Cell Systems

CellPress

Review

The structure is the message: Preserving experimental context through tensor decomposition

Zhixin Cyrillus Tan[1,*] and Aaron S. Meyer[1,2,3,4,*]
[1]Bioinformatics Interdepartmental Program, University of California, Los Angeles (UCLA), Los Angeles, CA, USA
[2]Department of Bioengineering, UCLA, Los Angeles, CA, USA
[3]Jonsson Comprehensive Cancer Center, UCLA, Los Angeles, CA, USA
[4]Eli and Edythe Broad Center of Regenerative Medicine and Stem Cell Research, UCLA, Los Angeles, CA, USA
*Correspondence: cytan@ucla.edu (Z.C.T.), ameyer@asmlab.org (A.S.M.)
https://doi.org/10.1016/j.cels.2024.07.004

SUMMARY

Recent biological studies have been revolutionized in scale and granularity by multiplex and high-throughput assays. Profiling cell responses across several experimental parameters, such as perturbations, time, and genetic contexts, leads to richer and more generalizable findings. However, these multidimensional datasets necessitate a reevaluation of the conventional methods for their representation and analysis. Traditionally, experimental parameters are merged to flatten the data into a two-dimensional matrix, sacrificing crucial experiment context reflected by the structure. As Marshall McLuhan famously stated, "the medium is the message." In this work, we propose that the experiment structure is the medium in which subsequent analysis is performed, and the optimal choice of data representation must reflect the experiment structure. We review how tensor-structured analyses and decompositions can preserve this information. We contend that tensor methods are poised to become integral to the biomedical data sciences toolkit.

Multiplex and high-throughput assays now enable the exploration of cell responses in unprecedented scale and detail. Consequently, studies of biological systems have increasingly focused on profiling biological systems across multiple contexts (Table 1). For instance, a panel of candidate therapies might be profiled using cell samples derived from multiple organs, with several features of their response measured over time (Figure 1A). Identifying how responses are shared or distinct across multiple cellular contexts and experimental conditions reveals more about the biological mechanism and enhances the generalizability of the results. At the same time, measuring cell lines and tissues across multiple parameters generates data with multiple dimensions (e.g., cell line, time, and experimental conditions), which necessitates reevaluating how we represent and analyze such information.

Representing multivariate data in a tabular form can sacrifice the ultimate insight that can be derived. It is not uncommon that studies with several dimensions are still laid out in rows and columns with some dimensions merged. For the example in Figure 1A, when the experiment is repeated over time, the columns must expand to combine two experimental parameters, drug and time point, such as "alfazumab—1 h," "alfazumab—3 h," "bravociclib—1 h," "bravociclib—3 h," etc. In this format, one may instinctively apply familiar off-the-shelf statistical approaches, such as principal-component analysis (PCA), because the data appears to be in matrix form.

So, what is the problem with this? As communication philosopher Marshall McLuhan famously stated,[12] the medium is the message. The choice of data structure influences its analysis and the subsequent insights. A tabular form implicitly treats each column and row as separated from one another, while merged dimensions diverge from this assumption. For instance, alfazumab—1 h and alfazumab—3 h share the same treatment, and alfazumab—1 h and bravociclib—1 h share the same timing; however, bravociclib—1 h and alfazumab—3 h differ in two distinct ways (Figure 1B). When flattening a multidimensional dataset into a table, we compromise this property.

To devise a more effective approach, the "medium" or structure of the experiment must be incorporated. The example experiment varies across three degrees of freedom: organ, drug, and time; this is best represented by a three-dimensional array or tensor (Figure 1C). A tensor representation aligns entries with shared meaning. For instance, when examined from the perspective of an organ (e.g., thymus and the green cubes), we find the pharmacodynamic profiles of all drugs on this organ; when viewed from a drug (e.g., foxtrotolol and the pink cubes), we find its impact on all organs over time (Figure 1C).

In this work, we aim to provide an overview of how tensor methods have been applied and benefited systems biology studies, and how they can be deployed more broadly. We propose that tensor methods should and will become an established part of the basic biomedical data sciences toolbox.

Defining tensors and tensor decomposition

Tensors are nothing more than multidimensional arrays.[13–15] Zero-, one-, and two-dimensional tensors are scalars, vectors, and matrices, respectively (Figure 1D). To avoid conflicting definitions of "dimension" in linear algebra, "mode" or "order" are used—three-dimensional, three-mode, and third-order tensors are all the same concepts. A matrix has two modes—columns and rows—but tensors over three modes do not have mode-specific names. When structuring biological data into a tensor,

All rights are reserved, including those for text and data mining, AI training, and similar technologies.

Table 1. Some examples of multivariate biological datasets

Ref.	Brief description	Data modality	Contexts
Omberg et al.[1]	gene expression in *S. cerevisiae* cultures	DNA microarray	genes, time points, conditions
Hore et al.[2]	gene expression across multiple human tissues	RNA sequencing	individuals, genes, tissues
Li et al.[3]	metabolite profiles across cancer cell lines in Cancer Cell Line Encyclopedia	liquid chromatography–mass spectrometry	cell lines, metabolites, genes
Jones et al.[4]	synovial fibroblasts cytokine secretion after exposed to drug perturbations	Luminex assay	samples, stimuli, inhibitors
Sikkema et al.[5]	human lung cell atlas	single-cell RNA sequencing	cell types, individuals, gene, anatomical locations
Lloyd-Price et al.[6]	metagenome data in Human Microbiome Projects	metagenomic whole genome shotgun sequencing	subjects, time point, body sites
Gross et al.[7]	protein expression change in human mammary epithelial cell after perturbation	reverse phase protein array	proteins, treatments, time
Durham et al.[8]	roadmap epigenomics data from ENCODE project[9]	various epigenomics data	cell types, assays, genomic positions
Kemper et al.[10]	height and weight-related traits from UK Biobank	physiological data	individuals, traits, time points
Williams et al.[11]	neuron recordings across time and trials in rodents and monkeys	neuronal firing rate	neurons, trials, time

each mode ideally relates to a varied parameter of the experiment, such as samples, genes, cell lines, treatments, concentrations, or time points.

Tensors can be analyzed by tensor decomposition. Before describing this, it is helpful to introduce the concept of rank-one tensors, the building block for tensor decomposition. Like with matrices, even large data tensors can be decomposed into a series of simple patterns, known as rank-one tensors. Unlike the concept of tensor mode, which is directly associated with the data dimensionality, the rank of a tensor is a separate and less evident concept that requires examining its entries. As an illustrative example, consider a smaller dataset with the response of cells from four organs to three drugs over two time points. By stacking the measurements at 1 h (a 4 × 3 matrix) on top of the measurements at 3 h (another 4 × 3 matrix), we obtain a 4 × 3 × 2 tensor with organ, drug, and time modes (Figure 1E). In this contrived example, along the drug mode, every vector is a multiple of (2, 1, 3). This indicates that all eight samples have the same drug-reaction profile. The measurements collected at the 3 h mark are double those at 1 h, suggesting that all measured effects increase to twice the magnitude from 1 to 3 h. The organ factor is (2, 1, 4, 3), indicating the ratio of the four organs' reaction magnitude: cells from the thymus react twice as much as cells from the skin, while cells from the pancreas and liver exhibit effects of four and three times as cells from the skin, respectively. Every entry in this tensor can be precisely computed by multiplying three numbers, each from the organ, drug, and time factor with their positions corresponding to their position in the tensor (Figure 1F). To describe this property, we define this tensor as the outer product of these three vectors (Figure 1G). Tensors that can be expressed as the outer product of a vector set are known as rank-one tensors. The number of vectors within the set is the order of this rank-one tensor; therefore, a rank-one tensor can have any number of modes. Rank-one tensors exhibit a single pattern association with each mode, enabling straightforward interpretation.

Most tensors are more complex than rank-one tensors. Nonetheless, by expressing them as the sum of rank-one tensors (Figure 1H), interpretation becomes significantly easier; they can be understood as the combination of these rank-one individual patterns. Even if we do not represent the original tensor exactly, if a small number of patterns can closely approximate the original tensor and capture essential information, we can still gain insights into the overall trends. This process of breaking down a complex tensor into the sum of a few patterns is known as tensor decomposition or tensor factorization.

A STEP-BY-STEP GUIDE ON TENSOR DECOMPOSITION

Structuring the data into a tensor format

Organizing a dataset into a tensor requires recognizing the structure defined by the experiment. In the example presented in Figure 1C, it is natural to use a three-mode tensor with organ, drug, and time modes.[16] Tensor order can extend beyond three dimensions if, for instance, each organ, drug, and time combination was performed across multiple assays (e.g., measurement of many genes or proteins).

Measurements can only be separated into a distinct mode when the mode's labels relate to a common experimental entity across which the data can be grouped accordingly.[17] For example, should multiple technical replicates for each condition be grouped in a separate mode? No, because the "sample 1" replicate of cells from the liver does not signify the same replicate as "sample 1" of cells from the skin. We may either

CellPress

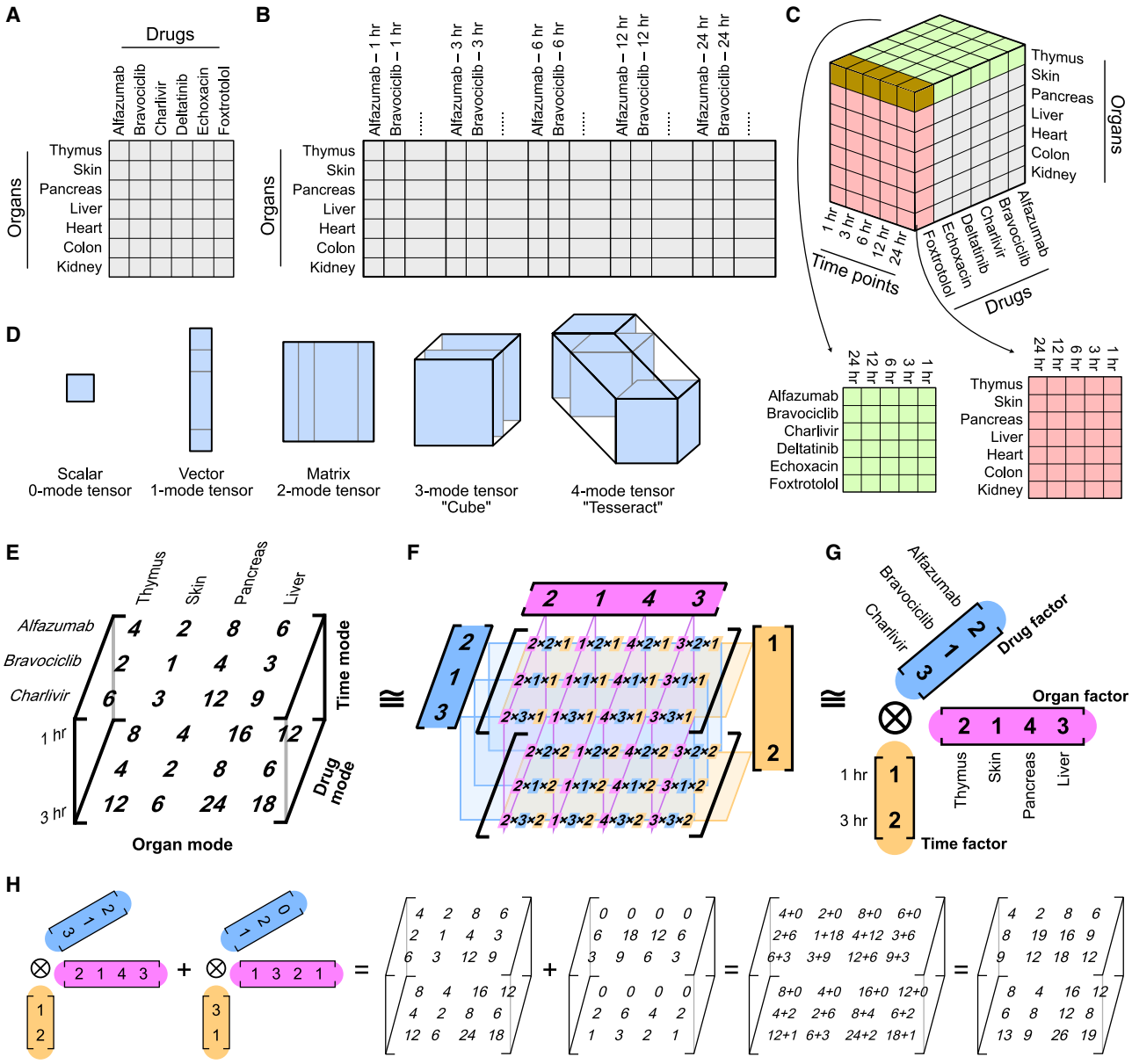

Figure 1. Basic concepts of tensor-structured data

(A) A dataset of cells collected from different organs responding to various drug treatments can be documented by a table.

(B) When the measurements are performed over multiple time points, the original columns in the table can be expanded into multilevel indices, recording both drug and time. This nonetheless breaks the assumption that all columns are equally related.

(C) Alternatively, the same data can be recorded as a three-dimensional array, with organ, drug, and time as three separate degrees of freedom. Here, the pink represents how every cell responds to foxtrotolol over time, and the green represents the pharmacokinetic profile of cells from the thymus over all treatments. The brown is shared by pink and green.

(D) Tensors are multidimensional arrays. A dimension of a tensor is a mode. 0, 1, and 2-mode arrays are known as scalar, vector, and matrix.

(E) An example of a rank-one tensor. A subset of the drug response dataset on cells from four organs responding to three drugs over two time points and has dimensions 4 × 3 × 2, organ by drug by time.

(F) Rank-one tensors are those whose every entry can be written as the product of a few numbers, one from each mode-specific vector, from their corresponding coordinates.

(G) A rank-one tensor can be written as multiple mode-specific factors joined by the vector outer product, ⊗.

(H) Even written as sets of vectors, these rank-one tensors should still be understood as arrays with numbers in every entry. Adding two tensors of the same shape is to add their corresponding positions together.

average these replicates during reformatting if their variation is not of particular interest or apply resampling strategies to preserve replicate variances.[18] However, if these samples represent a common set of patients—sample 1 is the same for all cell types, indicating that they came from the same individual—this justifies the inclusion of a corresponding mode. Similarly, single-cell measurements from different samples inherently come from different cells; therefore, single cells cannot

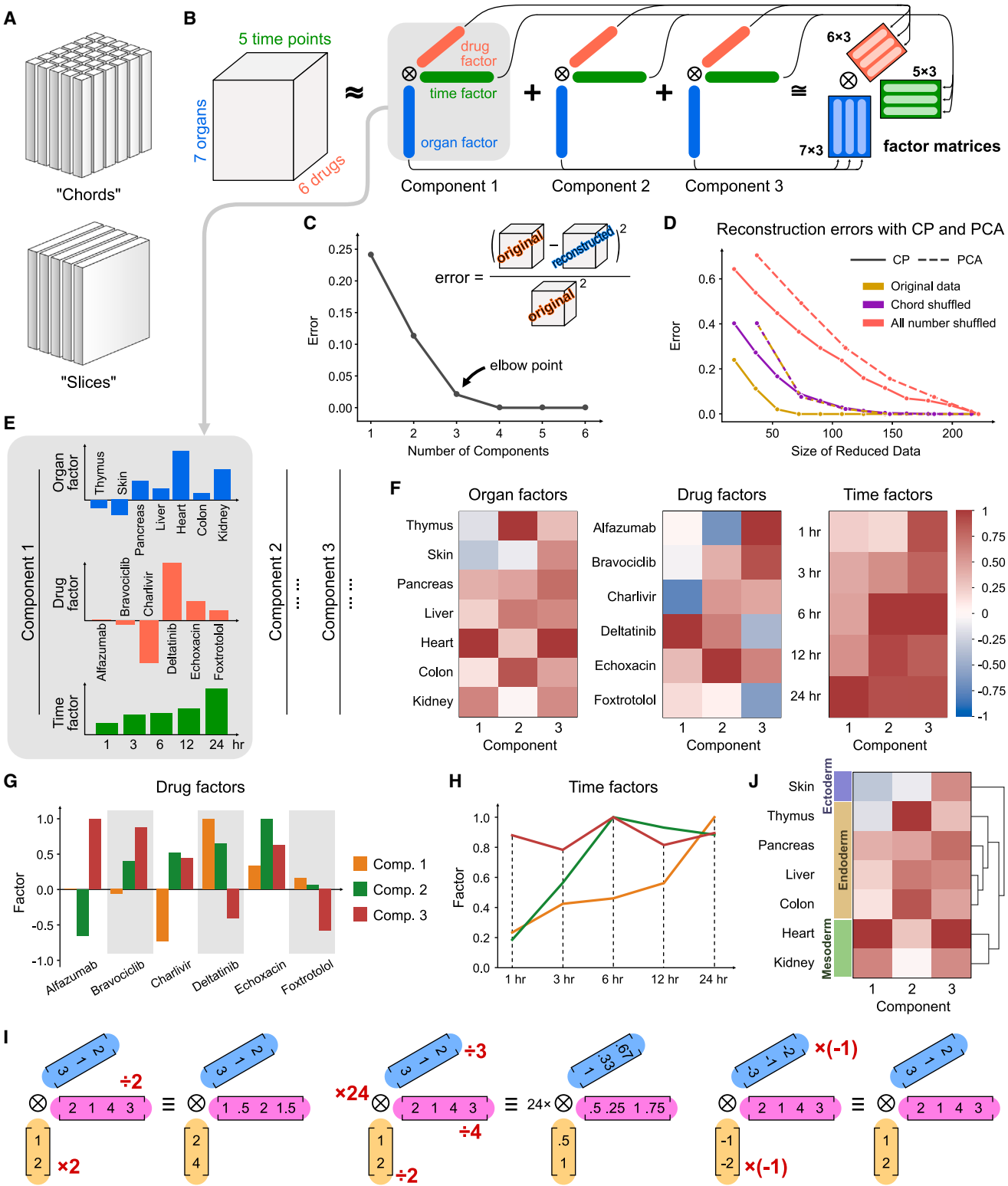

Figure 2. Fundamentals of CP tensor decomposition

(A) For a three-dimensional array, a chord is the entries across all labels on one single mode, and slices are entries across two modes.

(B) Canonical polyadic (CP) decomposition approximates a complicated tensor as the sum of a few rank-one tensors. In the example here, for a drug response tensor of 7 × 6 × 5, organ by drug by time, after being decomposed into 3 components, we will have 3 factors for each of the three modes. Organizing them into matrices, we will have three-factor matrices with shapes 7 × 3, 6 × 3, and 5 × 3 for organ, drug, and time, respectively.

(legend continued on next page)

form a distinct tensor mode when attempting to parallelize samples. They may only form a tensor mode when multi-omic assays are performed on identical cells. As another example, in single-cell analysis, when combining runs from different backgrounds, whether the clusters with the same label should be aligned depends on whether these cluster labels hold the same meaning across backgrounds. If the cluster labels are assigned randomly (e.g., in *k*-means), they are not equivalent between runs and therefore cannot form a separate cluster mode. However, if the clusters can be identified based on cell surface markers and cluster 1 consistently represents the same cell type, this cell type mode is justified.

In a tensor format, the items representing positions along a mode are treated separately. Therefore, the order of items on a mode is inconsequential to the tensor decomposition. For instance, switching the positions of 3 and 12 h on the time mode in the tensor in Figure 1C does not affect its decomposition results. For longitudinal measurements where sometimes the time points cannot be aligned perfectly, compromises may have to be made. One approach can be binning, where similar time points of different samples are grouped into one category. For instance, if one individual only has samples at 3 h, while another only has 4 h, a binned "3–4 h" category may be created to align them. Sometimes, several positions in the tensor may be left empty to maintain the data's logical structure (see "missing data and imputation" on decomposition with missing entries).

Tensor decomposition can benefit from appropriate data preprocessing, such as centering, scaling, and transformation. Centering and scaling operations are always associated with a specific mode, so they become more complex when data has multiple modes.[19] For a three-mode tensor, chords are an extension of columns in a matrix, whereas slices are all values associated with a specific position along one mode (Figure 2A). For example, chord-wise and slice-wise operations across organ mode, respectively, correspond to one type of measurement across all organs, and all numbers aligned to one organ. Centering is performed to make the data ratio-scaled, as tensor decomposition models assume. This means that a zero value indicates a true zero effect, making multiplications meaningful (i.e., doubling the number always equals twice the effect). Scaling is used to adjust the scale differences among variables to avoid larger values overshadowing the variation of interest, which also helps maintain numerical stability during solving. A common preprocessing choice is to mean-center across the subject/sample mode and then scale the standard deviation to one within other modes. Transformation is another technique that usually applies to nonlinear data to ensure the measurements are ratio-scaled before the decomposition.

Performing the decomposition

The decomposition method we review here is known as canonical polyadic (CP), parallel factor (PARAFAC), or canonical decomposition (CANDECOMP). Implementations of this method are available in software packages for various programming languages (Table 2).

CP decomposition requires a data tensor and the desired number of components. The component number is the number of rank-one tensors used to approximate the original data (Figure 2B). For each mode, the factors of each component can be regrouped into a factor matrix (Figure 2B, right), in which the first columns of each matrix represent the first factor, the second columns the second factor, and so on. Thus, if we take the outer product of the first columns (factor 1) in the three-factor matrices, we will obtain the first decomposed rank-one tensor, component 1. Repeating this process for each component and summing them up, we can reconstruct a tensor that approximates the original data (Figure 2B). To summarize, the decomposed factors can be either grouped by mode into factor matrices or by the factor indices into components. The goal of the decomposition algorithm is to make the reconstructed tensor match the original one as closely as possible.

The number of components

With CP decomposition, one must choose the number of components. Too few components will miss essential trends, while too many will lead to redundant factors, noise (overfitting), and poorer interpretability.

To quantify how well a decomposition with the chosen number of components fairly represents the original data, one can quantify the difference between the reconstructed tensor and the original data, or the reconstruction error. This value is calculated as the sum of squared differences between these two tensors, usually normalized by the sum of squares of the original data (Figure 2C). Smaller errors indicate a better fit. While the error can range from 0 to any positive number, a successful fit should result in an error below 1 when normalized. The reconstruction error consistently decreases with a greater number of components with diminishing returns, where each additional component improves the fit to a lesser degree (Figure 2C). Achieving a perfect fit to the data is typically not the goal of tensor

(C) Plotting the number of components against the error. Error is defined as the sum of squared differences normalized by the sum of squares of the original tensor. An optimal component number may be attained at the elbow point on the plot (in this example, 3 components), or the point at which an acceptable error is reached.
(D) Sizes of reduced data plotted against their reconstruction errors using CP or PCA. CP decomposition represents the original dataset more concisely than PCA, whereas the chord-shuffled and all-number-shuffled versions reduce the advantage of CP, indicating the underlying data structure influences data compression.
(E) Plotting every factor separately to visualize tensor decomposition results. Here, three bar plots demonstrate the three-mode-specific factors of component 1. The factors of other components are omitted but can be shown similarly.
(F) Heatmaps to visualize the factor matrices compactly. We can both inspect all factors of a component across modes for its interpretation and/or compare factors within a mode to distinguish their differences.
(G) Factors of a discrete variable mode (such as drug mode here) can be visualized with a bar plot.
(H) Factors of a continuous variable mode (such as time point mode here) can be visualized with a line plot.
(I) Demonstration of factors' scale indeterminacy. Scaling the factors coordinately (left), factoring the weights to a separate scalar (middle), or negating factors in pairs (right) all yield equivalent factorizations, as they all reconstruct to the identical rank-one tensor.
(J) Organ factor heatmap reordered by hierarchical clustering on the factorization results. Other information, like the organs' biological grouping, can be labeled next to the heatmap to identify their association with the factors.

Table 2. Selected tensor decomposition packages and the methods they implement

Programming language	Package	Decomposition methods: Reviewed in this work	Decomposition methods: Other methods	Constraints implemented
Python	TensorLy[20]	CP, Tucker, PARAFAC2, CMTF, CP partial least squares	partial Tucker, tensor train, CP/Tucker regression	nonnegativity, symmetry, regularization
MATLAB	Tensor Toolbox[21]	CP, Tucker	N/A	symmetry, sparsity, orthogonality
R	rTensor[22]	CP, Tucker	3-mode tensor SVD, multilinear PCA	N/A
	multiway[23]	CP, Tucker, PARAFAC2	simultaneous component analysis	nonnegativity

CP, canonical polyadic decomposition (also called PARAFAC or CANDECOMP); CMTF, coupled matrix-tensor factorization; SVD, singular value decomposition; PCA, principal-component analysis. Tucker decomposition here can be higher-order SVD (HOSVD), truncated HOSVD, or higher-order orthogonal iteration.

decomposition. While this is technically feasible by setting the component number equal to the tensor's theoretical rank,[14] in practice, this number is almost always too high for any practical use. To choose the optimal number of components, one may identify where the benefit of adding more components diminishes. This sometimes corresponds to the kink (or elbow) point on the error plot. However, such a transition point is not always evident.

The process described above resembles selecting component numbers in PCA, but with a few distinctions. Tensor decomposition is not a recursive process: the components of a 3-component decomposition are not necessarily a subset of the 4-component decomposition. On that account, one must experiment with every candidate component number to identify an optimal choice. Components are also not guaranteed to be ordered.[24] Therefore, to create an error plot, the decomposition must be run for each number of candidate components (Figure 2C).

The choice of component number directly relates to the data compression efficiency and fidelity trade-off (Figure 2D). Since a tensor can be approximated with its factorization results, which consist of fewer numbers, tensor factorization effectively compresses it. The smaller the size of the reduced data, the better the data compression ratio. However, this comes with the cost of a worse approximation (i.e., a larger reconstruction error) of the original data. For example, for a tensor with $7 \times 6 \times 5 = 210$ values, a 4-component decomposition will compress it down to $4 \times (7 + 6 + 5) = 72$ numbers; if using 3 components, only $3 \times (7 + 6 + 5) = 54$ numbers, but with a greater reconstruction error.

The reconstruction error also depends on whether the underlying data structure can be well-approximated by a low-rank model. As illustrated in Figure 2D, CP decomposition can represent the drug response dataset more concisely than PCA—achieving a smaller representation under the same fidelity or comparable size with lower error—since its underlying structure can be approximated well by the sum of multiple rank-one tensors. However, shuffling the chords in the data disrupts this low-rank tensor structure, causing the performance of CP to deteriorate. Further shuffling all numbers in the dataset completely eliminates the underlying low-rank matrix structure, thereby degrading even PCA's performance.

Another consideration is the inherent noise present in biological measurements. With too many components, tensor decomposition starts to fit trivial patterns, which are more likely to be noise. In principle, one should cease adding more components when the algorithm begins to overfit (fit the original data too closely but lose generality), is prone to excessive local minima, or starts to violate the properties of CP. These situations may be assessed respectively through imputation tests (see "missing data and imputation"), factor similarity tests (see "optimization algorithms"), or core consistency diagnostics (see "tucker decomposition: allowing all factors to interact").

Visualizing and interpreting the results

After validating the decomposition, the resulting factors can be inspected for biological insight. To provide a concrete example, we inspect our decomposition results shown in plots (Figures 2E–2H).

To visualize the results, one should design plots that describe how each factor is associated with the labels along each mode. Therefore, one can have one subplot for each factor (one from each mode) for each component, repeated for all components (Figure 2E). In these plots, the x axis indicates the labels, and the y axis shows the factor weights. For a more concise visualization, one can also plot each factor matrix made from factors from all components as a heatmap with colors representing the weights (Figure 2F). To visualize factors within a specific mode, bar plots and point plots generally work well for discrete labels such as samples, cell lines, or molecules (Figure 2G), while line plots are more suitable for continuous labels such as time or concentration (Figure 2H). Overall, visualization should optimally serve to present one's insights; there is no fixed rule.

The initial phase of interpretation involves delineating the meaning of each component pattern. This requires reading the plots across all modes. For instance, consider component 1 (Figure 2E). Within the organ factor, the largest signal originates from cells collected from the heart, followed by smaller weights from the cells of the kidney. The same information can also be captured from the first column of the organ factor heatmap (Figure 2F, left). Along the drug mode, the strongest signals appear on deltatinib in the positive direction and on charlivir in the negative. This can be read out from the heatmap (Figure 2F, middle) or the drug factor bar plot (Figure 2G) too.

The time mode factor, on the other hand, has increasing values over time in component 1. The orange line in the time factor plot best represents this trend (Figure 2H). Putting this information together, one concludes that component 1 mostly delineates a temporally increasing impact of deltatinib, positively, and charlivir, negatively, on cells from the heart (then the kidney). In practice, one can choose whichever plot best depicts the trend. Following the same logic, we see that component 2 unveils an effect of mostly echoxacin, positively, and alfazumab, negatively, on cells collected from the thymus, peaking at 6 h. Component 3 mostly indicates an effect of alfazumab and bravociclib on cells from mostly the heart that persists over time.

A specific mathematical intricacy, scale indeterminacy, can hinder clarity (Figure 2I). As the effect along each mode is multiplied together, scaling these factors in an opposing way, i.e., doubling one factor and halving another within a component, yields equivalent results (Figure 2I, left). This indicates that only the relative ratios of weights within a factor are certain, not the absolute values. Therefore, we should not compare the absolute weights between factors of different components, only the relative composition. To avoid ambiguity, one typically normalizes all factors to a defined scale, storing the weighting as a separate scalar (Figure 2I, middle). The issue of indeterminacy extends to negative factors: by the same logic, negating two factors simultaneously also yields equivalent results (Figure 2I, right). This is sometimes called sign indeterminacy.[25] One approach to avoid ambiguity is to make most modes positive by negating the factor vectors in pairs, ensuring that at most only one mode harbors factors with an overall negative effect (Figure 2I, right).

One can also compare across components within a single mode. Within the organ mode, for instance, components 1 and 3 assign similar factors to cells from the heart and kidney, unveiling shared localization in drug effect (Figure 2F, left). In the drug mode, each drug has different factors, suggesting that they have divergent interaction profiles (Figures 2F, middle, and 2G). Each time factor also has a distinct trend, ranging from stable (component 3) to increasing over time (component 1) and peaking (component 2) (Figure 2H). To better identify similar entries (e.g., drugs or organs) on a mode discovered by tensor decomposition, one can also perform hierarchical clustering on the factor matrix and reorder the entries accordingly (Figure 2J). This juxtaposes entries of similar factor weights, helping to reveal groupings of comparable entries. Additional contextual information, such as cell categories, sample classes, and patient statuses, can also be labeled next to the heatmap to help identify associations between the factors and their known groupings (Figure 2J).

DETAILS AND CONSIDERATIONS OF TENSOR DECOMPOSITION

The previous section presented an overview of employing tensor decomposition. However, several details of the fitting procedure may help in certain circumstances.

Optimization algorithms

Solving tensor decomposition is, in its essence, an optimization problem. The objective is to find a set of factor matrices that, when multiplied, render a reconstructed tensor with minimal error (Figure 3A). Common mathematical optimization algorithms, such as gradient descent or the Newton-Raphson method, can be employed here.[26] This "direct optimization" approach offers the advantage of versatility since many optimization methods allow additional constraints, making it possible to develop new decomposition schemes. However, its performance relies heavily on the chosen method and initialization values, since a substantial number of parameters must be simultaneously solved.

As an alternative approach, we can first notice that the factor matrices exhibit symmetry: swapping mode orders does not change the solving. Also, if we know the correct factors of all other modes, solving for one mode can be converted into an ordinary least squares problem. Thus, we can tackle one mode at a time using least squares while treating the others as constant, then repeat this for every mode (Figure 3B). We keep iterating until these factors converge. Over time, we can expect a monotonic decrease in the reconstruction error. This approach is called alternating least squares (ALS). Besides its efficiency, ALS often benefits from more stable and reproducible performance.[27]

Both methods require initial factor values. While a random initialization may be sufficient, a more informed estimation can expedite convergence. One such estimation involves using the principal components from a flattened version of the original tensor. This approach, known as singular value decomposition (SVD) initialization, usually yields more stable results and reduces the likelihood of a suboptimal solution (i.e., local minimum). However, neither initialization guarantees an optimal solution.

When the resulting factors are highly dependent on the starting point of the fitting, it can indicate that the optimization problem is ill-formed, suggesting that the chosen number of components is too large or that additional constraints would be helpful. The factor similarity test exploits this property to determine the appropriate component number.[11,28] In essence, this test quantifies to what extent different starting points change the resulting factors, helping determine up to how many components the factorization algorithm remains stable.

Missing data and imputation

Missing measurements frequently arise from experimental limitations. These omissions are not necessarily a result of oversight; certain measurements may be intentionally missing. This issue becomes particularly pronounced with tensors, as complete tensors require all possible combinations of all modes. Consequently, missing data can emerge simply from transforming a dataset into a tensor, even if the original data appears complete (Figure 3C). For instance, the example dataset in Figure 3C does not contain any missing values, but because the impacts of deltatinib and echoxacin after 6 h were not measured, the reformatted tensor contains missing chords (Figure 3C, right).

Tensor decomposition can be performed even with missing values in a tensor. This can be achieved either by ignoring the missing positions and only fitting the existing ones in direct optimization, or prefilling them with placeholders in the hope of updating these values iteratively through repeated factorization and reconstruction.[29,30] Note that zeros in a tensor will still be fit by the tensor decomposition algorithm, unlike explicitly missing values, so replacing missing values with zeros is

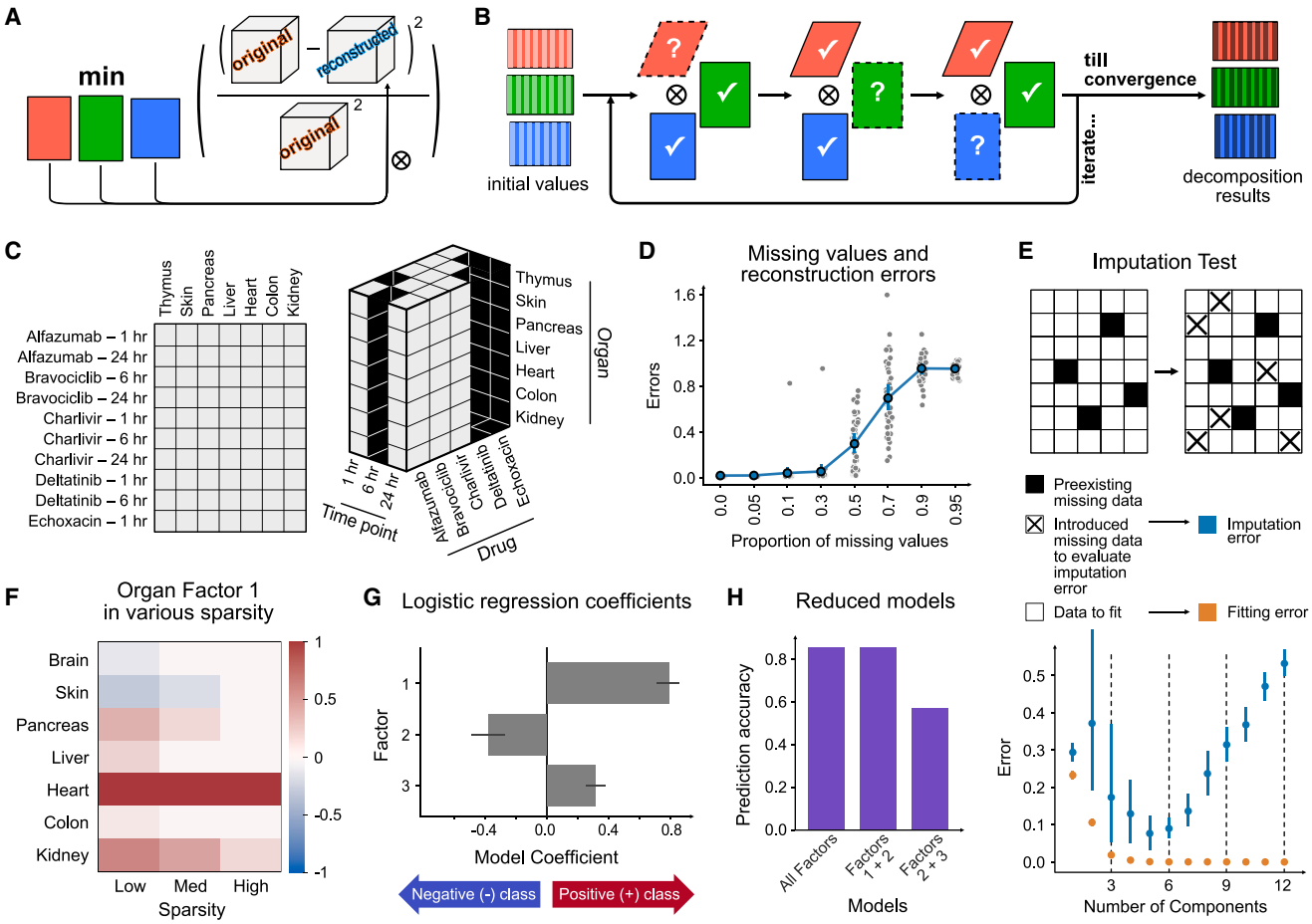

Figure 3. Technical details on applying CP to biological data

(A) Solving tensor decomposition is an optimization problem aiming to minimize the reconstruction error by adjusting the numbers in the factor matrices.

(B) Alternating least squares (ALSs) is another strategy besides direct optimization. Starting from a set of initial values, it optimizes one factor matrix at a time with linear least squares while holding the others constant. This process is repeated on each factor matrix until convergence is reached.

(C) A demonstration of how structuring data into tensor format may create missing values. Although the original table on the left does not contain any missing values, since not all drug-time pairs are measured, the reformatted three-mode tensor contains missing chords.

(D) Various proportions of missing data were introduced to the tensor to evaluate how missingness impacts reconstruction errors. Each gray point represents one of 40 runs with random missing patterns. The blue points and error bars show the average reconstruction errors and 95% confidence intervals, respectively.

(E) Demonstration of the imputation test. Ignoring the preexisting missing data, we arbitrarily introduce more missing positions, use the remaining data to fit the decomposition, and then compare the reconstructed (i.e., imputed) values with the original values at the positions we removed. Plotting against the number of components, the fitting errors should decrease monotonically with more components. However, the imputation error will eventually increase with excessive components due to overfitting. Error bars represent 95% confidence intervals from 30 independent runs after masking 30% of the data for the imputation test in the demonstration dataset.

(F) Sparsity in tensor factors. These organ factors 1 are in the order of increasing sparsity.

(G) Tensor decomposition factors can be used for response prediction when combined with regression. The coefficient of each factor indicates their association with the sample classes. Error bars represent the jackknife confidence intervals.

(H) For classification, the model may be reduced to using a subset of the factors.

incorrect. While the fitting process may still function even with a high proportion of missing values, the resulting factors can significantly deviate from those obtained with complete data. The extent of this deviation can vary widely depending on the underlying data structure and the specific missingness, but generally, a greater portion of missing data leads to larger reconstruction errors (Figure 3D).

Tensor decomposition also provides an avenue to impute the missing values of a tensor. Since a full tensor can be reconstructed from the resulting decomposed factors (Figures 1G and 2B), one can use these reconstructed values from tensor decomposition to replace the missing positions, effectively imputing them.[30] Compared with matrices, higher-order tensors benefit from the additional information from more shared coordinates. Tensor imputation through decomposition is not foolproof; it remains an area of ongoing research. Like matrix completion, it relies on inherent assumptions. If the original data cannot be approximated as lower-rank tensors (Figure 2B), the imputed values can significantly deviate from their true values. Other factors, such as the quantity and distribution of the missing values and the chosen component numbers and decomposition method, can also influence the accuracy of imputation. A tensor cannot be missing all its values across a slice. Thus, in situations where there are very few non-missing

values, it may be advantageous to consider discarding one position along a mode.

One can use imputative performance to assess the reliability of decomposition on a tensor or to determine its appropriate number of components. In an imputation test, one intentionally introduces additional missing values into the data (Figure 3E, top). Following decomposition, the entire tensor is reconstructed from the factors, and the left-out values are compared against their reconstructed versions. A substantial disparity indicated by a high imputation error indicates an unsuccessful decomposition, attributable to either an ill-suited dataset or an excessively high number of components. While the fitting error monotonically decreases with more components, the imputation error often shows an optimum at an intermediate number of components (Figure 3E, bottom).

Constraints on the factors

The optimization processes reviewed so far only aim to best fit the data. However, their results may suffer from low interpretability, overfitting, and instability. Numerical constraints on the factors can help with these issues. Although they may impact the goodness of fit, reasonable constraints can enhance the model's ability to reveal meaningful patterns, leading to more insightful discoveries. For example, one goal of constraints is to achieve sparsity, where a factor has nonzero values in only a few positions and renders others nearly or exactly zeros. This helps establish direct associations between factors and their effects.[2] For instance, in the hypothetical organ factors 1 in Figure 3F, the low-sparsity factor has weights on almost all organs, making interpretation more complicated. The high-sparsity version only has weights on the heart and the kidney, better indicating that this factor has the greatest association with these two organs. Regularization is commonly used to achieve sparsity in factors.

Nonnegativity is the most commonly used tensor decomposition constraint.[31] It aligns intuitively with the expectation that certain quantities in biology are inherently nonnegative: a cell cannot secrete a negative number of molecules, and a gene cannot be expressed at a negative level. Nonetheless, enforcing nonnegative factors may limit the tensor factors from modeling negative effects in biology, such as an upstream pathway that suppresses molecule secretion or inhibits gene expression. Another rationale for the nonnegativity constraint is to foster sparsity within the factors and avoid overfitting. Decompositions allowing negative factor values can yield degenerate components, where one component is strongly positive and another is strongly negative, mostly canceling each other out.[24] Enforcing all values in the factor matrices to be nonnegative obviates such occurrences, as the impact of any component cannot be counteracted by another. Nonnegative factorization often leads to minimal sacrifices in model error, solidifying its application in practice.[11]

Constraints can also be used to enforce biological knowledge in a decomposition.[32] For instance, in neuroscience, one may postulate minimal crosstalk among different brain regions and limit the brain region factors to be a diagonal matrix.[33] In molecular biology, one may employ orthogonalization of the factors to enforce a clean delineation between components and traits.[34] This usually lacks a standardized approach, as biological contexts vary and may require customized solving.[28]

Subsequent analysis

While tensor analysis often serves as an important step for distilling data into significant patterns, further analysis beyond the factor plots (Figures 2E–2H) is often required to learn what component patterns indicate about biology. The factor matrices serve as efficient summaries for individual patterns linked to their respective modes. Consequently, each matrix can be isolated for a detailed analysis of the variation within a specific mode of interest. For instance, the components associated with genes or molecules of particular interest from prior knowledge can be further examined to validate their agreement with known mechanisms.

The decomposed factors can also be used as reduced data to predict responses or sample classes when combined with regression. The scale and sign of the weights for each factor indicate its effect on the regressed quantity (Figure 3G). If only a subset of factors contributes to the effect of interest or the regression model can achieve comparable accuracy with fewer factors, the prediction model may use only a subset of them (Figure 3H). For example, in Figure 3H, prediction using only two factors, factors 1 and 2, performs just as effectively as all factors.

ADVANCED TENSOR METHODS BEYOND CP

In this section, we cover more advanced tensor decomposition methods. For more complex biological data, it is particularly crucial to choose a method that best reflects the structure of the expected patterns.

Tucker decomposition: Allowing all factors to interact

In CP decomposition, especially when there are more components, some factors may start to look similar within one mode. For example, in Figure 2F, the organ factors 1 and 3 appear similar. This redundancy arises from the inherent constraint of CP decomposition, where factors may only interact within the same component (Figure 2B). In other words, because CP does not allow interaction between drug factor 1 and time factor 3, a repetitive drug factor must be present in component 3 to capture a similar effect on organs. CP permits the existence of two identical factors in one mode, as long as their corresponding factors in other modes remain distinct. Therefore, the factors along a mode in CP decomposition may not succinctly summarize the trends in this mode.

Tucker decomposition is a different tensor decomposition model from CP with a more flexible construct.[35,36] It permits varying numbers of factors for each mode, and all factors across modes interact. For example, here we perform a (4,3,2)-rank Tucker decomposition on the $7 \times 6 \times 5$ drug response data tensor, in which the organ mode has 4 distinct factors, the drug mode 3, and the time mode 2 (Figure 4A). Consequently, there are $4 \times 3 \times 2 = 24$ factor interactions. Each interaction can be understood as a component in CP (Figure 4A, right). The magnitude of each interaction is characterized by its corresponding weight, and these 24 weights can be arranged into a $4 \times 3 \times 2$ core tensor (Figure 4B, left). The outcomes of a Tucker decomposition include a core tensor that models the factor interactions and three-factor matrices that represent the principal

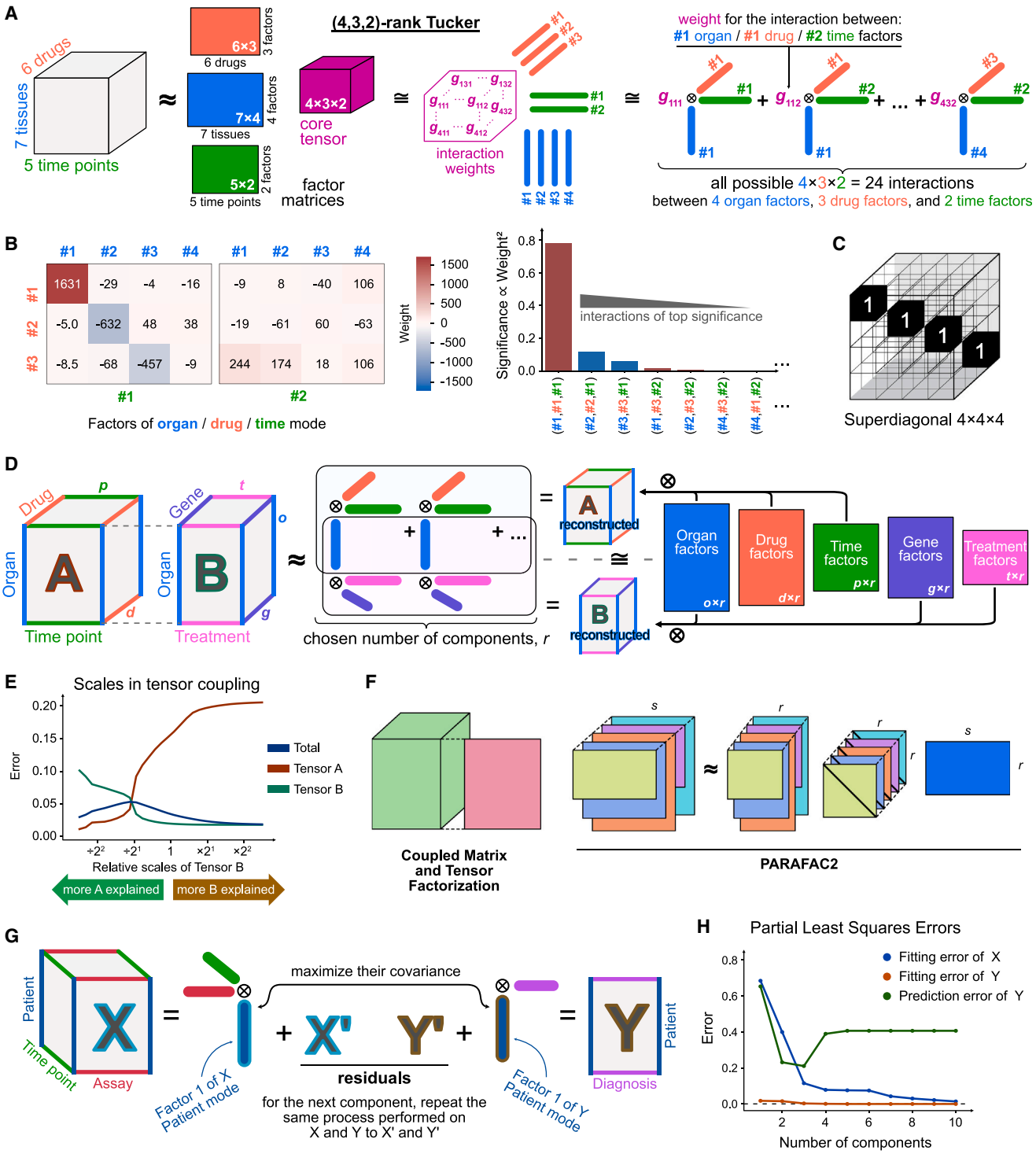

Figure 4. Tensor methods beyond CP: Tucker decomposition, coupling, and partial least squares

(A) Schematic of Tucker decomposition. This (4,3,2)-rank Tucker decomposition on the previous 7 × 6 × 5 drug response tensor allows all distinct 4 organ factors, 3 drug factors, and 2 time factors to interact. The weights of these 24 interactions are organized into a 4 × 3 × 2 core tensor. The results of a Tucker decomposition are a factor matrix for each mode and this core tensor.

(B) The core tensor of a Tucker decomposition. It can be visualized by showing the numbers in each slice. The significance of an interaction is proportional to its weight squared.

(C) A superdiagonal 4 × 4 × 4 tensor. CP is a special case of Tucker decomposition where the core tensor is superdiagonal.

(legend continued on next page)

components (the major trends) along those three modes (Figure 4A, middle).[14]

There are two ways to utilize the Tucker results. The factor matrices, consisting of the eigenvectors defined in higher dimensions, can be used as summaries of the modes. They can be visualized similarly to Figure 2F. One can also analyze the core tensor to identify the top interactions by significance, which is defined as proportional to their weights squared (Figure 4B, right). For example, here, ~78% of the variance can be explained by the interactions among factor 1s of organ, drug, and time.

CP decomposition is equivalent to a specific instance of the Tucker decomposition, wherein factors associated with different components are non-interacting, thus the core tensor assumes a superdiagonal form, signifying that all off-diagonal positions are zeros (Figure 4C). This superdiagonal property has been harnessed to test whether a CP decomposition is correctly implemented, known as core consistency diagnostic.[37] Specifically, after acquiring the CP factor matrices, if adding off-superdiagonal interactions to the retrofitted core tensor can improve the fitting considerably, the number of components may be inappropriate, or Tucker may be a better model than CP for this dataset.

Tucker decomposition offers a better mode-specific summary and more flexible analysis, which opens many possibilities for method development.[38] Many variants of Tucker decomposition, including higher-order SVD (HOSVD), have been applied to biological datasets.[1,39]

Coupling: Sharing factors across multiple tensors

The integration of (epi-)genomic, transcriptomic, and proteomic data, either in bulk or at the single-cell resolution, has provided opportunities for an integrated understanding of cellular processes. More broadly, biologists often encounter data fusion challenges when attempting to identify shared patterns among multiple data sources.[40] The joint analysis of several datasets can be formulated as coupling of tensors.[41]

Coupling arises when two or more datasets are collected with differing dimensions, but all tensors share at least one "coupled" mode (Figure 4D, left). Commonly coupled modes include samples or patients that are shared across multiple assays. For instance, there may be another dataset on cells from the same groups of organs measured in the previous dataset (Figure 4D, tensor A); this new dataset contains the gene expression of the cells from these organs under various treatments (Figure 4D, tensor B). In this case, the organ mode is shared, while each tensor has other uncoupled modes, such as drugs and genes. In a coupled decomposition, a shared mode will have a common factor matrix that is used by all tensors that comprise this mode (Figure 4D, right). In this way, this factor matrix succinctly reflects the trends across these two coupled tensors.

Visualizing and interpreting the results of a coupled tensor factorization operate like with CP (Figure 2F). Each tensor is decomposed into a series of rank-one components, and any coupled mode will have a single set of factors shared among all the tensors using it (Figure 4D, middle). All other modes will still have their own factor matrices (Figure 4D, right). In addition to examining components within a tensor, one can also compare the uncoupled private modes between two tensors to assess their associations. A unique advantage of coupling arises from missing data. If a certain tensor has missing entries, other tensors can share information through the coupled factors to improve imputation.

Coupling introduces a new issue. Because factorization minimizes the overall reconstruction error, the relative scaling among coupled tensors influences the priority in explaining patterns from each dataset. As the total variance of values can differ significantly across datasets collected from various assays, the decomposed factors can be dominated by one source if the data is not appropriately scaled. Typically, a range of scaling should be explored, and the overall and tensor-specific errors evaluated (Figure 4E). If the factor matrices are used to predict some outcomes, the prediction accuracy can also be used to compare various scalings and determine an optimal scaling.

Overall, coupling offers remarkable flexibility for data integration. Although we refer to the methods as coupled tensor factorization, matrices (2-way tensors) are also included. For example, many applications have used coupled matrix and tensor factorization to jointly analyze a tensor and a matrix (Figure 4F, left).[42,43] Coupling also expands the applicability of tensor methods to more irregularly shaped data, as illustrated by PARAFAC2.[44] PARAFAC2 is a method that decomposes a series of matrices, where one mode is shared while another is unaligned and variable in size (Figure 4F, right). This forms a ragged tensor to which CP or Tucker cannot be applied. PARAFAC2 projects the variable modes into a latent, uniform shared mode, identifying patterns not only on the shared mode but also across these matrices, effectively harnessing the benefits of coupling. Tensor coupling is an active field of method development, including combining it with other decomposition strategies (such as Tucker or partial least squares [PLS]).

Partial least squares: Informing decomposition by effects

Many scientific questions involve identifying how a series of measurements associate with a specific phenotype or outcome of interest. For example, one might associate patients' blood

(D) Schematic of coupled tensor decomposition. Here, two three-mode tensors, A and B, are coupled on the organ mode. Therefore, the organ factors are shared, while the drug and time factors are private to tensor A, and gene and treatment factors tensor B. The dimensionalities of them are indicated by the lowercase letters.

(E) The scaling issue in coupled tensor decomposition. When one of the coupled tensors has values with greater total variance, the factorization explains more variance in it if without proper scaling, leading to an uneven representation of the two datasets.

(F) Some other examples of tensor coupling: coupled matrix and tensor factorization (CMTF, left) and PARAFAC2 (right).

(G) Schematics of tensor partial least squares. Partial least squares is performed on two tensors, X and Y, with one aligned mode. During solving, the two separated X and Y factors of the aligned mode (patient mode in the example case) yield the maximal correlations. Partial least squares components are solved sequentially, as the next component is found by repeating the same process on the residuals, X′ and Y′, from the last round. Therefore, the components are in decreasing order of covariance explained.

(H) The performance of partial least squares can be evaluated by calculating the fitting errors of X and Y and the prediction errors of Y. Both fitting errors should decrease with more components, while the prediction errors (from cross-validation) of Y should initially decrease but eventually increase due to overfitting.

panel tensor with their diagnosis. In statistical terms, we have explanatory variables (X) and outcomes (Y), and our goal is to reveal only the patterns in X that uniquely associate with Y. This approach differs from simply coupling them where the joint variance of both X and Y is considered. Instead, the objective is to only capture the trends in X when they exhibit correlation with Y.

As mentioned previously, tensor decomposition factors can be combined with linear regression models. This two-step approach bears a resemblance to principal-component regression: first, the data is decomposed using tensor decomposition without considering the effects (Y); then, regression is applied to capture correlations between the decomposed factors and their effects. However, as the first step is performed without the knowledge of Y, the decomposed X factors are not guaranteed to associate with Y. To address these challenges, PLS methods have been developed in both classification form (PLS discriminant analysis) and regression form (PLS regression).[45]

Tensor PLS is designed to uncover relationships between two tensors, X and Y, for predictors and responses, wherein one mode is aligned (Figure 4G). For instance, consider tensor X representing medical tests on a group of patients over time, while matrix Y (a two-way tensor) records their diagnosis. The result of tensor PLS is analogous to performing two separate CPs on both X and Y simultaneously with the same number of components. After decomposition, they will each have a distinct patient factor matrix. However, PLS decomposes both datasets with the goal of maximizing the correlations between these two patient factors (Figure 4G). The factors of the other non-aligned modes in X and Y come after obtaining the patient factors and are defined to maximally capture variance within each dataset.[46] While the intricacies of the solving algorithm extend beyond the scope of this review, one helpful property to note is that tensor PLS is solved component-by-component. Each additional component is solved upon the residuals of X and Y (X′ and Y′), which are the original tensors subtracted by the solved components (Figure 4G), meaning that components are ordered by the covariance they explain. Therefore, in a correctly performed tensor PLS, the fitting errors of both X and Y should decrease monotonically as more components are added (Figure 4H). However, as a supervised learning method, tensor PLS does not always predict unseen samples better with more components due to the risk of overfitting. The optimal number of components can be determined through cross-validation, where a portion of the samples is left out during fitting to test the model's performance on them. It is expected that the prediction error of Y in cross-validation would initially decrease if the optimal number of components is greater than one (which is usually the case if Y is a matrix rather than a vector) and then increase after reaching the optimum (Figure 4H).

Overall, PLS has unique advantages when focused on a particular response. Since it is designed to specifically discover those patterns associated with a prediction of interest, PLS can predict the effect with fewer components compared with CP. Tensor PLS can be combined with Tucker decomposition and coupling in explanatory (X) tensors, and techniques are available to handle missing values.[47]

BIOLOGICAL INSIGHTS FROM TENSOR-BASED METHODS

Tensor decompositions have applications in virtually all fields of biological data analysis. In this section, we summarize several notable examples.

Applications in bioinformatics

In bioinformatic studies, multi-omics data may contain tens of thousands of genes and millions of genomic positions. Tensor methods can simplify these large datasets generated by high-throughput techniques into a succinct set of components and do so more efficiently than matrix-based counterparts. These reduced latent structures group genes based on their common patterns revealed by the data, easing the scale of effect prediction.

Hore et al. illustrated how tensor methods can be applied to condense genes in RNA-sequencing (RNA-seq) data across multiple tissues into associated factors to reduce the scale of statistical testing and to strengthen their statistical power.[2] To reveal gene networks, they structured the gene expression levels into a gene by individual by tissue tensor. After applying the tensor method, the data was reduced into around two hundred components, a great reduction from the tens of thousands of genes they originally dealt with. These components grouped the genes by activities and indicated in what tissues they were active. Using individual scores as genotypes for genome-wide scanning on SNPs, they discovered the components that were significantly associated with trans-expression quantitative trait loci (eQTLs) and revealed their specific pathway or epigenomic regulation.

Using tensor factors to cluster genes in the transcriptome is further exemplified by Wang et al.[25] With the increasing scale of multi-tissue datasets, classical clustering methods struggle to extract information from multi-way interactions in the transcriptome. To fully extract the three-way interactions between individuals, genes, and tissues, they applied constrained CP to RNA-seq and microarray measurements. Besides being able to run on three-dimensional data where traditional methods failed to reveal true patterns in simulated data, this tensor-based clustering method was shown to better test for differentially expressed genes with improved statistical power compared with single-tissue tests.

Durham et al.,[8] on the other hand, applied tensor methods to large epigenome projects such as Encyclopedia of DNA Elements (ENCODE)[9] and the Roadmap Epigenomics Project.[48] In these massive datasets, many cell type and assay pairs were not measured due to time and funding constraints. Therefore, the imputation of these data has been extensively studied.[49] Organizing the ENCODE data into a three-mode tensor, they found that tensor-based imputation outperformed alternative approaches, demonstrating that structuring the data in tensor form helps model and explain variation across the data.

Other tensor methods have been applied to epigenomic data too. For example, a variant of Tucker decomposition has been applied to model spatial association within topologically associating domains.[50] The decomposed factors directly link epigenomic state and chromosomal topology. Tensor decomposition can be also combined with machine learning methods. For

example, extending the work of Durham et al.,[8] the same group inputted the concatenated tensor factors from three different genomic resolutions into a feed-forward deep neural network to predict the epigenomic signals, allowing a multi-scale view of the genome.[51]

Applications in neuroscience

Neuroscience is among the earliest fields to employ tensor methods.[52,53] As electroencephalography and functional magnetic resonance imaging data are collected over time, any experiment involving more than one electrode and trial is guaranteed to be at least three-dimensional. Conventionally, the data has been converted into matrices by averaging multiple trials, inevitably losing information about trial-to-trial variation. Therefore, tensor methods, including both CP and Tucker decomposition, have been attractive to the neural signal processing community.[54]

Williams et al. presented a clean framework for applying tensor component analysis on large-scale neural data across time and trials.[11] Before running on the actual data, they demonstrated that tensor decomposition works well on simulated linear model neural networks and nonlinear recurrent neural networks, separating positive and negative cells with almost perfect accuracy. With the same simulations, PCA and independent component analysis failed to recover the right signal. They then applied the method to their experiments on mice's prefrontal activity and primate motor cortex. Nonnegative tensor decomposition was shown to cleanly separate neurons that were activated in various periods and associated with specific movements.

Applications in systems biology

Systems biology makes repeated measurements over different times, tissues, or spatial structures, so the data are naturally in tensor structure. These measurements may include sequencing, flow cytometry, or quantitative cell imaging, requiring solutions for data integration. Two specific concerns here are avoiding overfitting, as the datasets are often limited in size, and incorporating heterogeneous information. Therefore, nonnegative decomposition, imputation tests, coupling, and PLS have been used.

Tensor methods offer unique advantages for the study of systems biology by enabling concurrent comparison of multiple contexts and extracting their shared trends. For instance, Armingol et al. employed tensor decomposition to study cell-to-cell communication from RNA-seq data.[55] Contrary to many previous studies that cannot handle more than two cellular contexts simultaneously, by embedding communication matrices[56] into a four-mode tensor, they were able to characterize variation in cell-to-cell communication across several contexts coordinately.

The benefit of tensor decomposition in analyzing repeated measurements simultaneously can also be extended to compositional data in microbiology. Microbiome studies often take multiple samples from the same individual either longitudinally or spatially, but there is a lack of methods to account for both biological change and interindividual variability in them. Martino et al. took the tensor approach to deconvolute gut microbial sequencing data.[57] They demonstrated that unsupervised tensor decomposition can identify differentially abundant microbes, accounting for the high-dimensional, sparse, and compositional nature of microbiome data.

Tensor PLS can also be helpful in systems biology.[58] Netterfield et al. recently applied it in a study of DNA damage response.[59] They systematically profiled a human cell line with the treatment of DNA double-strand break-inducing drugs over time and concentrations, using tensor PLS to directly associate signaling to response, both as three-mode tensors, separating the time mode from drug concentrations. This allowed them to identify signals with time-dependent correlations with senescence and apoptosis. They also observed that tensor PLS required fewer parameters to predict the response than the conventional unfolded version.

CONCLUSIONS

In this work, we review the application of tensor decomposition to biological data analysis. The paramount lesson of this work is the profound influence of the chosen data representation, the medium, on our comprehension of the data itself and the analytical approach. The selection of data representation should be driven by the natural structure of the underlying data and experiment rather than mere mathematical expediency. Approaching this analysis appropriately improves on the insights one can derive from the data through better accuracy, more evident interpretation, and an enhanced ability to integrate data across studies and scales. While tensor methods have gained increased prominence, they have much broader potential yet.[60] Part of the field's maturation will arise from a broader appreciation and understanding of these techniques.

Nevertheless, tensor decomposition, in its current form, is not without limitations. First, it is still fundamentally linear, so it may fail on datasets with nonlinear characteristics. This does not forbid it from being an adequate baseline model though. Furthermore, the existing solving algorithms continue to grapple with numerical issues such as nonuniqueness in factors, instability when addressing missing data values, and challenges in hyperparameter tuning. These issues will be resolved by new theories and a broader appreciation of these techniques.

Data and code availability

The example drug response dataset and the code used to generate the plots in this review can be found at https://github.com/meyer-lab/tensor_review. The code was written in Jupyter Notebook in Python 3.12 with a list of packages required, which accompanies this review and serves as a basic tutorial for tensor method applications.

ACKNOWLEDGMENTS

The authors thank members of the Meyer lab for their helpful feedback in the preparation of this manuscript. This work was supported by NIH U01AI148119 to A.S.M., NIH U19AI172713 to A.S.M., and an Emerging Leader Award to A.S.M. from the Mark Foundation for Cancer Research.

AUTHOR CONTRIBUTIONS

Conceptualization, Z.C.T.; writing – original draft, Z.C.T. and A.S.M.; writing – review & editing, Z.C.T. and A.S.M.; funding acquisition, A.S.M.; supervision, A.S.M.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

1. Omberg, L., Meyerson, J.R., Kobayashi, K., Drury, L.S., Diffley, J.F.X., and Alter, O. (2009). Global effects of DNA replication and DNA replication origin activity on eukaryotic gene expression. Mol. Syst. Biol. *5*, 312. https://doi.org/10.1038/msb.2009.70.

2. Hore, V., Viñuela, A., Buil, A., Knight, J., McCarthy, M.I., Small, K., and Marchini, J. (2016). Tensor decomposition for multiple-tissue gene expression experiments. Nat. Genet. *48*, 1094–1100. https://doi.org/10.1038/ng.3624.

3. Li, H., Ning, S., Ghandi, M., Kryukov, G.V., Gopal, S., Deik, A., Souza, A., Pierce, K., Keskula, P., Hernandez, D., et al. (2019). The landscape of cancer cell line metabolism. Nat. Med. *25*, 850–860. https://doi.org/10.1038/s41591-019-0404-8.

4. Jones, D.S., Jenney, A.P., Swantek, J.L., Burke, J.M., Lauffenburger, D.A., and Sorger, P.K. (2017). Profiling drugs for rheumatoid arthritis that inhibit synovial fibroblast activation. Nat. Chem. Biol. *13*, 38–45. https://doi.org/10.1038/nchembio.2211.

5. Sikkema, L., Ramírez-Suástegui, C., Strobl, D.C., Gillett, T.E., Zappia, L., Madissoon, E., Markov, N.S., Zaragosi, L.-E., Ji, Y., Ansari, M., et al. (2023). An integrated cell atlas of the lung in health and disease. Nat. Med. *29*, 1563–1577. https://doi.org/10.1038/s41591-023-02327-2.

6. Lloyd-Price, J., Mahurkar, A., Rahnavard, G., Crabtree, J., Orvis, J., Hall, A.B., Brady, A., Creasy, H.H., McCracken, C., Giglio, M.G., et al. (2017). Strains, functions and dynamics in the expanded Human Microbiome Project. Nature *550*, 61–66. https://doi.org/10.1038/nature23889.

7. Gross, S.M., Dane, M.A., Smith, R.L., Devlin, K.L., McLean, I.C., Derrick, D.S., Mills, C.E., Subramanian, K., London, A.B., Torre, D., et al. (2022). A multi-omic analysis of MCF10A cells provides a resource for integrative assessment of ligand-mediated molecular and phenotypic responses. Commun. Biol. *5*, 1066. https://doi.org/10.1038/s42003-022-03975-9.

8. Durham, T.J., Libbrecht, M.W., Howbert, J.J., Bilmes, J., and Noble, W.S. (2018). PREDICTD PaRallel Epigenomics Data Imputation with Cloud-based Tensor Decomposition. Nat. Commun. *9*, 1402. https://doi.org/10.1038/s41467-018-03635-9.

9. Dunham, I., Kundaje, A., Aldred, S.F., Collins, P.J., Davis, C.A., Doyle, F., Epstein, C.B., Frietze, S., Harrow, J., and Kaul, R. (2012). An integrated encyclopedia of DNA elements in the human genome. Nature *489*, 57–74. https://doi.org/10.1038/nature11247.

10. Kemper, K.E., Sidorenko, J., Wang, H., Hayes, B.J., Wray, N.R., Yengo, L., Keller, M.C., Goddard, M., and Visscher, P.M. (2024). Genetic influence on within-person longitudinal change in anthropometric traits in the UK Biobank. Nat. Commun. *15*, 3776. https://doi.org/10.1038/s41467-024-47802-7.

11. Williams, A.H., Kim, T.H., Wang, F., Vyas, S., Ryu, S.I., Shenoy, K.V., Schnitzer, M., Kolda, T.G., and Ganguli, S. (2018). Unsupervised Discovery of Demixed, Low-Dimensional Neural Dynamics across Multiple Timescales through Tensor Component Analysis. Neuron *98*, 1099–1115.e8. https://doi.org/10.1016/j.neuron.2018.05.015.

12. McLuhan, M. (1964). The Medium is the Message. In Understanding Media: The Extensions of Man (McGraw-Hill).

13. Bro, R. (1997). PARAFAC. Tutorial and applications. Chemom. Intell. Lab. Syst. *38*, 149–171. https://doi.org/10.1016/S0169-7439(97)00032-4.

14. Kolda, T.G., and Bader, B.W. (2009). Tensor Decompositions and Applications. SIAM Rev. *51*, 455–500. https://doi.org/10.1137/07070111X.

15. Rabanser, S., Shchur, O., and Günnemann, S. (2017). Introduction to Tensor Decompositions and their Applications in Machine Learning. Preprint at arXiv. https://doi.org/10.48550/arXiv.1711.10781.

16. Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., et al. (2020). Array programming with NumPy. Nature *585*, 357–362. https://doi.org/10.1038/s41586-020-2649-2.

17. Yahyanejad, F., Albert, R., and DasGupta, B. (2019). A survey of some tensor analysis techniques for biological systems. Quant. Biol. *7*, 266–277. https://doi.org/10.1007/s40484-019-0186-5.

18. Caulk, A.W., and Janes, K.A. (2019). Robust latent-variable interpretation of in vivo regression models by nested resampling. Sci. Rep. *9*, 19671. https://doi.org/10.1038/s41598-019-55796-2.

19. Bro, R., and Smilde, A.K. (2003). Centering and scaling in component analysis. J. Chemom. *17*, 16–33. https://doi.org/10.1002/cem.773.

20. Kossaifi, J., Panagakis, Y., Anandkumar, A., and Pantic, M. (2019). TensorLy: Tensor Learning in Python. J. Mach. Learn. Res. *20*, 1–6.

21. Bader, B.W., and Kolda, T.G. (2006). Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. ACM Trans. Math. Softw. *32*, 635–653. https://doi.org/10.1145/1186785.1186794.

22. Li, J., Bien, J., and Wells, M.T. (2018). rTensor: An R Package for Multidimensional Array (Tensor) Unfolding, Multiplication, and Decomposition. J. Stat. Soft. *87*, 1–31.

23. Helwig, N.E. (2019). multiway: Component Models for Multi-Way Data. Software. http://CRAN.R-project.org/package=multiway.

24. Krijnen, W.P., Dijkstra, T.K., and Stegeman, A. (2008). On the Non-Existence of Optimal Solutions and the Occurrence of "Degeneracy" in the CANDECOMP/PARAFAC Model. Psychometrika *73*, 431–439. https://doi.org/10.1007/s11336-008-9056-1.

25. Wang, M., Fischer, J., and Song, Y.S. (2019). Three-way clustering of multi-tissue multi-individual gene expression data using semi-nonnegative tensor decomposition. Ann. Appl. Stat. *13*, 1103–1127. https://doi.org/10.1214/18-AOAS1228.

26. Acar, E., Kolda, T., and Dunlavy, D. (2009). An optimization approach for fitting canonical tensor decompositions. OSTI.Gov. https://doi.org/10.2172/978916.

27. Tomasi, G., and Bro, R. (2006). A comparison of algorithms for fitting the PARAFAC model. Comp. Stat. Data Anal. *50*, 1700–1734. https://doi.org/10.1016/j.csda.2004.11.013.

28. Roald, M., Schenker, C., Calhoun, V.D., Adali, T., Bro, R., Cohen, J.E., and Acar, E. (2022). An AO-ADMM Approach to Constraining PARAFAC2 on All Modes. SIAM Journal on Mathematics of Data Science *4*, 1191–1222. https://doi.org/10.1137/21M1450033.

29. Tomasi, G., and Bro, R. (2005). PARAFAC and missing values. Chemom. Intell. Lab. Syst. *75*, 163–180. https://doi.org/10.1016/j.chemolab.2004.07.003.

30. Acar, E., Dunlavy, D.M., Kolda, T.G., and Mørup, M. (2011). Scalable tensor factorizations for incomplete data. Chemom. Intell. Lab. Syst. *106*, 41–56. https://doi.org/10.1016/j.chemolab.2010.08.004.

31. Ahn, M., Eikmeier, N., Haddock, J., Kassab, L., Kryshchenko, A., Leonard, K., Needell, D., Madushani, R.W.M.A., Sizikova, E., and Wang, C. (2021). On Large-Scale Dynamic Topic Modeling with Nonnegative CP Tensor Decomposition. In Advances in Data Science Association for Women in Mathematics, I.D. Series, Y. Lou, X. Wang, and K. Welker, eds. (Springer International Publishing), pp. 181–210. https://doi.org/10.1007/978-3-030-79891-8_8.

32. Huang, F., Yue, X., Xiong, Z., Yu, Z., Liu, S., and Zhang, W. (2021). Tensor decomposition with relational constraints for predicting multiple types of microRNA-disease associations. Brief. Bioinform. *22*, bbaa140. https://doi.org/10.1093/bib/bbaa140.

33. Sen, B., and Parhi, K.K. (2017). Extraction of common task signals and spatial maps from group fMRI using a PARAFAC-based tensor decomposition technique. In 2017 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 1113–1117. https://doi.org/10.1109/ICASSP.2017.7952329.

34. Lyu, H., Wan, M., Han, J., Liu, R., and Wang, C. (2017). A filter feature selection method based on the Maximal Information Coefficient and Gram-Schmidt Orthogonalization for biomedical data mining. Comput. Biol. Med. *89*, 264–274. https://doi.org/10.1016/j.compbiomed.2017.08.021.

35. Tucker, L.R. (1966). Some mathematical notes on three-mode factor analysis. Psychometrika *31*, 279–311. https://doi.org/10.1007/BF02289464.

36. De Lathauwer, L., De Moor, B., and Vandewalle, J. (2000). A Multilinear Singular Value Decomposition. SIAM J. Matrix Anal. Appl. *21*, 1253–1278. https://doi.org/10.1137/S0895479896305696.

37. Bro, R., and Kiers, H.A.L. (2003). A new efficient method for determining the number of components in PARAFAC models. J. Chemom. *17*, 274–286. https://doi.org/10.1002/cem.801.

38. Sankaranarayanan, P., Schomay, T.E., Aiello, K.A., and Alter, O. (2015). Tensor GSVD of Patient- and Platform-Matched Tumor and Normal DNA Copy-Number Profiles Uncovers Chromosome Arm-Wide Patterns of Tumor-Exclusive Platform-Consistent Alterations Encoding for Cell Transformation and Predicting Ovarian Cancer Survival. PLoS ONE *10*, e0121396. https://doi.org/10.1371/journal.pone.0121396.

39. Omberg, L., Golub, G.H., and Alter, O. (2007). A tensor higher-order singular value decomposition for integrative analysis of DNA microarray data from different studies. Proc. Natl. Acad. Sci. USA *104*, 18371–18376. https://doi.org/10.1073/pnas.0709146104.

40. Acar, E., Bro, R., and Smilde, A.K. (2015). Data Fusion in Metabolomics Using Coupled Matrix and Tensor Factorizations. Proc. IEEE *103*, 1602–1620. https://doi.org/10.1109/JPROC.2015.2438719.

41. Acar, E., Kolda, T.G., and Dunlavy, D.M. (2011). All-at-once Optimization for Coupled Matrix and Tensor Factorizations. Preprint at arXiv. https://doi.org/10.48550/arXiv.1105.3422.

42. Tan, Z.C., Murphy, M.C., Alpay, H.S., Taylor, S.D., and Meyer, A.S. (2021). Tensor-structured decomposition improves systems serology analysis. Mol. Syst. Biol. *17*, e10243. https://doi.org/10.15252/msb.202110243.

43. Chin, J.L., Tan, Z.C., Chan, L.C., Ruffin, F., Parmar, R., Ahn, R., Taylor, S.D., Bayer, A.S., Hoffmann, A., Fowler, V.G., Jr., et al. (2024). Tensor modeling of MRSA bacteremia cytokine and transcriptional patterns reveals coordinated, outcome-associated immunological programs. PNAS Nexus *3*, pgae185. https://doi.org/10.1093/pnasnexus/pgae185.

44. Schenker, C., Wang, X., and Acar, E. PARAFAC2-based Coupled Matrix and Tensor Factorizations. 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). 10.1109/ICASSP49357.2023.10094562.

45. Kreeger, P.K. (2013). Using Partial Least Squares Regression to Analyze Cellular Response Data. Sci. Signal. *6*, tr7. https://doi.org/10.1126/scisignal.2003849.

46. Bro, R. (1996). Multiway calibration. Multilinear PLS. J. Chemometrics *10*, 47–61. https://doi.org/10.1002/(SICI)1099-128X(199601)10:1<47::AID-CEM400>3.0.CO;2-C.

47. Folch-Fortuny, A., Arteaga, F., and Ferrer, A. (2017). PLS model building with missing data: New algorithms and a comparative study. J. Chemom. *31*, e2897. https://doi.org/10.1002/cem.2897.

48. Bernstein, B.E., Stamatoyannopoulos, J.A., Costello, J.F., Ren, B., Milosavljevic, A., Meissner, A., Kellis, M., Marra, M.A., Beaudet, A.L., Ecker, J.R., et al. (2010). The NIH Roadmap Epigenomics Mapping Consortium. Nat. Biotechnol. *28*, 1045–1048. https://doi.org/10.1038/nbt1010-1045.

49. Ernst, J., and Kellis, M. (2015). Large-scale imputation of epigenomic datasets for systematic annotation of diverse human tissues. Nat. Biotechnol. *33*, 364–376. https://doi.org/10.1038/nbt.3157.

50. Zhu, Y., Chen, Z., Zhang, K., Wang, M., Medovoy, D., Whitaker, J.W., Ding, B., Li, N., Zheng, L., and Wang, W. (2016). Constructing 3D interaction maps from 1D epigenomes. Nat. Commun. *7*, 10812. https://doi.org/10.1038/ncomms10812.

51. Schreiber, J., Durham, T., Bilmes, J., and Noble, W.S. (2020). Avocado: a multi-scale deep tensor factorization method learns a latent representation of the human epigenome. Genome Biol. *21*, 81. https://doi.org/10.1186/s13059-020-01977-6.

52. Hunyadi, B., Dupont, P., Van Paesschen, W., and Van Huffel, S. (2017). Tensor decompositions and data fusion in epileptic electroencephalography and functional magnetic resonance imaging data. WIREs Data Min. Knowl. Discov. *7*, e1197. https://doi.org/10.1002/widm.1197.

53. Cong, F., Lin, Q.-H., Kuang, L.-D., Gong, X.-F., Astikainen, P., and Ristaniemi, T. (2015). Tensor decomposition of EEG signals: A brief review. J. Neurosci. Methods *248*, 59–69. https://doi.org/10.1016/j.jneumeth.2015.03.018.

54. Cong, F., Phan, A.-H., Astikainen, P., Zhao, Q., Wu, Q., Hietanen, J.K., Ristaniemi, T., and Cichocki, A. (2013). Multi-domain feature extraction for small event-related potentials through nonnegative multi-way array decomposition from low dense array EEG. Int. J. Neural Syst. *23*, 1350006. https://doi.org/10.1142/S0129065713500068.

55. Armingol, E., Baghdassarian, H.M., Martino, C., Perez-Lopez, A., Aamodt, C., Knight, R., and Lewis, N.E. (2022). Context-aware deconvolution of cell–cell communication with Tensor-cell2cell. Nat. Commun. *13*, 3665. https://doi.org/10.1038/s41467-022-31369-2.

56. Armingol, E., Officer, A., Harismendy, O., and Lewis, N.E. (2021). Deciphering cell–cell interactions and communication from gene expression. Nat. Rev. Genet. *22*, 71–88. https://doi.org/10.1038/s41576-020-00292-x.

57. Martino, C., Shenhav, L., Marotz, C.A., Armstrong, G., McDonald, D., Vázquez-Baeza, Y., Morton, J.T., Jiang, L., Dominguez-Bello, M.G., Swafford, A.D., et al. (2021). Context-aware dimensionality reduction deconvolutes gut microbial community dynamics. Nat. Biotechnol. *39*, 165–168. https://doi.org/10.1038/s41587-020-0660-7.

58. Chitforoushzadeh, Z., Ye, Z., Sheng, Z., LaRue, S., Fry, R.C., Lauffenburger, D.A., and Janes, K.A. (2016). TNF-insulin crosstalk at the transcription factor GATA6 is revealed by a model that links signaling and transcriptomic data tensors. Sci. Signal. *9*, ra59. https://doi.org/10.1126/scisignal.aad3373.

59. Netterfield, T.S., Ostheimer, G.J., Tentner, A.R., Joughin, B.A., Dakoyannis, A.M., Sharma, C.D., Sorger, P.K., Janes, K.A., Lauffenburger, D.A., and Yaffe, M.B. (2023). Biphasic JNK-Erk signaling separates the induction and maintenance of cell senescence after DNA damage induced by topoisomerase II inhibition. Cell Syst. *14*, 582–604.e10. https://doi.org/10.1016/j.cels.2023.06.005.

60. Mor, U., Cohen, Y., Valdés-Mas, R., Kviatcovsky, D., Elinav, E., and Avron, H. (2022). Dimensionality Reduction of Longitudinal 'Omics Data using Modern Tensor Factorization. PLoS Comput. Biol. *18*, e1010212. https://doi.org/10.1371/journal.pcbi.1010212.