Published in final edited form as:

Phys Biol. 2012 August ; 9(4): 045004. doi:10.1088/1478-3975/9/4/045004.

Modeling the latent dimensions of multivariate signaling datasets

Karin J. Jensen and Kevin A. Janes

Department of Biomedical Engineering, University of Virginia, Charlottesville, VA 22908 USA

Kevin A. Janes: kjanes@virginia.edu

Abstract

Cellular signal transduction is coordinated by modifications of many proteins within cells. Protein modifications are not independent, because some are connected through shared signaling cascades and others jointly converge upon common cellular functions. This coupling creates a hidden structure within a signaling network that can point to higher-level organizing principles of interest to systems biology. One can identify important covariations within large-scale datasets by using mathematical models that extract latent dimensions—the key structural elements of a measurement set. In this tutorial, we introduce two principal components-based methods for identifying and interpreting latent dimensions. Principal components analysis provides a starting point for unbiased inspection of the major sources of variation within a dataset. Partial least squares regression reorients these dimensions toward a specific hypothesis of interest. Both approaches have been used widely in studies of cell signaling, and they should be standard analytical tools once highly multivariate datasets become straightforward to accumulate.

1. Introduction

Biology is now awash with large-scale measurements of cell signaling [1]. High-throughput technologies can readily measure signaling-protein levels and modification states such as phosphorylation. Moreover, we can observe dozens to thousands of post-translational modifications and how they change with time under different environmental conditions and perturbations [2–6]. These modifications to signaling proteins propagate information flow through the cell. The question is how best to use these data to uncover patterns of regulation that may suggest how the underlying network operates.

If a large-scale dataset revolves around a single perturbation or stimulus, then “hit lists” ranking the largest-magnitude changes may suffice for gene or pathway discovery. However, when numerous perturbations or stimuli are involved concurrently, it is much harder to link stimulus- or perturbation-induced changes within the cell to phenotypic outcomes. For example, a specific small-molecule inhibitor should strongly block the activity of the target enzyme, but what happens if multiple inhibitors are combined and the cells are also challenged with a microbial pathogen? To make these types of inferences, we need data reflecting complex biological scenarios and a simplified representation of the measurements—we need a data-driven model [7].

Complex datasets benefit from models that address the fundamental challenge of dimensionality [8]. When large spreadsheets of measurements are recast as a vector algebra [7], each experimental condition appears as a projection along a set of dimensions defined by the measured variables (see below). If we could inspect the condition-specific projections along all measured variables (e.g., post-translational modifications), then we could possibly discern patterns within the measurements. The problem is that when interpreting highly multivariate datasets with hundreds or thousands of dimensions, we struggle to have

intuition beyond the three dimensions that we can see [9, 10]. Data-driven models simplify dimensions according to specific quantitative criteria, identifying a small number of “latent variables” that comprise a reduced dimensional space for prediction and analysis.

Dimensionality reduction of signaling data remains an active area of research [11, 12], but here we will review two established methods: principal components analysis (PCA) and partial least squares regression (PLSR). PCA and PLSR have been applied to signal transduction over the past several years, but they have a much longer history in data-rich fields such as spectroscopy, econometrics, and food science [13, 14]. The main distinction between the two methods lies in the overarching goal of the resulting model. PCA is an unsupervised method, meaning that dimensions are reduced based on intrinsic features of the data. Thus, PCA allows the data to “speak for itself”, but the corollary is that method is deaf to user input regarding the types of relationships that latent dimensions should uncover. It is here that PLSR excels. As a supervised method, PLSR starts with a hypothetical relationship between variables (dimensions) that are independent and those that are dependent. The algorithm then reduces dimensions to retain the hypothesized relationship as much as can be supported by the data by creating a linear regression model. In contrast to PCA, there are countless user-defined hypotheses that can be tested with PLSR, and models can even predict dependent variables given new input data. Thus, PCA is most useful as an explanatory tool for unbiased discovery of patterns within datasets [15, 16], whereas PLSR acts as a predictive tool for linking multivariate inputs to outputs [17–23]. The limitation of both of these methods is that the predictions are correlative—they provide a guide for causation that must be tested subsequently with mechanistic experiments.

The goal of this tutorial is to provide readers with a working knowledge of PCA–PLSR and their application to cell-signaling datasets. We begin by reviewing the basics of vector and matrix algebra that are essential for understanding how latent dimensions are identified. Then, we will provide a detailed introduction to PCA and PLSR, focusing on the underlying mathematics and the technical considerations important for real biological applications. In parallel, we will walk through an example of dimensionality reduction using a primary signaling dataset from a recent publication [24] (dataset and MATLAB scripts are available in the Supplementary Data). We conclude with a brief discussion of more advanced approaches and refer interested readers to further literature on these topics.

2. Vector and matrix algebra

Latent variables align with the important changes in a high-dimensional dataset. A model using latent variables must know where these changes “point” and how far each data point “moves” with respect to them. Therefore, it is natural to treat latent variables as *vectors*, which are defined as mathematical entities with a direction and a magnitude. Using vectors lends to a geometric interpretation where we can compare the angle and distance between the two vectors. Importantly, vector algebra easily generalizes to beyond three dimensions, which is essential for modeling large multivariate datasets. As we expand into higher dimensions, the utility of the vector description and its geometric interpretations becomes even more valuable.

The magnitude of a vector is often referred to as the vector *norm*. For a vector x in n dimensions,

$$x = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}$$

the Euclidean norm ($\|x\|$) is calculated as follows:

$$\|x\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$$

The vector norm is useful for normalization, since dividing a vector by its norm yields a scaled vector of unit length (or a *unit vector*).

Vectors with the same number of dimensions can be grouped together as a *matrix*. We refer to a matrix X with m rows and n columns as an $m \times n$ matrix. To help distinguish between vectors and matrices, we often name matrices with capital letters and vectors with lowercase letters. (Individual matrix elements use lowercase letters with two subscripts that refer to the row and column of the matrix.) X can thus be viewed as a collection of m row vectors (r), each with n dimensions, or as a collection of n column vectors (c), each with m dimensions:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix}$$

Matrices directly correspond to the data tables of large-scale experiments. The columns indicate the signaling *variables* measured during the experiments: phospho-proteins, catalytic activities, etc. The rows indicate the experimental *observations* where the variables were measured: a particular growth factor stimulation, time point, genetic background, etc. Each condition can thus be treated as a row vector, whose individual elements indicate how strongly that row vector projects upon the dimensions specified by the variables.

Treating data tables as matrices allows linear algebra to be performed on the matrix en route to building a data-driven model. However, while a matrix may appear similar to an array, it is important to recall that algebraic operations on matrices and vectors differ from those on arrays. When a matrix A is multiplied by a vector x (Ax), the result is a projection of the matrix onto that vector space. The vector samples the matrix across the shared dimension, combining the products of the individual elements. By extension, when two matrices are multiplied, we combine the products of the row elements of the first matrix with the corresponding column elements in the second matrix across the shared dimension. Thus, matrix and vector multiplication is defined only when the two elements share the inner dimension of the product. For an $m \times p$ matrix A and a $p \times n$ matrix B , the product AB exists and yields an $m \times n$ matrix (with the product summed across the shared inner dimension, p). By contrast, BA does not exist unless $m = n$. A real number can multiply a vector or matrix of any size by scalar multiplication, which simply scales each element of the vector or matrix by the real number.

One convenient transformation that we can perform on a matrix or vector without loss of information is to take its *transpose*, denoted by the superscript T . The transpose of a matrix is the same matrix where the rows in the matrix become the columns in the transposed matrix and the columns in the original matrix become the rows (for a square matrix, this is equivalent to reflecting the matrix across its main diagonal). So while a 2×3 matrix C cannot be multiplied by itself, C can be multiplied by C^T :

$$C = \begin{bmatrix} 2 & 1 & 3 \\ 1 & 6 & 5 \end{bmatrix} \quad C^T = \begin{bmatrix} 2 & 1 \\ 1 & 6 \\ 3 & 5 \end{bmatrix} \quad CC^T = \begin{bmatrix} 14 & 23 \\ 23 & 62 \end{bmatrix}$$

If a row or column of a matrix is simply a linear combination of the others, then there is no additional information provided by this row or column. The *rank* of a matrix tells us how many dimensions are truly distinct from one another. If each row of a matrix is an observation, the rank will tell us how many observations provide unique data. The rank also indicates how many variables (columns) are unique. The rank of the above matrix C is two, because one column can always be expressed as a linear combination of the other two; for example:

$$\frac{13}{11} \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \frac{7}{11} \begin{bmatrix} 1 \\ 6 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

Vectors in a matrix can be linearly independent, but that does not mean that their directions are non-overlapping. For instance, the second and third column vectors of C are independent, but their projection onto one another is nonzero ($1 \times 3 + 6 \times 5 = 33$), meaning that they partly point in the same direction. When vectors have a zero projection onto one another, they are said to be *orthogonal*. The simplest example of orthogonal vectors is the Cartesian pair of unit vectors: $[1 \ 0]$ and $[0 \ 1]$. Other two-dimensional vectors can be orthogonal—such as $[2 \ 1]$ and $[1 \ -2]$ —and the principle of orthogonality extends naturally to higher dimensions. If we can extract orthogonal vectors from a data matrix, these can serve as new axes onto which the data can be projected. The new axes act as latent variables in our experiment and are defined as a weighted combination of the variables that we measured. We are interested in choosing vectors that are orthogonal because the latent variables they define cannot be projected onto one another, making them more interpretable. This analysis can simplify the data by projecting it onto a smaller set of orthogonal vectors, which identify groups of variables that fluctuate together and describe the data more efficiently.

One way to identify orthogonal vectors is by calculating the *eigenvectors* of a data matrix. The eigenvectors (x) of a square $n \times n$ matrix A are special, in that they return a scaled version of themselves when multiplied by A :

$$Ax = \lambda x$$

where λ is a scalar called an eigenvalue. The eigenvector can be used as a new axis onto which we can project the data, with the eigenvalue indicating how strongly the data is projected onto that “axis”. Eigenvectors by definition are always orthogonal to one another; therefore, given a matrix of rank n , there can be up to n eigenvectors with their associated eigenvalues.

We defined eigenvectors for a square matrix where the number of observations equals the number of variables, but what if the data matrix is not square? Here, one can factorize a matrix into the product of three matrices by the process of *singular value decomposition* (SVD):

$$X = U \Sigma V^T$$

U and V^T are the left and right matrices of *singular vectors*, which are mutually orthogonal and are conceptually similar to eigenvalues. Σ is a diagonal matrix comprised of *singular values*, which are analogous to eigenvalues and correspond to the weightings of the singular vectors. By convention, the singular values in matrix Σ are organized in decreasing order from top left to bottom right. Focusing on the singular vectors with the largest associated singular values, it is possible to extract systemic patterns from large biological datasets [15].

3. Principal components analysis

SVD is not the only way to break down a data matrix into digestible pieces. *Principal components analysis* (PCA) decomposes an $m \times n$ matrix X into the product of two matrices:

$$X = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1\text{rank}(X)} \\ t_{21} & t_{22} & \cdots & t_{2\text{rank}(X)} \\ \vdots & \vdots & \ddots & \vdots \\ t_{m1} & t_{m2} & \cdots & t_{m\text{rank}(X)} \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{\text{rank}(X)1} & p_{\text{rank}(X)2} & \cdots & p_{\text{rank}(X)n} \end{bmatrix} = TP^T$$

where T is comprised of row vectors called *scores* (t) and P is comprised of column vectors called *loadings* (p). The scores and loadings vectors are ranked by their contribution to the overall variance in the dataset, so that models of the data can be built from the leading score-loading pairs:

$$X \approx \begin{bmatrix} t_{11} \\ t_{21} \\ \vdots \\ t_{m1} \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \end{bmatrix} + \begin{bmatrix} t_{12} \\ t_{22} \\ \vdots \\ t_{m2} \end{bmatrix} \begin{bmatrix} p_{21} & p_{22} & \cdots & p_{2n} \end{bmatrix} + \dots = t_1 p_1^T + t_2 p_2^T + \dots$$

The $t_i p_i^T$ vector product indicates the variance captured by the i^{th} *principal component* of the data. Principal components act as the latent variables of a PCA model.

Just as with eigenvectors and singular vectors, a defining characteristic of principal components is their orthogonality: each t is orthogonal from all other score vectors and each p^T is orthogonal from all other loadings vectors. While orthogonality constrains the direction of the vectors with respect to one another, it does not uniquely define the direction of any one vector. This yields an infinite number of possible solutions to the principal-component problem. PCA arrives at a solution by further imposing two criteria: principal components must 1) have loadings vectors of unit length (for convenience) and 2) must explain as much of the variance of the original or residual data as possible. Principal components are calculated iteratively, with first principal component maximizing the variance captured, and the second principal component calculated with the residual information not captured by the first principal component ($X - t_1 p_1^T$). By solving for the principal components that capture the maximum variance, new axes are identified that more efficiently represent the information contained in the original dataset. The early principal components are comprised of loadings vectors paired with the large-magnitude score vectors, indicating that the data is most strongly projected onto these principal component “axes”. Later principal components carry small-magnitude score vectors, such as noise among observations, which can be omitted from the PCA model.

There are several conceptual and mathematical similarities between PCA and SVD. The major distinction is that SVD is performed directly on X , whereas PCA is calculated using the covariance matrix of X . Loadings vectors (the columns of P) are the eigenvectors of the column covariance matrix (calculated as $X^T X$ if X has been preprocessed appropriately; see below), and row vectors (of the scores matrix T) are the eigenvectors of the row covariance matrix (calculated as XX^T). Indeed, if X is preprocessed by column centering (by subtracting the mean of each column from each entry in the column), then P^T obtained by PCA is equivalent to V^T obtained by SVD, with $T = U$. The correspondence breaks down when X is not preprocessed before PCA, because $X^T X$ then carries with it the mean values of the different variables in X (see below).

To illustrate the application of PCA in practice, we selected a primary signaling dataset of phosphoprotein measurements collected by phospho-ELISA [24] (figure 1(a)). HL1 murine cardiomyocytes were pretreated with one of seven signaling inhibitors or 15 inhibitor pairs and then infected with the cardiotoxic virus, coxsackievirus B3 (CVB3). At six time points over 24 hr after CVB3 infection, cells were analyzed for nine phosphoproteins: Akt, ATF2, CREB, ERK, GSK3 β , HSP27, I β , JNK, and P38. In parallel, the extent of viral progeny release (VPR) was measured by plaque assay as a measure of productive virus infection and propagation within host cells. The VPR measurements will be used later when building a supervised model with PLSR.

We first simplify the dynamic nature of the phosphoprotein measurements by taking the time integral. Although not required for PCA, the time integral provides a simple metric that captures the magnitude and duration of the signaling event [25]. Integrated signals have been informative in many data-driven models and are thus used here for simplicity [17–19, 26]. However, we note that many other metrics have been explored and could be useful in specific circumstances, such as for multiphasic signaling events [17, 19, 27].

Next, we must preprocess the data so that each phosphoprotein has an equal opportunity to contribute to the principal components of the model. As mentioned above, data-driven models are built by retaining the largest variations or covariations within a dataset [7, 13, 14]. Unprocessed data carries with it the heterogeneous characteristics of the original signaling measurements [1], which can bias a model. In the original phospho-ELISA measurements of figure 1(a), for example, the mean of the AKT time integral (taken across all inhibitor conditions) is 174.2 and its variance is 1360. By contrast, the mean of P38 is 1043.5 and its variance is 247,880. Using these raw data, P38 will contribute much more to the model relative to AKT, simply because the mean values are larger and they vary more. The contribution is disproportionate, however, because the larger and more-variable P38 measurements may simply reflect a more-sensitive assay. Thus, some form of preprocessing is required to place all the data on equal footing.

Using fold-change normalizations is customary in cell-signaling studies [28–30], but this only partly accounts for the intrinsic differences among measurements. The more-common way to preprocess a measurement is to center each variable by subtracting its mean and then scale the centered variable by dividing by its standard deviation. This standardization turns each variable column into a nondimensional z-score, which indicates the number of standard deviations that an observation lies away from the mean. Centering is not appropriate when baseline data are included in the model, because the act of centering will move the baseline from zero to a negative value [13, 17]. However, in this example, all samples were infected with CVB3 and thus z-scoring is appropriate without uninfected baseline data.

After preprocessing the phospho-ELISA dataset (X), we must now define scores and loadings vectors according to the eigenvalue profile as introduced above. Numerically, this

can be achieved by the nonlinear iterative partial least squares (NIPALS) algorithm [14]. NIPALS starts by selecting a row vector from X and defining it as a provisional score vector (t_1 for the first principal component). X is then projected onto t_1 to define a provisional loadings vector (p_1):

$$p_1^T = \frac{t_1^T X}{t_1^T t_1}$$

Next, X is projected onto this loadings vector after normalization ($p_{1,norm}$) to define a new score vector ($t_{1,new}$):

$$p_{1,norm} = \frac{p_1}{\|p_1\|}$$

$$t_{1,new} = X p_{1,norm}$$

If $t_{1,new} = t_1$ within a numerical tolerance, then the NIPALS algorithm has converged, and t_1 and $p_{1,norm}$ represent the scores and loadings vectors for the first principal component. The contribution of the first principal component is then subtracted from X , and the second principal component is derived from the residual in the same way that the first was derived from X . This iterative procedure can continue up to the rank of X , but often the first several principal components (eigenvalues of the covariance matrix) contain the bulk of the standardized variation within the data. For example, the two leading principal components of our phospho-ELISA dataset capture 45% of the overall variation, and the first five capture 87%.

We can graphically portray the projections of the signaling variables onto the PCA principal components via a loadings plot (figure 1(b)). Variables that are similarly loaded along a principal component share common features. For example, the phospho-kinases ERK, GSK3, and JNK are all loaded strongly along the second principal component, suggesting that they are regulated together in the context of the viral infection. We would thus expect to see these proteins show a similar behavior pattern under the different treatments. This is most clearly visible in the first two inhibitor combinations (figures 1(a)–(b), single asterisks) where ERK, GSK3, and JNK share a strong relative activation. However, the weaker covariation of the three kinases in other inhibitor combinations also informs the analysis and helps to determine their final projection. The simplest explanation for the ERK–GSK3 – JNK grouping is that these pathways are activated by a common upstream signaling event, possibly via one of the initial stages of viral infection. Identifying this signaling event could be a motivation for future experimental studies, which focus on signaling proteins upstream of ERK, GSK3, and JNK.

Another example of grouped phospho-proteins is P38, CREB, and HSP27, which are loaded along the first principal component in part because they are all strongly activated by the eighth and twelfth inhibitor combinations (figures 1(a)–(b), double asterisks). Note also that ATF2 is positively loaded on the first principal component but negatively loaded on the second principal component, because it is strongly activated by conditions 8 and 12 but inhibited by conditions 1 and 2. AKT and I B are not strongly loaded on either principal component, and inspection of the primary data indicates why—neither is consistently activated or inhibited by conditions 1+2 or 8+12. Accordingly, these two phospho-proteins are strongly loaded on the third principal component as part of the residual information not captured by the first two principal components (not shown).

Data-driven modeling results should always be followed up in this way with respect to the primary dataset. It provides a reality check for the model and a more objective appreciation for patterns in the measurements. Although conditions 1+2 or 8+12 could be rationalized as discriminating conditions after PCA, they probably would not be the first patterns identified by visual inspection. Therein lies a major strength of data-driven modeling approaches [7].

4. Partial least squares regression

PLSR extends naturally from PCA with a few important additions. First, rather than decomposing a single matrix into scores and loadings, the data is split into two blocks that specify a hypothesized relationship between them. The independent block (X) contains the input variables, and the dependent block (Y) contains the output variables. The hypothesis is that the input variables determine the output variables. In our example, we retain the phospho-ELISA measurements as the independent block and further assign the VPR measurements of viral propagation as the dependent block (figure 1(a)). Thus, our working hypothesis is that the phospho-proteins in X quantitatively control the VPR outcomes in Y .

Both the independent and dependent blocks can now be decomposed into their respective scores-loadings vectors:

$$\begin{aligned} X &= TP^T \\ Y &= UQ^T \end{aligned}$$

The relationship between the two blocks is specified as a linear regression between the scores of the independent block and the dependent block:

$$U \approx TB \Rightarrow Y \approx TBQ^T$$

(Note that, in our example, Y has a rank of one and therefore $Y \approx TB$.) The regression coefficient matrix (B) allows predictions of Y to be made from condition-specific score projections of X .

We could decompose X and Y separately by PCA and regress the principal components afterwards (this approach is termed principal components regression [14]). However, PLSR models improve the relationship between the two blocks by implementing a key algorithmic change in the NIPALS algorithm. Rather than have two PCA-type decompositions of X and Y occur independently, the scores vectors t and u are exchanged during each iteration to define a principal component. X is first projected onto a provisional score vector selected from Y (u_1 for the first principal component) to define a provisional loadings vector (p_1):

$$p_1^T = \frac{u_1^T X}{u_1^T u_1}$$

X is projected onto this loadings vector after normalization to define a provisional score vector for X (t_1), and then Y is projected onto t_1 to define a provisional loadings vector for Y (q_1):

$$p_{1,norm} = \frac{p_1}{\|p_1\|}$$

$$t_1 = X p_{1,norm}$$

$$q_1^T = \frac{t_1^T Y}{t_1^T t_1}$$

Last, Y is projected onto the normalized q_1 to define a new score vector ($u_{1,new}$) that will be projected on X in the next iteration:

$$q_{1,norm} = \frac{q_1}{\|q_1\|}$$

$$u_{1,new} = Y q_{1,norm}$$

The algorithm stops when score vector (t_1) for X has stabilized as described for PCA above. The projection of X onto u_1 and Y onto t_1 is what biases the decomposition of each block toward the variance of the other. The result is a principal components-based model that is optimized to capture the overall variation in Y by using the variation in X .

PLSR typically leads to a dramatic reorganization of the loadings vectors relative to those obtained by PCA [7]. For our phospho-ELISA–VPR model, we see many new groupings of signaling variables (figure 1(c)). P38 now maps with JNK and GSK3 in the second quadrant of the loadings plot, because all three phospho-kinases have a notable anticorrelation with VPR (figures 1(a), (c)). ERK and CREB also are slightly anticorrelated with VPR. However, some of the later inhibitor combinations have high ERK–CREB and high VPR, which causes these two proteins to be loaded less heavily on the second principal component compared to P38, JNK, and GSK3. As with PCA, these types of interpretations are critical for getting the most out of a PLSR model. The difference is that the PLSR loadings highlight specific groups of correlations between the independent and dependent blocks, whereas PCA loadings highlight coordinated variation more generally.

Interpretability of a data-driven model is often improved by “subspace rotation” [15]. Even though PCA and PLSR principal components are uniquely determined, the factorization is degenerate (just as the number eight can be factored into 8×1 , 4×2 , etc.). We can exploit this property by rotating the scores and loadings vectors so that they coincide with specific variables and conditions more directly. For example, note in the loadings plot of the phospho-ELISA–VPR model that most of the phospho-proteins are diagonally situated along the first two principal components. By rotating the axes counterclockwise by $\sim 35^\circ$ (figure 1(c), gray), we define a new coordinate system that is equally predictive (provided that the scores vectors and regression coefficients are similarly rotated) but cleaner in its loadings. In the new coordinate system, the first axis consists of positive contributions from AKT and negative contributions from ERK–CREB, whereas the second axis consists of positive contributions from P38–JNK–GSK3 and negative contributions from I B –ATF2. These types of rotations are valuable when seeking to assign biological functions to latent dimensions [17, 21].

One question related to interpretability is how many latent variables should be retained in the model. More principal components will obviously capture more of the training data, but there is a danger of overfitting. Fortunately, PLSR has an objective strategy that uses crossvalidation to identify the optimal number of latent dimensions. Before explaining the crossvalidation procedure, however, we must clarify the distinction between model calibration and model prediction. Model calibration involves the $Y_{cal} \approx TBQ^T$ values associated with observations that were used during the model training. Thus, Y_{cal} indicates

fitted values, whose root mean-squared error from the measured values (Y_{meas}) will always decrease when more principal components are used (figure 1(d), solid). Conversely, prediction involves the $Y_{pred} \approx TBQ^T$ values associated with observations that were omitted from the model training. Y_{pred} indicates true predictions as long as the input observations are independent from the training set. Importantly, Y_{pred} error *increases* beyond a critical number of principal components (figure 1(d), hollow), which is a red flag for overfitting.

During “leave-one-out” crossvalidation, each observation is withheld individually and a PLSR model is built on the remaining $m - 1$ observations. Then, the trained model is asked to predict the one observation withheld from the training, and the withholding-training-prediction cycle is repeated for all m observations. This procedure generates a crossvalidated prediction for each observation. It also allows an estimate of prediction uncertainty based on how much the calibration fluctuates when the other observations are withheld. The “jackknifed” standard error of the prediction ($SE_{Y_{pred}}$) [31] is defined as:

$$SE_{Y_{pred}} = \sqrt{\frac{m-1}{m} \sum_{i=1}^{m-1} (Y - Y_{cal,i})^2}$$

where $Y_{cal,i}$ represents the model calibration from the i^{th} crossvalidation run that includes Y . Other error models exist for estimating Y_{pred} uncertainty when Y has not been measured explicitly [13].

To assess overall model stability and predictive ability, we must check how much the TBQ^T values degrade when our observations are used for crossvalidated prediction rather than for calibration (figures 1(e), (f)). Root mean-squared error is less useful here, because it applies to standardized variables and its magnitude is difficult to interpret. The squared Pearson correlation is often used to compare measured data with calibrations or predictions [32]. However, we have found that PLSR models often retain a good Pearson correlation even when the predictions are no longer quantitatively accurate [33]. We thus developed a more-stringent fitness metric [26] that quantifies the extent of one-to-one mapping between measurements and calibrations-predictions:

$$\text{Fitness} = 1 - \frac{\sum_{i=1}^m (Y_{pred,i} - Y_i)^2}{\sum_{i=1}^m (Y_{pred,i})^2 - \frac{\left(\sum_{i=1}^m Y_{pred,i}\right)^2}{m}}$$

When $Y_{pred} = Y$ for all m observations, the fitness is one, with the value falling to below zero with increasing deviations from this line (figure 1(e), (f), green). In our example, we see a reasonable fitness of the calibration (we consider ~ 0.5 to be acceptable), but the performance degrades substantially when crossvalidated predictions are considered. Note in figure 1(f) that the crossvalidated predictions of VPR are still strongly correlated with the measured values ($R = 0.64$), even though the quantitative accuracy is poor. Thus, we conclude from our analysis that this particular PLSR model can be used as a descriptive tool for concisely visualizing patterns in the data, but it should not be used as a predictive tool

for new experiments. Publication-quality PLSR models should be strongly and stably predictive [2, 17–19, 21, 22].

5. Advanced topics and further reading

In this tutorial, we focused on standard approaches for identifying latent dimensions, but there are other more-sophisticated ways of organizing the initial dataset or defining the optimization criterion. One challenge with the flat data-table format is that it struggles to capture the organization of highly systematic datasets [1]. For example, if multiple signaling measurements are collected at various time points per condition, how is the temporal information kept separate from the signaling information for each condition? One workaround is to unfold the time component as additional variables—either directly or as time-derived metrics [17]—and then perform the decomposition. However, the stability of such decompositions can be problematic, and the resulting latent dimensions do not completely reflect the organization of the original dataset.

To address this problem, Bro developed a multiway variant of PLSR that decomposes data cubes or hypercubes in a way akin to principal components [34]. Score vectors are retained, but loading vectors are replaced with weight vectors that capture the latent dimensions associated with each “way” of the dataset:

$$X \approx \sum_{i=1}^n t_i \prod_{j=1}^m w_{i,j}^T$$

where n is the number of latent dimensions in the model, m is the number of ways in the dataset (e.g., $m = 2$ for datasets organized by signaling variables and time variables), and $w_{i,j}$ is the weight vector for the i^{th} latent dimension along the j^{th} way of the dataset. The end result is a more-heavily constrained model that allows a direct mapping to the organization of the starting dataset.

There are also several common variants of PCA, which differ from the standard method in the constraints on the latent dimensions. For example, independent components analysis (ICA) requires latent variables to be minimally statistically dependent rather than orthogonal [35]. The ICA constraint leads to independent components with minimal overlap, in contrast to orthogonal principal components, which are often a weighted blend of the measurement variables. ICA may be valuable in circumstances where collections of measured signaling variables are coordinating distinct functions within the cell.

One simple alteration to PCA is to constrain the elements of the loadings vectors to be positive. This non-negative matrix factorization [36] forces latent dimensions to build upon one another when combined, rather than offset. The non-negativity constraint is useful in fields such as face recognition and linguistics. However, its utility for signal transduction has not been explored, likely because many signaling pathways are known to block the activation of others, undermining the system as a purely additive one.

Other more-elaborate variants of PCA have been developed that incorporate prior biological knowledge. Network component analysis (NCA) uses connectivity information about measured variables to bias the decomposition toward groups of variables with direct interconnections [37]. In doing so, NCA identifies latent dimensions that are supported by network topology and thus hopefully more mechanistic. The drawback is that NCA requires more observations than PCA, and there are particular connectivity restrictions that may not

hold for all networks. However, these types of approaches may become feasible as large-scale systematic measurements of signal transduction become more commonplace [1].

Last, we remind that PCA-PLSR models, like all models, are as valuable when they fail as when they succeed. A recent variant of PLSR, called model breakpoint analysis, embraces this view and allows additional predictions to be extracted from data-driven models [33]. As discussed above, PLSR is fundamentally a linear model, which is valid when there exists a linear mapping between the independent and dependent blocks. By starting with a valid PLSR model, perturbing the data in the independent block with a nonlinear mask applied to each column, and then retraining, one can identify critical failures (breakpoints) where the model abruptly stops making accurate predictions. Model breakpoint analysis delves into these failed models to identify signaling variables whose loadings change coincidentally with the breakpoint. The analysis highlights a small set of variables that might not otherwise be evident in a standard loadings analysis. Model breakpoints and other engineering-inspired “failure analyses” [38] will likely become more important as still-higher dimensional data spaces are reduced to their latent variables.

6. Conclusions

In classical signal processing, electrical engineers work seamlessly between the spatial-time domain and the frequency domain [39]. Although the information is equivalent, sometimes there are clear analytical advantages to working in one domain compared to the other. The same could be argued for the latent-variable space provided by principal components. By performing an eigenvalue-type decomposition of the original data space, latent dimensions capture the variation and covariation that we most often care about and display this information in an intuitive way.

Signal transduction is fundamentally a multivariate process. Now that experimental platforms have developed to embrace this complexity, the statistical approaches introduced here should become even more important. Already, most signaling experiments are multivariate, be it a mass spectrometry experiment or simply a large panel of immunoblots. Fortunately, signaling biochemists can leverage the tools from other data-rich fields [13, 14] that faced the same challenges decades earlier.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

We thank Paul A. Jensen for critically reviewing this manuscript. Work in the Janes lab is supported by the National Institutes of Health Director's New Innovator Award Program (1-DP2-OD006464), the American Cancer Society (120668-RSG-11-047-01-DMC), the Pew Scholars Program in the Biomedical Sciences, and the David and Lucile Packard Foundation. K.J.J. is partly supported by a predoctoral award from the Joanna M. Nicolay Melanoma Foundation.

References

1. Albeck JG, MacBeath G, White FM, Sorger PK, Lauffenburger DA, Gaudet S. Collecting and organizing systematic sets of protein data. *Nat Rev Mol Cell Biol.* 2006; 7:803–812. [PubMed: 17057751]
2. Cosgrove BD, Alexopoulos LG, Hang TC, Hendriks BS, Sorger PK, Griffith LG, Lauffenburger DA. Cytokine-associated drug toxicity in human hepatocytes is associated with signaling network dysregulation. *Mol Biosyst.* 2010; 6:1195–1206. [PubMed: 20361094]

3. Saez-Rodriguez J, Alexopoulos LG, Epperlein J, Samaga R, Lauffenburger DA, Klamt S, Sorger PK. Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction. *Mol Syst Biol.* 2009; 5:331. [PubMed: 19953085]
4. Wolf-Yadlin A, Hautaniemi S, Lauffenburger DA, White FM. Multiple reaction monitoring for robust quantitative proteomic analysis of cellular signaling networks. *Proc Natl Acad Sci U S A.* 2007; 104:5860–5865. [PubMed: 17389395]
5. Bendall SC, Simonds EF, Qiu P, Amir el AD, Krutzik PO, Finck R, Bruggner RV, Melamed R, Trejo A, Ornatsky OI, Balderas RS, Plevritis SK, Sachs K, Pe'er D, Tanner SD, Nolan GP. Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum. *Science.* 2011; 332:687–696. [PubMed: 21551058]
6. Irish JM, Hovland R, Krutzik PO, Perez OD, Bruserud O, Gjertsen BT, Nolan GP. Single cell profiling of potentiated phospho-protein networks in cancer cells. *Cell.* 2004; 118:217–228. [PubMed: 15260991]
7. Janes KA, Yaffe MB. Data-driven modelling of signal-transduction networks. *Nat Rev Mol Cell Biol.* 2006; 7:820–828. [PubMed: 17057752]
8. Catchpoole DR, Kennedy P, Skillicorn DB, Simoff S. The curse of dimensionality: a blessing to personalized medicine. *J Clin Oncol.* 2010; 28:e723–724. author reply e725. [PubMed: 21041706]
9. Huber PJ. Experiences With 3-Dimensional Scatterplots. *J Amer Statistical Assoc.* 1987; 82:448–453.
10. Tufte, ER. *Envisioning Information.* Cheshire, CT: Graphics Press; 1990.
11. Qiu P, Simonds EF, Bendall SC, Gibbs KD Jr, Bruggner RV, Linderman MD, Sachs K, Nolan GP, Plevritis SK. Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE. *Nat Biotechnol.* 2011; 29:886–891. [PubMed: 21964415]
12. Barbano PE, Spivak M, Flajolet M, Nairn AC, Greengard P, Greengard L. A mathematical tool for exploring the dynamics of biological networks. *Proc Natl Acad Sci U S A.* 2007; 104:19169–19174. [PubMed: 18032599]
13. Martens, H.; Martens, M. *Multivariate Analysis of Quality: An Introduction.* Chichester, U.K: John Wiley & Sons; 2001.
14. Geladi P, Kowalski BR. Partial Least-Squares Regression - a Tutorial. *Anal Chim Acta.* 1986; 185:1–17.
15. Alter O, Brown PO, Botstein D. Singular value decomposition for genome-wide expression data processing and modeling. *Proc Natl Acad Sci U S A.* 2000; 97:10101–10106. [PubMed: 10963673]
16. Kuruvilla FG, Park PJ, Schreiber SL. Vector algebra in the analysis of genome-wide expression data. *Genome Biol.* 2002; 3:RESEARCH0011. [PubMed: 11897023]
17. Janes KA, Albeck JG, Gaudet S, Sorger PK, Lauffenburger DA, Yaffe MB. A systems model of signaling identifies a molecular basis set for cytokine-induced apoptosis. *Science.* 2005; 310:1646–1653. [PubMed: 16339439]
18. Kumar N, Wolf-Yadlin A, White FM, Lauffenburger DA. Modeling HER2 effects on cell behavior from mass spectrometry phosphotyrosine data. *PLoS Comput Biol.* 2007; 3:e4. [PubMed: 17206861]
19. Kumar D, Srikanth R, Ahlfors H, Lahesmaa R, Rao KV. Capturing cell-fate decisions from the molecular signatures of a receptor-dependent signaling response. *Mol Syst Biol.* 2007; 3:150. [PubMed: 18059445]
20. Kemp ML, Wille L, Lewis CL, Nicholson LB, Lauffenburger DA. Quantitative network signal combinations downstream of TCR activation can predict IL-2 production response. *J Immunol.* 2007; 178:4984–4992. [PubMed: 17404280]
21. Miller-Jensen K, Janes KA, Brugge JS, Lauffenburger DA. Common effector processing mediates cell-specific responses to stimuli. *Nature.* 2007; 448:604–608. [PubMed: 17637676]
22. Gordus A, Krall JA, Beyer EM, Kaushansky A, Wolf-Yadlin A, Sevecka M, Chang BH, Rush J, MacBeath G. Linear combinations of docking affinities explain quantitative differences in RTK signaling. *Mol Syst Biol.* 2009; 5:235. [PubMed: 19156127]

23. Janes KA, Kelly JR, Gaudet S, Albeck JG, Sorger PK, Lauffenburger DA. Cue-signal-response analysis of TNF-induced apoptosis by partial least squares regression of dynamic multivariate data. *J Comput Biol.* 2004; 11:544–561. [PubMed: 15579231]
24. Garmaroudi FS, Marchant D, Si X, Khalili A, Bashashati A, Wong BW, Tabet A, Ng RT, Murphy K, Luo H, Janes KA, McManus BM. Pairwise network mechanisms in the host signaling response to coxsackievirus B3 infection. *Proc Natl Acad Sci U S A.* 2010; 107:17053–17058. [PubMed: 20833815]
25. Asthagiri AR, Reinhart CA, Horwitz AF, Lauffenburger DA. The role of transient ERK2 signals in fibronectin- and insulin-mediated DNA synthesis. *J Cell Sci.* 2000; 113:4499–4510. [PubMed: 11082043]
26. Gaudet S, Janes KA, Albeck JG, Pace EA, Lauffenburger DA, Sorger PK. A Compendium of Signals and Responses Triggered by Prodeath and Prosurvival Cytokines. *Mol Cell Proteomics.* 2005; 4:1569–1590. [PubMed: 16030008]
27. Janes KA, Albeck JG, Peng LX, Sorger PK, Lauffenburger DA, Yaffe MB. A high-throughput quantitative multiplex kinase assay for monitoring information flow in signaling networks: application to sepsis-apoptosis. *Mol Cell Proteomics.* 2003; 2:463–473. [PubMed: 12832460]
28. Miller-Jensen K, Janes KA, Wong YL, Griffith LG, Lauffenburger DA. Adenoviral vector saturates Akt pro-survival signaling and blocks insulin-mediated rescue of tumor-necrosis-factor-induced apoptosis. *J Cell Sci.* 2006; 119:3788–3798. [PubMed: 16940353]
29. Cohen-Saidon C, Cohen AA, Sigal A, Liron Y, Alon U. Dynamics and variability of ERK2 response to EGF in individual living cells. *Mol Cell.* 2009; 36:885–893. [PubMed: 20005850]
30. Goentoro L, Shoval O, Kirschner MW, Alon U. The incoherent feedforward loop can provide fold-change detection in gene regulation. *Mol Cell.* 2009; 36:894–899. [PubMed: 20005851]
31. Efron, B.; Tibshirani, RJ. *An Introduction to the Bootstrap.* London: Chapman and Hall; 1993.
32. Wold S, Sjostrom M, Eriksson L. PLS-regression: a basic tool of chemometrics. *Chemometrics Intell Lab Syst.* 2001; 58:109–130.
33. Janes KA, Reinhardt HC, Yaffe MB. Cytokine-induced signaling networks prioritize dynamic range over signal strength. *Cell.* 2008; 135:343–354. [PubMed: 18957207]
34. Bro R. Multiway calibration. Multilinear PLS. *J Chemometr.* 1996; 10:47–61.
35. Liebermeister W. Linear modes of gene expression determined by independent component analysis. *Bioinformatics.* 2002; 18:51–60. [PubMed: 11836211]
36. Lee DD, Seung HS. Learning the parts of objects by non-negative matrix factorization. *Nature.* 1999; 401:788–791. [PubMed: 10548103]
37. Liao JC, Boscolo R, Yang YL, Tran LM, Sabatti C, Roychowdhury VP. Network component analysis: reconstruction of regulatory signals in biological systems. *Proc Natl Acad Sci U S A.* 2003; 100:15522–15527. [PubMed: 14673099]
38. Abdi A, Tahoori MB, Emamian ES. Fault diagnosis engineering of digital circuits can identify vulnerable molecules in complex cellular pathways. *Sci Signal.* 2008; 1:ra10. [PubMed: 18941139]
39. Oppenheim, AV.; Willsky, AS.; Hawab, SH. *Signals and Systems.* Upper Saddle River, NJ: Prentice Hall; 1996.

Figure 1.

PCA-PLSR modeling of phosphoprotein signaling data and cell outcomes. (a) Time-integrated phospho-ELISA data and viral progeny release (VPR) for a pairwise inhibitor screen in HL1 cardiomyocytes infected with coxsackievirus B3 [24]. Color maps show the variable-specific z-score across all inhibitor conditions, and asterisks indicate specific observations that illustrate the principal components of the PCA model. VPR is regressed as a function of phosphoprotein data in the PLSR model. (b) Loadings plot for the first two principal components of a PCA model of the phosphoprotein signaling data. Asterisks link principal component loadings to specific observations in figure 1(a). (c) Loadings plot for the first two principal components of PLSR model linking phosphoprotein signaling data to VPR. Note how the loadings have shifted relative to the PCA loadings in figure 1(b). Gray dashed lines indicate a subspace rotation that aligns principal components more clearly with the starting measured variables. (d) Defining the optimal number of PLSR principal components by leave-one-out crossvalidation. Root-mean-squared error of fitted observations (Calibration, solid) are shown in comparison to the error of observations omitted during model training (Prediction, hollow). Note that calibration errors decrease monotonically with increasing principal components, whereas prediction errors show a minimum with two principal components. (e), (f) PLSR models with acceptable calibration fitness may have poor prediction fitness. Measured VPR data are plotted versus the fitted calibration values or the leave-one-out prediction values of the model. Standard errors on the predictions were estimated by jackknifing.