Supplemental Information

NIPALS Algorithm

**Algorithm 1 – Nonlinear Iterative Partial Least square (NIPALS) algorithm for PCA.**

```
initialize X₀ = X
for i = 1, ..., r
    initialize fᵢ as the first column in Xᵢ-1
    1. qᵢ = Xᵢ-1 fᵢ / (fᵢ T fᵢ)₁/₂
    2. qᵢ = qᵢ / (qᵢ T qᵢ)₁/₂
    3. fᵢ = Xᵢ-1 qᵢ
    4. Check convergence of fᵢ and qᵢ, if not, go back to step 1; otherwise step 5.
    5. Xᵢ = Xᵢ-1 - fᵢ qᵢ T # deflation step
```

This algorithm is particularly useful in the analysis of omics data because, when compared to SVD, it requires less computation time and can handle missing data. The NIPALS algorithm calculating components and loadings via regression for each dimension (step 1 to 3), hence a small number of missing values will not affect the procedure. For a higher order solution, the same procedure is applied on the residual matrix calculated from the deflation step (step 5 in algorithm 1). The residual matrix can be viewed as the regression of Xᵢ-1 onto fᵢ or removing the variance explained by fᵢ from Xᵢ-1. Therefore, it is faster when the matrix X is large because it calculates a subset of PCs whereas SVD computes all PCs. Of particular interest in multi-omics data analysis, this algorithm may be generalized to discover the correlated structure in more than one datasets (see sections on the analysis of multi omics datasets).

NMF

Non-negative matrix factorization (NMF) is an approach adopted from signal processing where it was used to solve the blind source deconvolution problem and has been widely applied in clustering analysis, face recognition and text mining [1]. NMF is also called self-modeling curve resolution or positive matrix factorization. NMF can be defined by the following model:

\[ X = WH + E, \text{ subject to the constraint } W, H >= 0. \]  \hspace{1cm} (9)

W and H are nxr and rxp matrices, respectively; E is an error term accounting for inaccurate reconstruction. Similar to PCA and other decomposition approaches, NMF seeks to explain the principal sources of variance in a dataset using a small number \( r \) of vectors. But NMF solves this problem by minimizing the sum of squared errors E. Furthermore, it forces a positive or non-negative constraint on the resulting data matrices and [similar to Independent Component Analysis; 2] does not require orthogonality or independence in the components. This allows NMF to identify overlapping patterns in components. At the same time, the non-negative constraint guarantees that only the additive combinations of latent variables are allowed, which has more intuitive meaning since many biological variables could only be represented by positive values, such as protein concentration and count data.

The duality diagram

In the 1970s, French statisticians Cazes [3], Cailliez and Pages [4] developed a unifying framework, called the duality diagram, which provides an elegant approach to formulate dimension reduction
methods in a similar way [see also 5, 6-8]. This framework is based on the statistical triplet $(X, R, D)$ where $X$ is a matrix with $n$ rows (observations or samples) and $p$ columns (variables or genes). The row-column duality suggested that rows and columns can be viewed in a symmetrical way, i.e. the matrix could be viewed as consisting of $n$ points (variables/genes) defined in the space $\mathbb{R}^p$; or $p$ points (samples) defined in the space $\mathbb{R}^n$.

$R$ and $D$ are positive symmetric matrices with dimension $p \times p$ and $n \times n$, respectively, which provide a metric used to compute inner products in $\mathbb{R}^p$ and $\mathbb{R}^n$, respectively. From a geometrical point of view, analyzing the statistical triplet $(X, R, D)$ can be formulated as either finding principal axes of a dataset containing $n$ points in $\mathbb{R}^p$ or as finding the principal components of $p$ points in $\mathbb{R}^n$. The introduction of $R$ and $D$ makes the duality diagram a highly versatile framework that includes many dimension reduction methods as special cases. For example, PCA in the original scale can be formulated as a duality diagram with $R = I$ (identity) and $D$ is a diagonal matrix consisting of uniform row weights ($1/n$). CA can be formulated as a duality diagram by defining $R$ and $D$ as the marginal frequencies of the original matrix and by standardizing $X$ so that it captures the departure from independence of the original data. If only $R$ or $D$ are defined as above, it formulates the NSCA. Co-inertia analysis or decomposition of multiple matrices is a simple extension of this. Further examples and mathematical details are available in several excellent reviews [7-9]. Dimension reduction approaches based on the duality diagram are implemented in the R package ade4 [10].

Co-Inertia Analysis

CIA is a descriptive non-constrained approach for coupling pairs of data matrices. It was originally proposed to link two ecological tables [9, 11], but it has already been successfully applied in omics data analysis [12, 13]. CIA is implemented under the duality diagram framework in the ade4 package. In this scheme, CIA analyzes two statistical triplets $(X, L, D)$ and $(Y, R, D)$. In physics, the inertia of a set of points relative to a reference point is defined by the weighted sum of squared distances between each considered point and the reference point. Correspondingly, the inertia of a centered matrix (mean is equal to zero) is simply the sum of the squared matrix elements. The inertia of the matrix $X$ defined by the metrics $L$ and $D$ is the weighted sum of its squared values, that is:

$$\text{trace}(XLX^T D).$$

(1)

The inertia equals the total variance of $X$ when $X$ is centered, $L$ is the Euclidean metric and $D$ is a diagonal matrix with $l_i = 1/n$. However, the concept of inertia is more flexible since different metrics may be used to account for different types of data. For example, if $L$ and $D$ are defined according to CA the inertia of $X$ is proportional to the $\chi^2$ statistics.

When coupling a pair of datasets, the co-inertia between two matrices, $X$ and $Y$, is calculated as

$$\text{trace}(XLX^T DYR^T Y D).$$

(2)

CIA decomposes the co-inertia criteria into a set of orthogonal axes.

Regularized generalized CCA

Recently, Tenenhaus et al. proposed regularized generalized CCA, which provides a unified framework for different multi-table multivariate methods [15]. This method was defined as the following problem:
arg max \( q_{i\ldots q_k} \sum_{k,j=1; k \neq j}^K c_{kj} g(\text{cov}(X_k q_k, X_j q_j)) \)

subject to the constraint \( \tau_k \| q_k \|^2 + (1 - \tau_k) \text{var}(X_k q_k) = 1 \)

where \( c_{kj} \) is a linkage function so that \( c_{kj} = 1 \) if data \( k \) and \( j \) are connected and otherwise \( c_{kj} = 0 \). The function \( g \) represents different optimization criteria; it could be an identity (i.e. sum of covariance criterion), a square function (i.e. sum of squared covariance criterion) or an absolute function (i.e. sum of the absolute covariance criterion). Last, \( \tau \) is a shrinkage parameter ranging from 0 to 1; setting \( \tau \) to 0 will force the “canonical variates” or “components” to unit variance (i.e. \( \text{var}(X_k q_k) = 1 \)), in which case the covariance is equal to the correlation (i.e. criterion used by GCCA). As described before (equation (15)), the correlation criterion is better in explaining the correlated structure across datasets, thus discarding the variance within each individual dataset. Setting \( \tau \) equal to 1 will normalize the loadings \( q_k \) to 1, which applies the covariance criterion. A value between 0 and 1 will lead to a compromise between the two options. The introduction of the above three parameters \( (c,g,\tau) \) enables RGCCA to perform many similar methods, including GCCA, CIA and CPCA, as particular cases. In MCIA and CPCA, the squared covariance between the individual datasets are compared to the global structure. Therefore, it is comparable with RGCCA when \( X_j \) is the concatenated matrix and \( g \) is the squared function (equation 17). In this context, GCCA can be expressed as a mixed correlation and covariance criteria [15].

**R Code to generate figures**

R code to generate figures is available as a supplement. An rmarkdown version of this, image files of the results and the data necessary to generate the figures is available on the github repository https://github.com/aedin/NCI60Example. To install this package from github, type the following in R.

```r
library(devtools)
install_github("aedin/NCI60Example")
library(NCI60Example)
data(nci60)
```