesting distribution of $\mu_2$ scores is observed than compared to $\mu_1$. Although there is still a strong correlation between $N_d$ and the controllability score, we see that $\mu_2$ better distinguishes between TFs with similar $N_d$ values than $\mu_1$. Recall that $\mu_2$ is a measure of closeness to the target, so a lower value is better, and we see in general that control over more nodes allows us to get closer to the target.

4.4.3 $\mu_3$

We computed $\mu_3$ for 222 TFs on the four datasets and eight accessibility conditions, similar to $\mu_1$ and $\mu_2$. The averaged results are plotted in Figure 7, with the vertical axes in log$_{10}$ scale.
Here the trends differ for each dataset.

4.5 Including Structural Constraints

To incorporate biology into the model, we impose structure on $A$ consistent with TF relationships, according to the methodology in section 2.3. First, we motivate the model we will consider:

We briefly venture away from the time-invariant model (2) and associated notation in order to propose a nonlinear model for the genome as an idea for future work. We then return to the linear setting using a similar model that includes structural constraints on $A$ and on which the controllability evaluations $\mu_1, \mu_2, \mu_3$ can be applied.

Let $x[k]$ denote RNA-seq data at time $k$ (either gene or MB-resolution), and $y[k]$ denote Fiedler vector data (MB-resolution). Consider the system

\[
\begin{align*}
\mathbf{x}[k+1] &= (A_1 + A_2(y[k])) \mathbf{x}[k] \\
\mathbf{y}[k+1] &= A_3 \mathbf{y}[k]
\end{align*}
\]

(27)

For the $x$ dynamics, the linear operator $A[k] := A_1 + A_2(y[k])$ is now dependent on $y$ and therefore on time. The proposed model has $A_1$ as the hardwired transcription factor network (idea first proposed in [36]), which is invariant between cell types, and $A_2(y[k])$ encodes the time-dependent and cell-type-dependent euchromatin/heterochromatin, accessible/inaccessible portions of the network.

Let $\sigma(y)$ denote where the element-wise sign of the vector $y$ is positive, e.g., $\sigma([0, -1, 2, 3, -4]) = [0, 0, 1, 1, 0]$. In this way $\sigma$ encodes euchromatin and heterochromatin with 1’s and 0’s, respectively.

Then define

\[ A_2(y) := -\text{diag}(1 - \sigma(y)) A_1 \]

(28)

so that

\[ A_1 + A_2(y) = (I - \text{diag}(1 - \sigma(y))) A_1 = \text{diag}(\sigma(y)) A_1 \]

(29)
effectively sets the rows of $\mathbf{A}_1$ corresponding to heterochromatin bins to 0. The bold $\mathbf{1}$ represents a vector of all ones of appropriate size and the ‘diag’ operator assigns a vector to a diagonal matrix (MATLAB notation). Equation (27) is rewritten as

$$\begin{align*}
\begin{cases}
\mathbf{x}[k+1] &= \text{diag}(\sigma(\mathbf{y}[k])) \mathbf{A}_1 \mathbf{x}[k] \\
\mathbf{y}[k+1] &= \mathbf{A}_3 \mathbf{y}[k]
\end{cases}
\end{align*}$$

Equation (30) is given by

$$\begin{align*}
\begin{cases}
\mathbf{x}[k+1] &= \text{diag}(\sigma(\mathbf{y}[k])) \mathbf{A}_1 \mathbf{x}[k] \\
\mathbf{y}[k+1] &= \mathbf{A}_3 \mathbf{y}[k]
\end{cases}
\end{align*}$$

In this formulation, the evolution of $\mathbf{y}$ is independent of $\mathbf{x}$, although in reality we expect it is also coupled somehow with $\mathbf{x}$. Structure can be imposed on $\mathbf{A}_1$ based on transcriptional interactions, where in particular $\mathbf{A}_{ij} \neq 0$ if and only if gene $x_j$ produces a transcription factor that can bind to gene $x_i$. As above, the set of non-interacting index pairs is denoted $\mathcal{I}_A$.

Having knowledge of $\mathbf{x}[k], \mathbf{y}[k]$ for $k = 1, \cdots, M$, $\mathbf{A}_3$ can be computed from DMD and $\mathbf{A}_1$ can be approximated by solving the optimization problem

$$\min_{\mathbf{A}_1} \sum_{i=1}^{M-1} \left\| \mathbf{x}[i+1] - \text{diag}(\sigma(\mathbf{y}[i])) \mathbf{A}_1 \mathbf{x}[i] \right\|$$

subject to the structural constraints $\mathcal{I}_A$.

Although analogous results to Theorems 1 and 2 exist for time-dependent linear systems, we instead present a time-independent version of this model and leave time-dependent data-guided control in general as future work. For the FIB time series, the sign pattern of the Fiedler vector does not change drastically over time, and therefore, let the vector $\hat{\mathbf{y}} := \min_k \mathbf{y}[k]$. Then $\text{diag}(\sigma(\hat{\mathbf{y}}))$ encodes with 1’s the MB bins which are euchromatin for all time points. Define

$$\mathbf{A}_c := \text{diag}(\sigma(\hat{\mathbf{y}})) \mathbf{A}_1$$

so that (30) becomes

$$\begin{align*}
\begin{cases}
\mathbf{x}[k+1] &= \mathbf{A}_c \mathbf{x}[k] \\
\mathbf{y}[k+1] &= \mathbf{A}_3 \mathbf{y}[k]
\end{cases}
\end{align*}$$
In this case $A_3 = A_{S}$ from DMD and has already been analyzed above. We now compute $A_c$ according to the optimization problem (9). Using the new matrix $A_c$, the same 222 $B$ matrices are evaluated again using $\mu_1$, $\mu_2$, and $\mu_3$.

The computations can be carried out for either gene- or MB-resolution expression data, and therefore let $A_c$ denote the matrix obtained for gene-resolution, and similarly $A_C$ for MB-resolution.

4.6 Constrained Results
We computed $\mu_1$, $\mu_2$, and $\mu_3$ for 222 TFs on the two coupled datasets and eight accessibility conditions. The averaged results are plotted in Figure 8.

5 Discussion

Dataset comparison. The alignment of the structural and functional DMD eigenvalues is the first striking similarity between the two datasets.

Reprogramming recommendations.

Shortcomings. The Fiedler vector is used to determine the accessibility of a node, but it does not do so perfectly. Theoretically, any heterochromatin region of DNA should have zero expression. There are 2,531 genes of the 2,605 considered which have a mean expression greater
than 1 RPKM; however, 25.8% of those genes are labeled heterochromatin by the Fiedler vector for at least one of the eight FIB time points. While we are impressed that the Fiedler vector accurately identified 74.2% of the expressed genes as euchromatin, we must admit that imperfect accessibility quantification can influence our results.

**Future Work.** We will consider tensor representation to simultaneously capture function, structure and dynamics, and develop data driven model identification and control approaches \[37, 38\].

**To-do List:**

1. DMD mode analysis
2. Fix figures where dots overlap

**References**


Appendix

0.1 Spectral Graph Theory

Graph theory plays an important role in modeling and analyzing the genome architecture and function. Relevant concepts are reviewed for supporting and framing our analysis [1, 2]. We define a graph \( G = (V, E) \) where \( V = \{v_1, v_2, \ldots, v_N\} \) is a finite set of vertices with the cardinality \( N \), and \( E \) is the edge set consisting of elements of the form \( \{v_i, v_j\} \). The adjacency matrix \( A(G) \) (or simply \( A \) for short) is the symmetric \( N \times N \) matrix encoding the adjacency relationships in the graph \( G \), such that \( [A(G)]_{ij} = 1 \) only if \( \{v_i, v_j\} \in E \), otherwise \( [A(G)]_{ij} = 0 \), with \( [G]_{ij} \) denoting the \( ij \)-th entry of its matrix argument. The degree of a given vertex, denoted by \( d(v_i) \), is the cardinality of the neighborhood set of \( v_i \). This degree is equivalently expressed via the adjacency matrix by \( d(v_i) = \sum_{j \in N} [A(G)]_{ij} \). The degree matrix, \( D(G) \), is defined as a diagonal matrix with the \( i \)-th diagonal entry given by \( d(v_i) \). The Laplacian of \( G \) is defined by

\[
L(G) = D(G) - A(G)
\] (34)

and the normalized variant is given by

\[
L(G) = D(G)^{-1/2} \left( D(G) - A(G) \right) D(G)^{-1/2}
\] (35)

For a connected graph, let the ordered eigenvalues of \( L(G) \) be denoted by \( \lambda_1, \lambda_2, \ldots, \lambda_N \). The relation \( 0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N \) holds. The second smallest eigenvalue \( \lambda_2 \) is called the Fiedler number, or Fiedler value of the graph, \( G \). The associated eigenvector is called the Fiedler vector. According to the entry signs of the Fiedler vector (+, or -) vertices of a graph can be grouped into two clusters, with each cluster having relatively stronger within-cluster connections and weaker between-cluster connections [2]. More generally, instead of considering binary connections between pairs, weights can be assigned to each edge such that \( [A(G)]_{ij} = w_{ij} \) only if \( \{v_i, v_j\} \in E \) (otherwise \( [A(G)]_{ij} = 0 \)) to characterize the connection strengths. The associated degree matrix, \( D(G) \), and Laplacian, \( L(G) \), are still defined in the same way as previously presented. Fiedler number and clustering based on the Fiedler vector can also be defined and performed identically.

The Fiedler vector can also be used for extracting topologically associating domains (TADs). The sign pattern of a Fiedler vector can divide the chromosome to local units, which can be considered as TADs at relatively large scale. In order to get finer structures, we can recursively compute the Fiedler vector to split obtained TADs until the Fiedler value of the region is higher than some threshold.

0.2 Koopman Mode Analysis

In this section we briefly review Koopman operator theoretic framework for nonlinear analysis, further details can be found in [13, 14]. Consider an autonomous nonlinear discrete time dynamical system

\[
x_t = f(x_{t-1})
\] (36)

where, \( x_t \in M \subseteq \mathbb{R}^d \) is a state vector, \( f : M \rightarrow M \) is a function which describes the nonlinear state evolution. Let \( \mathcal{F} \) be space of observables which are scalar-valued functions \( \theta : M \rightarrow \mathbb{C} \) (where, \( \mathbb{C} \) denotes the complex plane) defined on the state space. The Koopman operator is a linear operator \( \mathcal{U} : \mathcal{F} \rightarrow \mathcal{F} \) which maps \( \theta \) into a new function \( \mathcal{U}\theta \), as follows

\[
(\mathcal{U}\theta)(x) = \theta(f(x)).
\] (37)

Although the dynamical system (36) is nonlinear and evolves on a finite-dimensional space, the Koopman operator \( \mathcal{U} \) is linear but infinite-dimensional. The eigenvalues \( \lambda \) of Koopman

operator, referred to as Koopman eigenvalues (KEs), and the eigenfunctions $\phi$ of Koopman operator, referred to as Koopman eigenfunctions (KEFs) are defined as follows:

$$\mathcal{U}\phi = \lambda \phi.$$  \hfill (38)

The set of all KEs $\lambda_j, j = 1, 2, \cdots$ is called the point spectrum of the Koopman operator [15].

Let $h : \mathbb{M} \rightarrow \mathbb{R}^m$ be a vector valued observable. If each of the $m$ components of $h$ lie within the span of KEFs $\phi_j, j = 1, 2 \cdots$, then $h$ can be expanded in terms of these eigenfunctions as [15],

$$h(x) = \sum_{j=1}^{\infty} \phi_j(x)u_j,$$  \hfill (39)

where, $u_j \in \mathbb{C}^m$ are complex valued vectors. Since, $\phi(x_t) = \lambda \phi(x_{t-1})$, the time evolution $h(x_t)$ can be expressed as

$$h(x_t) = \sum_{j=1}^{\infty} \lambda_j^t \phi_j(x_0)u_j.$$  \hfill (40)

We will refer to this expansion as Koopman Mode Analysis (KMA) or Koopman Mode Decomposition (KMD) following [39], with $u_j$ being the Koopman modes (KMs) associated with eigenfunction $\phi_j$ and the observable $h$. The modes are spatial fields that identify spatial correlations in the data, while the corresponding eigenvalues define growth/decay rates and oscillation frequencies for the mode. If the dynamics have only a finite number of discrete spectra (peaks) in complex plane, then a finite truncation of expansion (40) gives a good approximation of the dynamics.

KMD can be thought of as a generalized Fourier analysis, and offers several advantages over Discrete Fourier Transform [40]. Also note that compared to Principal Component Analysis (PCA) modes, each KM represents only one frequency component. Thus, KMD is expected to decouple dynamics at different time scales more effectively [41]. Koopman analysis has been applied in several domains such as fluid mechanics [42, 40, 13], model validation [43], building diagnostics [44], power grids stability analysis [41, 39], computer vision applications [45, 46], nonlinear stability analysis [47], sensor fusion [48], and nonlinear estimation [49] to name a few.

### 0.3 Connection Between DMD and KMD

Dynamic Mode Decomposition (DMD) originally introduced in fluids community [50], characterizes the nonlinear dynamics through analysis of some approximating linear systems. It uses Arnoldi type methods for computing DMD eigenvalues/modes as empirical Ritz values/vectors from sequential data. There is of course no guarantee that analyzing the DMD approximating operator is meaningful for data generated by nonlinear dynamics. It was shown in [42] that under certain conditions DMD eigenvalues/modes are equivalent to KEs/KMs. This connection has been further strengthened in exact DMD approach [10], which generalizes the notion of approximating linear system used in DMD, and is able to handle non-sequential data. Several other variants, e.g., optimized DMD [40], sparse DMD [51], streaming DMD [52] and multi resolution DMD [53] have also been proposed.

We next discuss the conditions under which exact DMD produces KEs/KMs/KEFs. Let $\{x_1, \cdots, x_N\}$ be different initial conditions and let

$$y_k = h(x_k), \quad \tilde{y}_k = h(f(x_k)).$$  \hfill (41)

where, $h = (h_1, \cdots, h_m)^* \in \mathbb{R}^m$ is an observable. Let $X$ and $X'$ be data matrices whose columns are formed by $y_k$ and $\tilde{y}_k$. This is as in usual DMD, except that we have access to an observable $h$ rather than the direct state $x$.

The dataset $\{y_k, \tilde{y}_k\}_{k=1}^N$ is said to be linearly consistent if the null space of $X'$ contains the null space of $X$. For example, if $X$ is full rank than its null space is $\{0\}$ and thus the dataset is linearly consistent trivially. Data consistency can be checked by verifying the condition $X'(I - X'^tX) = 0$, see [10].
We next discuss the necessary and sufficient conditions giving connection between exact DMD and KMD, see [10] for further details. Qualitatively, the sufficient condition require that the “observables” must span a space that contains the relevant KEFs, and the data must be sufficiently rich to capture their dynamical behavior. The necessary condition is related to linear consistency of the data.

1 Sufficient Condition: Let \( \phi \) be KEF with KE \( \lambda \) in the span of scalar observables \( \{h_i\}_{i=1}^{m} \), i.e.
\[
\phi(x) = w^*h(x)
\]
and let \( w \in \mathcal{R}(X) \), where \( \mathcal{R} \) denotes the range space. Then \( w \) is adjoint DMD mode, i.e.
\[
w^*A = \lambda w^*.\]
Thus, if the set of observable is sufficiently large (\( \phi \in \{h_i\}_{i=1}^{m} \)) and data is sufficiently rich i.e. \( w \in \mathcal{R}(X) \), then adjoint DMD modes and eigenvalues correspond to KEF and KE, respectively.

2 Necessary Condition: If \( \phi(x) = w^*h(x) \) and \( w^*A = \lambda w^* \), then \( w^*AX = w^*X' \). Note that if data is linearly consistent, i.e. \( AX = X' \) then above condition is satisfied trivially. However, it is not necessary for all the data to be linearly consistent in order for a particular eigenvalue/eigenvector of \( A \) to correspond to a true KE/KEF: it is necessary for the data to be linearly consistent in a particular direction, namely that determined by the corresponding left eigenvectors \( w \). Finally, note that while linear consistency is necessary condition for an eigenvector/eigenvalue of \( A \) to correspond to a KE/KEF, it is not sufficient condition.

The connection between Koopman modes and DMD modes goes as follows. Let \( A \) have a full set of right eigenvectors \( \{v_i\}_{i=1}^{n} \), and let \( w_i \) be the corresponding left eigenvectors. Assume normalization \( w_i^*v_j = \delta_{ij} \). Thus, for any observable \( h(x) \), we get
\[
h(x) = \sum_{i=1}^{n} (w_i^*h(x))v_i = \sum_{i=1}^{n} \phi_i(x)v_i,
\]
and,
\[
h(f(x)) = \sum_{i=1}^{n} \lambda_i \phi_i(x)v_i,
\]
provided sufficient condition above holds true. Thus,
\[
h(x_t) = \sum_{i=1}^{n} \lambda_i \phi_i(x_{t-1})v_i,
\]
which is analogous to KMD (40), and this \( v_i \) are the KMs \( u_i \).

In summary under appropriate conditions as stated above, DMD eigenvalues and DMD modes correspond to KEs and KMs, respectively while the adjoint DMD modes define the KEFs.

0.4 Scaling of DMD Modes

DMD practitioners often identify DMD modes of interest using their norms: it is generally assumed that modes with larger norm are dynamically more important. For sequential time series, this if often done using so-called DMD spectra, which plot DMD mode norms as function of their corresponding frequencies in Hz, one can use
\[
f_i = \frac{\text{imag } \log(\lambda_i)}{2\pi \Delta T}
\]
where, \( \Delta T \) is sampling period in seconds. Sometimes, the norms of weighted by the magnitude of the corresponding DMD eigenvalues to penalize spurious modes with large norms but quickly decaying contributions to the dynamics.

Another approach proposed in [10] it to normalize the right eigenvectors of \( \tilde{A} \), i.e. \( \tilde{v} \) to have a unit norm.

A different scaling has been proposed in [54]. Let \( \tilde{A} = \Sigma^{-1/2}\tilde{\Lambda}\Sigma^{1/2} \), then the eigenvalues of \( \tilde{A} \) are identical to the eigenvalues of \( \hat{A} \), while the eigenvectors of \( \tilde{A} \) get scaled by \( \Sigma^{1/2} \).