OXFORD

Briefings in Bioinformatics, 19(2), 2018, 325–340

doi: 10.1093/bib/bbw113
Advance Access Publication Date: 9 December 2016
Paper

A review on machine learning principles for multi-view biological data integration

Yifeng Li, Fang-Xiang Wu, and Alioune Ngom

Corresponding author: Yifeng Li, Information and Communications Technologies, National Research Council Canada, Ottawa, Ontario, Canada.
E-mail: yifeng.li@nrc-cnrc.gc.ca

Abstract

Driven by high-throughput sequencing techniques, modern genomic and clinical studies are in a strong need of integrative machine learning models for better use of vast volumes of heterogeneous information in the deep understanding of biological systems and the development of predictive models. How data from multiple sources (called multi-view data) are incorporated in a learning system is a key step for successful analysis. In this article, we provide a comprehensive review on omics and clinical data integration techniques, from a machine learning perspective, for various analyses such as prediction, clustering, dimension reduction and association. We shall show that Bayesian models are able to use prior information and model measurements with various distributions; tree-based methods can either build a tree with all features or collectively make a final decision based on trees learned from each view; kernel methods fuse the similarity matrices learned from individual views together for a final similarity matrix or learning model; network-based fusion methods are capable of inferring direct and indirect associations in a heterogeneous network; matrix factorization models have potential to learn interactions among features from different views; and a range of deep neural networks can be integrated in multi-modal learning for capturing the complex mechanism of biological systems.

Key words: data integration; multi-omics data; random forest; multiple kernel learning; network fusion; matrix factorization; deep learning

Introduction

In this big data era, information grow almost exponentially in volume, variety and complexity [1]. For example, in current biomedical research, it is not uncommon to have access to a large amount of data from a single patient, such as clinical records (e.g. age, sex, histories, pathologies and therapeutics), high-throughput omics data (e.g. genomics, transcriptomics, proteomics and metabolomics measurements) and so on, all

Yifeng Li is a research scientist at the National Research Council Canada since 2015. Recognized by the Governor General's Gold Medal, he obtained Ph.D. from the University of Windsor in 2013. From 2013 to 2015, supported by the NSERC Postdoctoral Fellowship, he was a post-doctoral trainee at the University of British Columbia. His research interests include sparse machine learning models, deep learning models, matrix factorizations, feature selection, data integration, large-scale optimization, big data analysis in bioinformatics and health-informatics, gene regulations and cancer studies. He is a member of IEEE and CAIAC.

Fang-Xiang Wu is a professor of the Division of Biomedical Engineering and the Department of Mechanical Engineering at the University of Saskatchewan. His current research interests include computational and systems biology, genomic and proteomic data analysis, biological system identification and parameter estimation, applications of control theory to biological systems. He has published more than 260 technical papers. Dr Wu is serving as the editorial board member of three international journals, the guest editor of several international journals and as the program committee chair or member of several international conferences. He has also reviewed papers for many international journals. He is a senior member of IEEE.

Alioune Ngom received his Ph.D. in 1998 at the University of Ottawa, and is currently a professor at the School of Computer Science, University of Windsor. Before joining UWindsor in 2000, he has held an assistant professor position at Lakehead University. His main research interests include but are not limited to computational intelligence and machine learning methods, and their applications in the fields of computational biology and bioinformatics. His current research includes gene regulatory network reconstruction, protein complex identification, sparse representation learning, network clustering and biomarker selection. He is a member of IEEE.

Submitted: 26 July 2016; **Received (in revised form):** 12 October 2016

under proper multi-party consents. In this article, we use the term 'multi-view data' to denote any kinds of heterogeneous (could be homogeneous) data that provide complementary information to characterize a biological object, phenomenon or system from various aspects. Such data may be of different types and from different sources, follow different statistical distributions, possess different semantics, suffer from different levels of imprecisions and contain different kinds of uncertainties. Specifically, we are interested in four types of multi-view data: (1) multi-view data with different groups of samples measured by the same feature set (or called multi-class data), (2) multi-view data with the same set of objects (samples) but several distinct feature sets, (3) multi-view data measuring the same set of objects by the same set of features in different conditions (can be represented by a three-way *sample* × *feature* × *condition* tensor) and (4) multi-view data with different features and different sample sets in the same phenomenon or system, which can be further transformed to multi-relational data.

The type-2 and type-4 multi-view data described above are often referred as multi-omics data. Generation of type-2 multi-omics data requires collaborative efforts within a big consortium such as the Cancer Genome Atlas [2], the Encyclopedia of DNA Elements Consortium [3], the Roadmap Epigenomics Project [4] and the Genotype-Tissue Expression Project [5]. The wide existence of type-4 multi-omics data is often owing to an uncoordinated contribution of independent (small) projects. Single-omics data only describe a biological process at one specific molecular level. For example, whole genome (or exome) sequencing [6] detects single-nucleotide and structural variations (genetic level); ChIP-seq [7] identifies transcription factor binding sites (protein–DNA interactomic level) and histone modifications (epigenomic level) in the entire human genome; DNase-seq [8] detects open chromatin regions for transcription factor binding loci (epigenomic level); whole genome bisulfite-seq [9] allows to build methylomes which are key to understand gene regulation, X chromosome inactivation and cancerogenesis (epigenetic level); RNA-seq [10] can be used to capture gene expression level, and discover alternative splicing, gene fusion and novel isoforms (transcriptomic level); microRNA (miRNA)-seq [11] snapshots expression of micro-RNAs that regulate mRNA translations (translation level); and protein arrays [12] and mass spectrometers are useful to detect concentration of proteins [13] and metabolites [14] (proteomic and metabolomic levels). To identify acting pathways from DNA variations and epigenetic changes to proteins and metabolites, omics data at each level should be generated for the same tissues. Single-omics data enumerated above have the following characteristics: (1) high dimensionality, (2) redundancy, (3) highly correlated features and (4) non-negativity. On top of them, multi-omics data have the following characteristics: (1) mutual complementarity, (2) causality and (3) heterogeneity.

In bioinformatics, there are five types of data-driven analyses where integrative machine learning methods are required. The first is multi-class feature selection and classification problem, where given multiple groups of objects measured using the same set of features, one is often interested in selecting key features responsible for the separation of these groups. One example is meta-analysis or pan-analysis of gene profiles from many distinct tumor tissues. Performance of classification can be measured by area under the receiver operating characteristic curve (auROC) for balanced data and area under the precision-recall curve (auPRC) for imbalanced data. Second, integrating multi-omics data of the same set of labeled objects is expected to escalate prediction (classification or regression) power, for example, the early detection of cancers based on multi-platform data. Third, in the above setting but without class labels, the task becomes an unsupervised learning to discover novel groups of samples. Tumor subtyping is such a commonly conducted analysis. Performance of clustering can be computationally quantified using simulation study, multi-view extensions of index criteria [15] and enrichment analysis. Fourth, given multiple heterogeneous feature sets observed for the same or group of samples, the interactions among inter-view features could be crucial to understand the pathways of a phenotype. The obtained potential pathways should be validated by computational enrichment analysis and wet-lab experiments. Last, given homogeneous and heterogeneous relations within and between multiple sets of biological entries from different molecular levels and clinical descriptions, inferring the relations between inter-set entries is named association study in a complex system. The findings should be finally tested by wet-lab experiments.

On the one hand, multi-view data provide us with an unprecedented opportunity to understand a complex biological system from different angles and levels (e.g. genotype–phenotype interactions [16] and cancer studies [17]), and make precise data-driven predictions (e.g. drug response prediction [18]). For instance, intelligent learning systems have been successfully used in the genome-wide detection of *cis*-regulatory regions [19] to combine sequence information, transcription factor binding, histone modifications, chromatin accessibility as well as 3D genome information (such as DNA shapes and genomic domain interactions) for a comprehensive description of *cis*-regulatory activities. On the other hand, it poses a tough challenge for machine learning experts and data scientists to wisely optimize the use of these data for specific needs. In accordance with when multi-view data are incorporated into a learning process, data fusion techniques can be classified as early, intermediate or late integration methods [20]. In early integration methods, features from different data are concatenated into a single feature vector before fitting an unsupervised or supervised model. In late integration, separate models are first learned for individual views, then their outputs are further combined to make the final determination. An intermediate strategy globally involves data integration in a learning process. Thus, the design of a computational intelligent model should be determined by the nature of multi-view data, the need of an analysis and the complexity of incorporating these multi-view data.

In the frontier of big biological data analytics, it becomes indispensable to investigate the fundamental principles of integrating multi-view data to provide bioinformaticians and data scientists an aerial view and guide to choose and devise multi-view methods. In this review, we focus on integrative machine learning principles for the five types of analyses using the four categories of multi-view data. This article is a significant extension of our preliminary discussion of data integration in a workshop [21]. During the preparation of this extension, we learned that other studies also independently recognized that data integration is an urgent need in current and future bioinformatics, for example, the work of [22], which highlights network methods and non-negative matrix factorizations (NMFs); however, our study covers more comprehensive discussions, such as tree-based methods and multi-modal deep learning, as well as the latest advances, such as partial-least-squares-based models and similarity network fusion (SNF) approaches. For giving the readers an overview of this article, Table 1 is provided as a summary of various machine-learning-based analyses for the four types of multi-view data. The following sections are organized based on machine learning methodologies. We shall discuss

Table 1. An overview of integrative analyses that can be conducted by machine learning methods on four types of multi-view data. Details regarding these methods and applications are described in separate sections

Integrative method	Type of multi-view data			
	Multi-class data (type-1)	Multi-feature-set data (type-2)	Tensor data (type-3)	Multi-relational data (type-4)
Feature concatenation		Classification Regression Feature selection		
Bayesian models or networks	Classification Feature selection	Classification Regression Feature selection Pathway analysis		
Ensemble learning	Classification Feature selection	Classification Regression Feature selection		
Multiple kernel learning	Classification	Classification Regression clustering		Association study
Network-based methods				Association study
Multi-view matrix or tensor factorization	Classification Feature selection	Classification Feature selection Pathway analysis Clustering	Classification Clustering	Association study
Multi-modal learning		Classification Clustering Association study		

simple feature concatenation, Bayesian models, tree-based ensemble methods, multiple kernel learning, network-based methods, matrix factorizations and deep neural networks.

Feature concatenation

Concatenating all features into a single vector seems to be the 'simplest' principle that brings all features together. Heterogeneous data contain much richer information than homogeneous data. But it is challenging to combine heterogeneous features. Applying feature selection techniques to the concatenated data may improve performance [23]. Sparse linear discriminative models, such as support vector machines (SVMs) [24] and the LASSO family [25], play an important role in the classification, regression and feature selection of multi-class data with combined homogeneous or heterogeneous features. A general linear model can be formulated as $y = g(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b)$ where $\boldsymbol{x}$ is a vector of predictor variables; y is a binary discrete response variable (classification problem) or real-valued variable (regression problem); $\boldsymbol{w}$ is the weight vector; b is a scalar bias term; and $g(\cdot)$ is an inverse link function. From an optimization perspective, linear models can be generally written as

$$\min_{\boldsymbol{w},b} \sum_{n=1}^{N} l(y_n, \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n + b) + \lambda r(\boldsymbol{w}), \tag{1}$$

where $l(\cdot)$ is a loss function, $r(\cdot)$ is a regularization term and λ balances the trade-off between fitting error and model complexity. By making $\boldsymbol{w}$ sparse, using, for example, l_1-regularization, predictor variables corresponding to non-zero weights are taken as important features for the response variable. Thus, sparse linear models can be used for simultaneous prediction and variable selection. The l_1-norm-based shrinkage is known as LASSO. Important generalizations of LASSO include (1) elastic net [26] that uses a weighted combination of l_1- and l_2-norms to select highly correlated predictor variables together, (2) group LASSO [27] that selects among predefined groups of predictor variables and (3) graphical LASSO [28] that applies l_1-regularization to produce sparse precision matrix for estimating sparse graphs. Among them, (sparse) group LASSO [29] is useful to incorporate prior knowledge in the grouping of features [30]. It can also be applied to concatenated multi-omics data with features from the same view hierarchically grouped. Graphical LASSO is useful to incorporate structured information. For example, a graph regularization is used in [31] to select a connected component from SNP network associated to a phenotype.

LASSO is variable selection consistent if and only if the training data $\boldsymbol{X}$ satisfies the irrepresentable condition [32]. But its generalizations—adaptive LASSO [33] and randomized LASSO—using the stability selection [34] are consistent variable selection procedures. Sparse linear models using a robust loss function, such as hinge loss, is robust to outlier samples [35]. The sparse regularization techniques reduce model complexity, thus prevent model selection from overfitting.

The concatenated features require additional downstream processing that may lead to loss of key information. First, because multi-view data are observed in different forms that may include continuous features, discrete features, characters and even graphic data types, converting them into acceptable types (e.g. continuous to discrete, categorical to binary coding) is necessary for certain models such as hidden Markov models. Second, features from multiple views usually have different scales. Particularly for discriminative models, it is sometimes

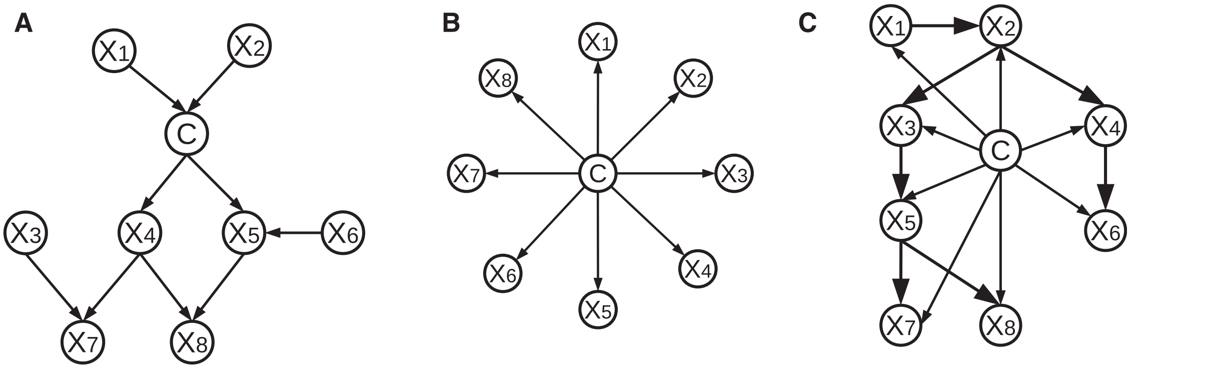

Figure 1. Bayesian network classifiers. (A) General Bayesian network classifier. The class variable is treated as an ordinary node. (B) Naïve Bayes classifier. Features are assumed to be conditionally independent given the class variable as their common parent. (C) Tree-augmented naïve Bayes classifier. Sharing the class variable as parent, the features have a tree structure (in bold edges).

necessary to normalize or standardize the combined features to reduce bias and speed up training process. The feature concatenation strategy followed by normalization is commonly used in linear models such as SVMs and LASSOs. Moreover, feature concatenation is often unworkable with modern data which possess a high dimensionality and rich structural information. For instance, converting medical text documents into a bag of words and combining it together with vectorized image pixels may certainly ignore the importance of language semantics and local structures in images.

Bayesian methods to integrate prior knowledge

In a field, there might exist some *a priori* available knowledge associated with each class, which is useful in a predictive model to better characterize the objects under investigation. From a Bayesian perspective, one can thus consider incorporating this prior knowledge into a learning model, while using the features of interest as regular input features. Suppose we have prior knowledge and input feature data represented by $\boldsymbol{X}^{(p)}$ and $\boldsymbol{X}^{(r)}$, respectively. In a two-class problem (that is, $y \in \{-1, +1\}$), the class prior of a sample can be defined as a logistic function:

$$p(y = +1|\boldsymbol{x}^{(p)}, \boldsymbol{\beta}) = \frac{e^{\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}^{(p)}}}{1 + e^{\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}^{(p)}}}, \tag{2}$$

where $\boldsymbol{\beta}$ is a vector consisting of the coefficients of the corresponding prior features, indicating their contributions to the class prior. Using Bayes' Theorem, the posterior can be decomposed into class-conditional and class-prior probabilities:

$$\begin{aligned} &p(y = +1|\boldsymbol{x}^{(p)}, \boldsymbol{x}^{(r)}, \boldsymbol{\theta}) \\ &= \frac{p(\boldsymbol{x}^{(r)}|\boldsymbol{x}^{(p)}, y = +1, \boldsymbol{\theta})p(y = +1|\boldsymbol{x}^{(p)}, \boldsymbol{\theta})}{p(\boldsymbol{x}^{(r)}|\boldsymbol{x}^{(p)}, \boldsymbol{\theta})} \\ &\propto p(\boldsymbol{x}^{(r)}|y = +1, \boldsymbol{\alpha}_{+1})p(y = +1|\boldsymbol{x}^{(p)}, \boldsymbol{\beta}), \end{aligned} \tag{3}$$

where $\boldsymbol{\alpha}_{+1}$ is the parameter of the $+1$ class-conditional distribution (likewise, $\boldsymbol{\alpha}_{-1}$ is the parameter of the -1 class-conditional distribution). The exact form of the class-conditional distributions can be formulated by certain models (such as normal distribution, mixture of Gaussians and naïve Bayes). The model parameter $\boldsymbol{\theta} = \{\boldsymbol{\alpha}_{+1}, \boldsymbol{\alpha}_{-1}, \boldsymbol{\beta}\}$ can be learned from training data $\{\boldsymbol{X}^{(p)}, \boldsymbol{X}^{(r)}, \boldsymbol{y}\}$ by maximum likelihood or maximum *a posteriori* estimation. For instance, CENTIPEDE is a Bayesian mixture model developed for the prediction of transcription factor binding sites [36] by using position weight matrices score, evolutionary conservation score and transcription start site proximity as prior knowledge, and ChIP-seq experiments as regular features. In case of multi-class problems (that is $y \in \{1, \ldots, C\}$ where $C > 2$), the class prior can be extended to a multinoulli distribution:

$$\begin{aligned} h_{\theta}(\boldsymbol{x}^{(p)}, \boldsymbol{\theta}) &= \begin{bmatrix} p(y = 1|\boldsymbol{x}^{(p)}, \boldsymbol{\theta}) \\ \vdots \\ p(y = C|\boldsymbol{x}^{(p)}, \boldsymbol{\theta}) \end{bmatrix} \\ &= \frac{1}{\sum_{c=1}^{C} e^{-\boldsymbol{\beta}_c^{\mathrm{T}}\boldsymbol{x}^{(p)}}} \begin{bmatrix} e^{-\boldsymbol{\beta}_1^{\mathrm{T}}\boldsymbol{x}^{(p)}} \\ \vdots \\ e^{-\boldsymbol{\beta}_C^{\mathrm{T}}\boldsymbol{x}^{(p)}} \end{bmatrix}. \end{aligned} \tag{4}$$

Then the parameters to be learned from training data become $\{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_C, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_C\}$.

Bayesian methods are well-known for their capability of incorporating various prior knowledge in predictive or exploratory models. However, it may be difficult to find useful information as prior features. Furthermore, it is often hard to assume proper class-conditional distributions, especially for complex systems. In case of many-class problems, finding a suitable class-conditional distribution for each individual class becomes unattainable in practice.

Bayesian methods for data of mixed types

Bayesian networks (BNs) [37] can naturally model multi-view data with mixed distributions for classification and feature-interaction identification purposes. As a typical model of probabilistic graphical models, a BN can be represented by $\{\boldsymbol{S}, \boldsymbol{\theta}\}$, where $\boldsymbol{S}$ denotes its graphical structure whose nodes represent variables and directed edges indicate dependencies between pairs of variables and $\boldsymbol{\theta}$ the set of parameters of the variables' conditional distributions. Suppose there are M visible variables (say $\boldsymbol{X} = [X_1, \ldots, X_M]$) and no latent variable, BN decomposes the joint probability of $\boldsymbol{X}$ to

$$p(\boldsymbol{X}) = \prod_{m=1}^{M} p(X_m|\Pi(X_m)), \tag{5}$$

where $\Pi(X_m)$ is the parent set of variable X_m. The dependency structure and parameters of the conditional distributions can be learned from data. Given a learned BN, the values of invisible variables (if any) can be inferred from (partially) observed data. The values of variables in a BN can be all discrete [38], all continuous [39] or a mixture of them [40]. Thus, variables of different types can be naturally integrated using BN. In the general procedure of applying BN for classification (and regression) purpose (1A), the network (with the class variable being a node) can be learned from labeled training data. The classes of unlabeled samples can be inferred using the learned network [41]. Furthermore, BN can be simultaneously used to select features by only taking variables in the Markov blanket of the class node. For example, in Figure 1A, features X_1, X_2, X_4, X_5 and X_6 form the Markov blanket of the class node; thus, they should be reported as key features. Moreover, prior (biological) knowledge can be easily incorporated in structure learning and parameter estimation of BNs [42].

However, three obstacles challenge us to apply BNs in data integration. First, searching for the optimal BN structure is a NP-complete problem [38]. Second, the number of parameters may be much larger than the sample size. Third, inference in a BN is intractable. They imply that using BN for high-dimensional data becomes exponentially difficult. Thus, heuristic structure learning algorithms and restrictions (or assumptions) on the model structure and conditional distributions are usually made to alleviate the curses of dimensionality. The model structure is often assumed to be sparse. For example, in naïve Bayes classifier [43] (Figure 1B), the features are assumed to be conditionally independent of each other given the class variable (say C) as common parent, so that the joint probability $p(C, \boldsymbol{X})$ can be factorized as

$$p(C, \mathbf{X}) = p(C) \prod_{m=1}^{M} p(X_m|C). \tag{6}$$

The class label can thus be inferred by

$$\frac{p(C|\mathbf{X}) = \frac{p(C,\mathbf{X})}{p(\mathbf{X})} = p(C) \prod_{m=1}^{M} p(X_m|C)}{p(\mathbf{X}).} \tag{7}$$

Naïve Bayes classifier is famous as a slim and swift BN model because learning its structure is needless and the inference of class label is straightforward. Tree-augmented naïve Bayes classifier [44] (Figure 1C) relaxes the independence among features by a tree structure. It outperforms the naïve Bayes classifier but keeps the efficiency of model learning.

In model selection of BNs, the Bayesian information (BIC) criterion and the minimum description length (MDL) criterion are asymptotically consistent [45]. Model averaging and single model selection with respect to a penalized criterion, such as BIC and MDL, help BNs to avoid overfitting of data [46]. In the presence of incomplete data, BNs is robust to missing values by ignoring them when computing sufficient statistics in parameter estimation.

Trees of mixed data types and ensemble learning

Decision trees should be considered as integrative models because a mixture of discrete and continuous features can be simultaneously integrated [47] without the need to normalize features. Classification and regression trees are representatives of rule-based prediction models. When recursively building a classification tree, a feature (even a subset of features) that splits the classes the best, in term of a scoring function, is selected to create a node. At each node, rules are established to branch out different classes downward. Different from black-box models, the learned hierarchy of rules (tree) are well interpretable. Meanwhile, decision trees can be applied to select features by ranking attributes with respect to their summed improvements for class purity. In a decision tree with T internal nodes, the importance score of the i-th feature can be defined by $s(X_i) = \sum_{t=1}^{T} g(t) I(v(t) = i)$, where $I(v(t) = i) \in \{0, 1\}$ indicates whether the i-th feature is selected in the t-th node to split the corresponding data region and $g(t)$ is the gain of class purity measured, for example, by Gini index [47, 48]. Because each feature is used to learn decision rules, multi-view data of various data types (discrete, categorical and continuous) can be considered together without normalization. The values of continuous variables are partitioned into intervals of different lengths, thus decision rules can be created for continuous variables of a variety of distributions without the need to standardize input data. In fact, decision trees remain invariant under feature scaling and transformation. However, decision trees are sensitive to noise, thus have a poor capability of generalization. Moreover, building a decision tree for high-dimensional data could consume an unaffordable amount of time.

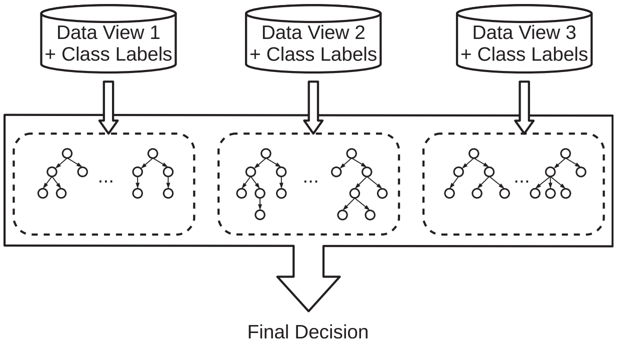

Figure 2. Late integration of multi-view data using ensemble learning (e.g. random forest).

The overfitting issue of decision trees can be overcome by collective intelligence, that is, ensemble learning [49, 50], which builds a population of decision trees as weak learners for the state-of-the-art performance. Bagging [51] and boosting [52, 53] are popular ensemble learning models, where bagging simply combines the decisions of multiple weak learners, while boosting tweaks the weak learners to focus on hard examples. However, a few trees learned in this manner may be highly correlated. Moreover, learning a collection of decision trees for multi-view data with many features becomes much more unaffordable.

Random forest addresses the above two challenges by randomly picking up features in the construction of trees. Although the randomness degrades the interpretability, the importance of features can still be obtained by out-of-bag (OOB) randomization or Gini index. In the former method, the importance score of the i-th feature is defined as the difference of OOB errors between using the original OOB samples and using the OOB samples where the values of the i-th feature are permuted. In the latter method, Gini indices [47] of the i-th feature in individual trees in the forest can be averaged as the importance score [48]. Although widely used in practice, many properties of random forest models stay unknown. It has been just shown that random forests are resistant to outliers [54]. Recent studies also prove that not all random forest models are universally consistent [55, 56].

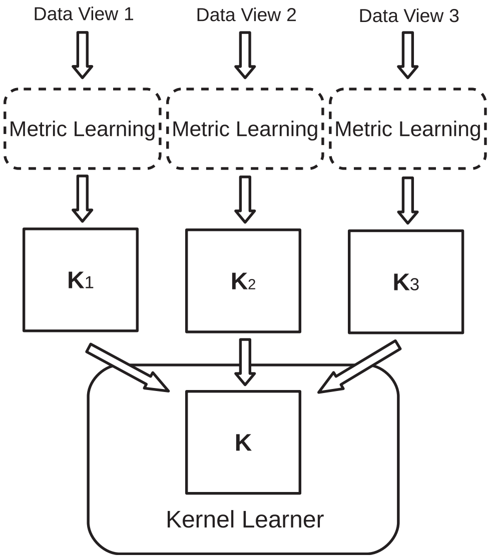

Figure 3. Multiple kernel learning for data integration. Metric learning may be applied to learn suitable similarity matrices.

There are three ways to integrate data by ensemble learning. The first way is to use the concatenated features as input of random forest. The second way is to build multiple trees for each data view, and then use all learned trees of all views to vote for the final decision [57, 58]. An example of using random forest as a late integration method is illustrated in Figure 2. More elegant combination methods are discussed in [59]. This ensemble-learning-based data integration strategy has several advantages. First, this method can be easily manipulated and its outcomes are well interpretable. Second, class imbalance problems can be elegantly addressed by random forest in its bootstrapping [60]. Third, granularity of features can be carefully considered in the step of sampling features [61]. However, because it is a late-integration strategy, the interactions of features from separate sources cannot be detected. The third way is to obtain new meta-features from multi-view data instead of using the original features. This idea is from West's group who incorporate both clinical factors and genomic data in predictive survival assessments [62]. A meta-feature (named meta-gene in [62]) is defined as the first principal component of a cluster of genes grouped by a clustering algorithm. Then, the model grows a forest of statistical classification and prediction trees. In each tree, features used in the nodes are decided by the significances of Bayesian factor tests on the features (meta-genes and clinical factors). Multiple significant features can be distributed in multiple trees so that the correlations between trees are reduced. The final decision is determined by a weighted combination of the decisions of all trees, where the probabilities of trees are used as combination weights. One advantage of the meta-feature-based ensemble model is that information from different views can be incorporated via clustering in the model learning. Because the meta-features are used instead of the original features, complexity of trees can thus be reduced.

Kernel learning and metric learning

Kernel matrix $k(\mathbf{X}, \mathbf{X})$, rather than the original features, is only required as input of kernel classification (e.g. SVM [24]), regression (e.g. support vector regression [63]), clustering (e.g. spectral clustering [64]) and feature extraction (e.g. sparse representation, SR [65]) methods. Thus, the problem of data integration can be transformed to kernel integration in the sample space rather than the heterogeneous feature space. Multiple kernel learning (MKL) [66, 67] is an intermediate integration technique that first computes kernel (or similarity) matrices separately for each data view, then combines these matrices to generate a final kernel matrix to be used in a kernel model. Suppose there are K kernel functions, $\{k_1(\cdot,\cdot), \ldots, k_V(\cdot,\cdot)\}$, corresponding to V views, $\mathbf{X} = \{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(V)}\}$, where matrix $\mathbf{X}^{(v)}$ represents samples (in columns) using the v-th set of features (in rows). The combined similarity between samples $\mathbf{x}_i$ and $\mathbf{x}_j$ can be computed by

$$k(\mathbf{x}_i, \mathbf{x}_j) = f_{\boldsymbol{\eta}}(k_1(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(1)}), \ldots, k_V(\mathbf{x}_i^{(V)}, \mathbf{x}_j^{(V)})), \tag{8}$$

where $f_{\boldsymbol{\eta}}$ is either a linear or nonlinear function with parameter $\boldsymbol{\eta}$. In the simplest case, it is a weighted linear combination of kernel matrices:

$$k(\mathbf{X}, \mathbf{X}) = \sum_{v=1}^{V} \eta_v k_v(\mathbf{X}^{(v)}, \mathbf{X}^{(v)}), \tag{9}$$

where $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_V]$ can be either assigned before learning, or determined in the learning procedure. Figure 3 shows an example of MKL-based integration system. The individual similarity (kernel) matrix of a data view can be computed by an off-the-shelf kernel function semantically sensible for the specific view or by sophisticated metric learning [68]. Metric learning [69, 70, 71] aims to learn a metric function from data such that the distances between within-class samples are closer, whereas the distances between inter-class samples are farther. A key strength of kernel methods is that their optimizations are independent of the number of features, which is known as dimension free [72]. However, large-scale optimization corresponding to a large sample size remains a major bottleneck. For example, an optimal MKL task can essentially be a semidefinite programming problem [73]. Moreover, because a kernel method is treated as a 'black box' with respect to identifying informative features and the integration occurs in sample space, MKL should not be considered when identifying feature interactions. SimpleMKL-based [74] multiple kernel learning has been applied in [75] to predict cancer prognosis by using gene expression data, DNA methylation data, miRNA expression data and copy number variations, showing significantly better auROC over single-view methods.

Network-based approaches to integrate multiple homogeneous networks

Multi-view data of cohort samples can be integrated in the sample space by network fusion methods. Although these methods are essentially nonlinear MKL models, as they are mainly presented in the context of biological network mining, we discuss them in this separate section. Given multiple networks with identical nodes but different edges, an established idea is to fuse these networks in a final network reflecting common and view-specific connections. As a typical example of this principle, SNF [76] integrates mRNA expression, DNA methylation and miRNA of cohort cancer patients for tumor subtyping and survival prediction. SNF first constructs a sample-similarity network for each view where a node represents a sample and a weighted edge reflects the similarity between two samples. Then, it uses a message-passing-theory-based method to iteratively update each network making it more similar to the other networks. Finally, a patient-similarity network is generated by fusing all individual networks. Essentially, this network-based method falls into the kernel learning principle discussed in the 'Kernel learning and metric learning' section. Kernel clustering methods, for example, spectral clustering, can be applied on the patient-similarity network to find distinct groups of patients, that is, subtyping. Comparison assessment on multi-omics cancer data showed that SNF outperforms iCluster [77] and feature concatenation in terms of Cox log-rank test and a cluster index score.

Similarly, using this principle, multi-view data of cohort features can be integrated in the feature space. Here, we take the reconstruction of gene regulatory networks (GRNs) as an example. Given multiple RNA-seq data sets generated under different perturbations of the same cell system, a GRN can be learned for each data set, and then fused by SNF to a final GRN. Likewise, GRNs learned by different algorithms can also be fused by a method like SNF to achieve robust result, which implements the philosophy—'the wisdom of crowds' [78]. A similar network fusion method has been proposed in KeyPathwayMinerWeb for pathway enrichment analysis using multi-omics data [79]. Another similar method is called 'FuseNet', which models multi-omics data using Markov network allowing non-Gaussian distributions, and represents the model parameters by shared latent factors to collectively infer a holistic gene network [80].

To learn knowledge from multiple networks with identical nodes but different edges, instead of fusing these networks in a final network, Wu's group recently proposed an energy-based approach to integrate information from multiple networks [81, 82, 83, 84, 85]. Considering a Boolean network with N vertices, there are totally 2^N different configurations. In such a configuration, a node can be labeled as either 1 or 0. If all genes related to a specific disease in a gene network are labeled as 1 and other as 0, then a specific disease corresponds to a specific configuration of the gene network. The probability of the configuration can be calculated by a Boltzmann distribution function $p(\mathbf{X})$ with respect to the network configuration as follows:

$$p(\mathbf{X}) = \frac{1}{Z}e^{-kH(\mathbf{X})}, \tag{10}$$

where $H(\mathbf{X})$ is the Hamiltonian (energy) of the configuration $\mathbf{X}$, k is a constant parameter and Z is called the partition function and defined as $Z = \sum_{\mathbf{X}\in\{\mathbf{X}\}} e^{-kH(\mathbf{X})}$, where $\{\mathbf{X}\}$ is a set consisting of all possible configurations of a network. The energy function $H(\mathbf{X})$ depends on the network structure and the labeling of the configuration, which can be mathematically expressed as

$$H(\mathbf{X}) = -\alpha N_1 - \beta N_{10} - \gamma N_{11} - N_{00}, \tag{11}$$

where N_1 is the number of nodes with label 1, N_{00} is the number of edges connecting neighboring nodes with both 0's, N_{10} is the number of edges connecting a node with label 0 and another with 1, N_{11} is the number of edges bridging two adjacent nodes with label 1 and α, β and γ are parameters to be learned. Letting $\theta = \{\alpha, \beta, \gamma\}$, we can calculate the posterior probability of node n labeled as 1 as follows:

$$p(X_n = 1|X_{-n}, \theta) = \frac{e^{\alpha+(\beta-1)M_{n0}+(\gamma-\beta)M_{n1}}}{e^{\alpha+(\beta-1)M_{n0}+(\gamma-\beta)M_{n1}} + 1}, \tag{12}$$

where M_{n0} and M_{n1} are the numbers of neighbors of node n with label 0 and 1, respectively. For multiple networks, although the labelings of the configuration are the same, their network connections are different. As a result, the energy functions of a specific configuration in multiple networks vary, yet are naturally additive. Suppose that we have V gene networks, the energy of a specific configuration with multiple networks can be calculated as

$$\begin{aligned} H(\mathbf{X}) &= \sum_{v=1}^{V} H^{(v)}(\mathbf{X}) \\ &= -\alpha N_1 - \sum_{v=1}^{V}(\beta^{(v)}N_{10}^{(v)} + \gamma^{(v)}N_{11}^{(v)} + N_{00}^{(v)}). \end{aligned} \tag{13}$$

Accordingly, the posterior probability of node n labeled as 1 can be calculated by

$$p(X_n = 1|X_{-n}, \Theta)$$

$$\frac{= e^{\alpha+}\sum_{v=1}^{V}(\beta^{(v)} - 1)M_{n0}^{(v)} + (\gamma^{(v)} - \beta^{(v)})M_{n1}^{(v)}}{e^{\alpha+}\sum_{v=1}^{V}(\beta^{(v)}-1)M_{n0}^{(v)}+(\gamma^{(v)}-\beta^{(v)})M_{n1}^{(v)}} + 1, \tag{14}$$

where $\Theta = \{\alpha, \beta^{(v)}, \gamma^{(v)}, v = 1, \ldots, V\}$. Given a set of known genes related to a specific disease, one can learn the parameters in model (12) with a single network or (14) with multiple networks by Gibbs sampling algorithms [81, 82] or maximum likelihood algorithms [83, 84]. Then the probability that an undetermined gene is related to the specific disease can be calculated by model (12) or (14). In both models, only the information of direct neighbors is considered. The results in [81, 84] showed that the method with integration of multiple networks outperforms that with single networks in terms of auROC. To include the information of indirect neighbors, a graph-kernel-based method is developed in [85]. The comparison in [85] reported its superior performance over those methods only using information of direct neighbors.

Network-based methods for fusing multiple relational data

In association studies such as gene–disease associations and genotype–phenotype associations, the relation between two types of objects can also be represented by a matrix denoted by $\mathbf{X}$ where $x_{i,j}$ indicates the strength of relation between objects i and j, or by a network where nodes represent objects and (weighted) edges indicate the presence of associations. Thus, the association problems, can be solved by either kernel (relational) matrix factorization methods or graphical methods, even a mixture of both. Based on the number of relationships, association studies can be categorized to two-relational or multi-relational problems.

In two-relational association studies, the challenge is how to integrate multiple relational (or adjacency) matrices of two sets of biological entries. For example in gene–disease studies, the question is how to integrate multiple known gene–disease relational matrices obtained by different measurements for inferring new relevant candidate genes or candidate diseases given a pivot set of genes. The kernelized Bayesian matrix factorization method is an effective method to infer a bipartite graph by multiple data source integration [86, 67]. Recently, Lan *et al.* [87] have used this method to infer potential miRNA–disease associations by integrating sequence and functional information of miRNA with semantic and functional information of diseases. Their experimental results demonstrated that this method could not only effectively predict unknown miRNA–disease

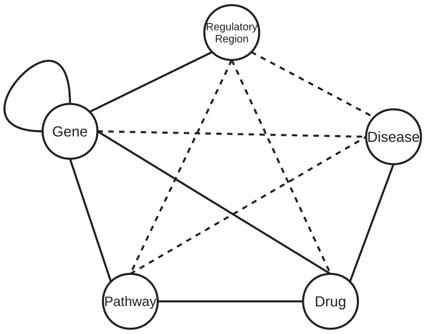

Figure 4. An example of multiple association studies represented in a multiplex heterogeneous network where each node represents a set of objects of the same type, and each edge represents an association matrix. Gene–gene, RR–gene, gene–drug, gene–pathway and drug–disease associations are given (marked in bold lines), whereas associations such as RR–disease and gene–disease associations are yet to be inferred (marked in dashed lines). There might exist multiple paths for indirect associations, for example in RR–disease associations, we can have RR-gene-pathway-drug-disease and RR-gene-drug-disease.

associations, but also outperformed its competing methods in terms of auROC.

A heterogeneous network can be constructed to represent homogeneous associations (such as gene–gene associations) in homo-networks and heterogeneous associations (such as gene–disease associations) in heter-networks. Given a network, a random walk is a path that starts at a prespecified node and randomly moves to its neighbor, then to neighbor's neighbor, and so on. Random walks can explain the observed behaviors of many random processes and thus serve as a fundamental model for the recorded stochastic activities. Random walk methods have been applied on either two-relational heterogeneous networks (such as gene–phenotype associations [88], drug–target interactions [89] and miRNA–disease associations [90, 91]) or multi-relational heterogeneous networks (for example, drug–disease associations [92]) to infer novel candidate relations.

Tri-matrix factorizations, combined with network methods, are found useful in association studies of multiple sets of biological entries, where pair-wise associations are represented in relational matrices. Figure 4 is such an example of multi-relational associations, where each node represents a set of homogeneous objects and each edge represents a relational matrix. Data fusion approach with penalized matrix tri-factorization (DFMF) [93] is a model that can integrate multi-relational data. Given multiple sets of distinct biological objects $\{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_V\}$, and their relations represented in matrices $\mathbf{X}^{i,j} \in \mathbb{R}^{N_i \times N_j}$ or $\mathbf{X}^{i,j} \in \mathbb{R}_+^{N_i \times N_j}$, where $i, j \in \{1, 2, \ldots, V\}$, the basic idea is to decompose each given pair-wise association matrix as $\mathbf{X}^{(i,j)} \approx \mathbf{W}^{(i)}\mathbf{S}^{(i,j)}(\mathbf{W}^{(j)})^{\mathrm{T}}$, where rows of $\mathbf{W}^{(i)}$ and $\mathbf{W}^{(j)}$ are the latent factors for object sets $\mathcal{E}_i$ and $\mathcal{E}_j$, respectively, $\mathbf{S}^{(i,j)}$ governs the interactions between $\mathcal{E}_i$ and $\mathcal{E}_j$. DFMF can only infer new associated objects in a directly given pair of associations, which is essentially a matrix completion problem. Compared with MKL, random forest and relational learning by matrix factorization, DFMF achieved higher auROC for prediction of gene function and prediction of pharmacologic actions. The methodology used in Meduca [94] extends DFMF to a solution for inferring the most significant size-*k* modules of objects indirectly associated to a given set of pivot objectives in a multiplex of association data. As illustrated in Figure 4, given RR-gene (RR stands for regulatory region), gene–pathway, gene–drug, disease–drug associations, the Meduca method can addresses two questions, for example: (1) Given a pivot set of RRs associated to a subtype of breast cancer, how to detect other RRs that are also significantly associated to this tumor subtype [the so-called candidate-pivot-equivalence (CPE) regime]; (2) Given a pivot set of diseases, how to find the relevant RRs [the so-called candidate-pivot-inequivalence (CPI) regime]. To realize this, Medusa first uses collective matrix factorization (i.e. the DFMF model [93]) to generate latent data matrices for individual associations, and then produces a connection matrix chaining a chosen set of latent data matrices from the source objects to the target objects. For CPE problems, a significant size-*k* module can be obtained based on connections. For CPI problems, the evaluation of candidate objects is based on visibility. Because there are multiple paths from the source objects to the target objects, Medusa combines all possible connection matrices to compute the scores of each candidate objects for the evaluation of size-*k* modules. In the prediction of gene–disease associations, Medusa obtained higher auPRC and auROC than random walk.

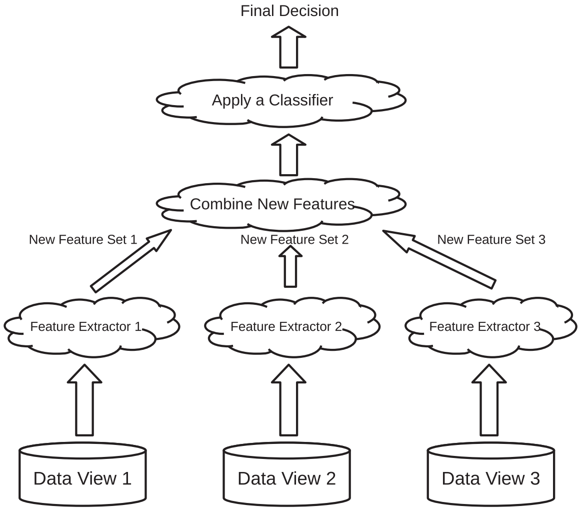

Figure 5. An integrative predictive model based on separate view-wise feature extraction.

Feature extractions and matrix factorizations for detecting shared and view-specific components

While it is often challenging to combine features of multiple views in the original input spaces, new features generated by feature extraction methods can be easily combined. As illustrated in Figure 5, an idea is to extract new features from each data view first, and then incorporate these new features together. Finally, a classification or clustering algorithm can be applied on the combined features. Depending on the nature of an individual data view, a feature extraction method learns the representations of samples in a new feature space. Matrix factorization methods, such as principal component analysis [95, 96], factor analysis (FA) [97, 98], NMF [99, 100, 101, 102], SR [65] and tensor decomposition methods [103, 104], are commonly used feature extraction models. Other dimensionality reduction methods than matrix decompositions, such as autoencoder [105] and restricted Boltzmann machine (RBM) [106], can be applied as well. There are several benefits of using feature extraction in data integration. First, the natures of heterogeneous data from multiple omics data can be separately well counted. Despite the original data types, the new features in the

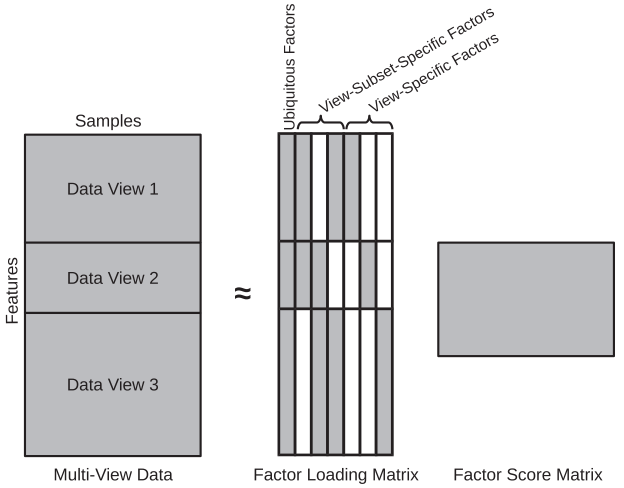

Figure 6. Data integration based on Bayesian group factor analysis. Zero blocks are marked in white.

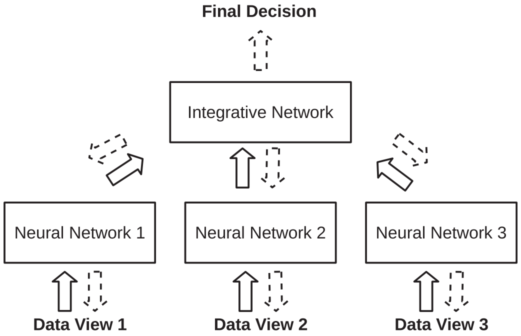

Figure 7. Additive multi-modal deep learning for data integration. Different deep learning models can be applied, as sub-networks, to individual data views. An integrative network combines information from the sub-networks. The model can be either directed or undirected; either supervised or unsupervised. Bottom-up arrows indicate a discriminative model. Downward or undirected connections indicate a generative model.

corresponding feature spaces are usually numeric, implying an easy concatenation. Second, the high-dimensionality is dramatically reduced so that the downstream analysis will be more efficient. Third, extracting new features separately for each data view implements the principle of divide and conquer, thus computational complexity can be significantly reduced. Fourth, relational data can be well incorporated by kernel feature extraction methods [107]. However, one pitfall of the feature-extraction-based integrative principle is that the interactions (correlation, dependency or association) between input features from different views cannot be taken into account in the separated feature extraction procedures.

To consider the interactions between features from different views, (Bayesian) multi-view matrix factorization methods can be applied to extract new features on the feature-wise concatenated matrix. By inducing group-wise (that is view-wise) sparsity on the basis matrix (that is factor loading matrix), Bayesian group factor analysis (GFA) [108, 109] detects ubiquitous and view-specific factors (see Figure 6), which is informative to discover features from multiple views involved in a potential pathway. As a special case of GFA, Bayesian canonical correlation analysis (CCA) [110] only detects correlated factors between two views. Sparse GFA has been developed for biclustering multi-view data with co-occurring samples [111]. Simulation study revealed that sparse GFA could more accurately recover predefined biclusters in terms of F_1 score compared with factor analysis for bicluster acquisition [112]. Sparse GFA was applied to predict drug response of cancer cell lines supported by multi-omics data, where the prediction was modeled as inferring missing values (drug response), cross-validation performance was measured by correlation coefficients and real predictions were validated by enrichment analysis.

NMF has been extended for multi-view data for clustering and latent FA. Similar to iCluster [77], MultiNMF [113] and EquiNMF [114] allow all views to share a unique coefficient matrix for collective clustering, while Joint-NMF [115] restricts all views to share the same basis matrix for finding common factors among multi-omics data. Using ubiquitous basis matrix and view-specific basis matrices as well as view-specific coefficient matrices, integrative-NMF [116] is able to detect homogeneous and heterogeneous factors from multi-omics data. Comparative studies showed that integrative-NMF significantly outperforms joint-NMF on both a simulated data set with heterogeneous noise in terms of a module-detection score and a real ovarian cancer multi-omics data set in terms of purity indices. The discovered modules were validated using pathway enrichment analysis. Although not in a non-negative setting but closely relevant, joint and individual clustering (JIC) [117] and joint and individual variation explained (JIVE) [118] use both common and view-specific coefficient matrices. Comparison showed that JIC is advantageous over iCluster on both simulated data in terms of precision and multi-omics (RNA-seq and miRNA-seq) breast cancer data in terms of validation using clinical information. This idea can be easily generalized to any matrix factorizations, including NMF. Because high-throughput sequencing platforms generate read-count data, which are naturally non-negative, multi-view NMF models have a great potential for various analyses such as tumor subtyping, pathway analysis and biomarker selection.

Partial least squares (PLS, more precisely it should stand for projection to latent structures) is a long-standing model that integrates two-view data together [119]. Given centered predictor data $\mathbf{X} \in \mathbb{R}^{M^{(X)} \times N}$ and the corresponding response data $\mathbf{Y} \in \mathbb{R}^{M^{(Y)} \times N}$ where $M^{(X)}$ is the number of predictor variables, $M^{(Y)}$ the number of response variables and N the number of samples, it projects $\mathbf{X}$ and $\mathbf{Y}$ into two new K-dimensional spaces, spanned by columns of $\mathbf{W}^{(X)} \in \mathbb{R}^{M^{(X)} \times K}$ and $\mathbf{W}^{(Y)} \in \mathbb{R}^{M^{(Y)} \times K}$, respectively, where the covariance between their representations $H^{(X)}$ and $H^{(Y)}$ are maximized. A typical PLS can be formulated as

$$\mathbf{X} = \mathbf{W}^{(X)}\mathbf{H}^{(X)} + \mathbf{E}^{(X)}, \tag{15}$$

$$\mathbf{Y} = \mathbf{W}^{(Y)}\mathbf{H}^{(Y)} + \mathbf{E}^{(Y)}, \tag{16}$$

where $\mathbf{W}^{(X)}$ and $\mathbf{W}^{(Y)}$ denote orthonormal basis matrices with K columns, $\mathbf{H}^{(X)}$ and $\mathbf{H}^{(Y)}$ symbolize coefficient matrices, as well as $\mathbf{E}^{(X)}$ and $\mathbf{E}^{(Y)}$ are error terms. By making $\mathbf{W}^{(X)}$ and $\mathbf{W}^{(Y)}$ sparse, PLS can be applied to select key predictor variables relevant to the response variables. For example, the predictor variables corresponding to non-zero elements in the k-th column of $\mathbf{W}^{(X)}$ can be treated as important features to explain the response variables corresponding to non-zero elements in the k-th column of $\mathbf{W}^{(Y)}$. The attraction of PLS-based integrative models is their capability of selecting co-varying features among multiple feature sets [120]. Orthogonal projections to latent structures is an extension of PLS that improves interpretability by removing

Table 2. Implementations of machine learning methods for multi-view data analysis

Method	Tool name [Ref.]	Functionality	URL	Language
Feature concatenation	glmnet [155]	LASSO, elastic net	cran.r-project.org/web/packages/glmnet	R
	scikit-learn [156]	LASSO, elastic net, SVM	scikit-learn.org	Python
	grplasso [157]	group LASSO	cran.r-project.org/web/packages/grplasso	R
	SGL	Sparse group LASSO	cran.r-project.org/web/packages/SGL	R
	SPAMS [158]	(Sparse) group LASSO using proximal algorithms	spams-devel.gforge.inria.fr	R, Python, MATLAB
	ParsimonY	Overlapping group LASSO	github.com/neurospin/pylearn-parsimony	Python
	glasso [28]	Graphical LASSO	cran.r-project.org/web/packages/glasso	R
Bayesian models or networks	bnlearn [159]	Bayesian network learning and inference; does not support mixed types; naïve Bayes and tree-augmented naïve Bayes classifiers	cran.r-project.org/web/packages/bnlearn	R
Ensemble learning	Random forest [160]	random forest	cran.r-project.org/web/packages/randomForest	R
	scikit-learn [156]	random forest	scikit-learn.org	Python
Multiple kernel learning	Mklaren [161]	Simultaneous multiple kernel learning and low-rank approximation	github.com/mstrazar/mklaren	Python
	LibMKL [162]	Soft margin MKLs	sites.google.com/site/xinxingxu666	MATLAB
	SimpleMKL [74]	MKL SVMs	asi.insa-rouen.fr/enseignants/arakoto/code/mklindex.html	MATLAB
	GMKL [163]	Generalized MKL based on gradient descent and SVM	research.microsoft.com/en-us/um/people/manik/code/GMKL/download.html	MATLAB
Network-based method	SNF [76]	Similarity network fusion	compbio.cs.toronto.edu/SNF	R, MATLAB
	KeyPathwayMiner [79]	Extract all maximally connected sub-networks	tomcat.compbio.sdu.dk/keypathwayminer	Java
	FuseNet [80]	Infer networks from multi-omics data	github.com/marinkaz/fusenet	Python
	scikit-fusion [93]	Data fusion based on DFMF	github.com/marinkaz/scikit-fusion	Python
	Medusa [94]	Collective-matrix-factorization-based indirect association discovery	github.com/marinkaz/medusa	Python
Multi-view matrix or tensor factorization	pls [164]	Partial least squares and principal component regression	cran.r-project.org/web/packages/pls	R
	spls [165]	Sparse PLS regression and classification; simultaneous dimension reduction and variable selection	cran.r-project.org/web/packages/spls	R
	O2PLS [166]	O2-PLS	github.com/selbouhaddani/O2PLS	R
	K-OPLS [167]	Kernel-based PLS	kopls.sourceforge.net	R, MATLAB
	CCAGFA [109]	GFA and CCA	cran.r-project.org/web/packages/CCAGFA	R
	GFAsparse [111]	Sparse GFA for biclustering	research.cs.aalto.fi/pml/software/GFAsparse	
	iCluster [77]	Integrative clustering of multiple genomic data types	cran.r-project.org/web/packages/iCluster	R
	r.jive [118]	JIVE	cran.r-project.org/web/packages/r.jive	R

(continued)

Table 2. Continued

Method	Tool name [Ref.]	Functionality	URL	Language
	iNMF [116]	Integrative NMF	github.com/yangzi4/iNMF	Python
	MVMF	Multi-view NMFs for feature pattern discovery from multi-class data	github.com/yifeng-li/mvmf	Python
	Tensor Toolbox [168]	Operations of multi-way arrays	www.sandia.gov/tgkolda/TensorToolbox	MATLAB
	N-way Toolbox [169]	Multi-way PARAFAC, PLS, and Tucker models	www.models.life.ku.dk/nwaytoolbox	MATLAB
	Sparse PARAFAC [170]	Sparse PARAFAC	www.models.life.ku.dk/sparafac	MATLAB
	CMTF [171]	Coupled matrix and tensor factorization	www.models.life.ku.dk/joda/CMTF_Toolbox	MATLAB
	NTFLAB	Non-negative tensor factorizations	www.bsp.brain.riken.jp/ICALAB/nmflab.html	MATLAB
Multi-modal learning	multimodal [144]	Multi-modal DBMs	www.cs.toronto.edu/nitish/multimodal, github.com/nitishsrivastava/deepnet	Python

from $\mathbf{X}$ the systematic variation that is not correlated to $\mathbf{Y}$ [121]. O2-PLS further generalizes this idea in a predictive model in both ways, namely, $\mathbf{X} \leftrightarrow \mathbf{Y}$, by modeling the $\mathbf{Y}$-orthogonal variation in $\mathbf{X}$, the $\mathbf{X}$-orthogonal variation in $\mathbf{Y}$ and the $\mathbf{X}$-$\mathbf{Y}$ joint co-variation [122, 123]. O2-PLS has been applied to integrate transcript and metabolite data in plant [124]. OnPLS extends O2-PLS for more than two views by using a global one-versus-one strategy [125, 126] (other PLS-related models are also discussed in [125]). It extracts shared components maximally co-varying across all views and local components specific to combinations of views or individual views. It has been applied to integrate transcriptomic, proteomic and metabolomic data for discovering shared pathways [127]. Indeed, pathway analysis corroborated that the computationally identified pathways are biologically meaningful. In case of strongly nonlinear associations between $\mathbf{X}$ and $\mathbf{Y}$, kernel-based orthogonal projections to latent structures (K-OPLS) was proposed to carry out PLS on a high-dimensional feature space [128].

In addition to matrix factorization models, their extensions—tensor decompositions [129, 103, 130] should be also considered for dimensionality reduction (in classification or clustering) and FA on multi-view data naturally represented by a tensor [131].

Many matrix decomposition models are non-convex from the optimization perspective. Thus, their performances are affected by initial values. Moreover, feature selection based on l_1-norm regularized sparse matrix factorizations may suffer from inconsistency. Applying the strategy of stability selection on matrix factorizations can make the procedure consistent [132]. Matrix or tensor factorization models assuming Gaussian or Poisson distributions to the data are not robust to outlying data points. Robust factorizations, such as robust PLS [133], robust NMF [134] and tensor factorization using l_1-norm [135], are developed to address this issue. In the near future, more work needs to be done on robust multi-view matrix factorizations. Missing values can be nicely handled by weighted matrix factorizations [101] and Bayesian matrix factorizations [136, 137], which can ignore missing entries in their learning. Future developments of new multi-view matrix factorization tools should consider the functionality of dealing with missing values. Overfitting is not an issue for penalized and Bayesian multi-view matrix factorizations.

Multi-modal deep learning

The deep neural network-based [138] multi-modal structure, illustrated in Figure 7, is another option to integrate multi-view data with heterogeneous feature sets, and capture their high-level associations for prediction, clustering and handling incomplete data. The basic idea is to select a specific sub-network for each view, and then integrate the output of individual sub-networks in higher layers. The sub-networks provide the flexibility of choosing appropriate deep learning models respectively for individual data views, such as deep belief net (DBN) [139] or deep Boltzmann machine (DBM) [140] for binary, Gaussian or count data, convolutional neural network [141] for image data, recurrent neural network [142] for sequential signal and deep feature selection (DFS) [143] for choosing discriminative features. The sub-networks can be either directed or undirected. The whole model can be supervised or unsupervised. The work in [144] is a typical example of multi-modal learning for image-text data. In this model, a Gaussian-Bernoulli DBM and a replicated softmax DBM are respectively constructed for continuous image data and word-count data; a top layer connects both sub-networks to learn their joint representation. This multi-modal DBM can be applied to generate image given text, generate text given image, classify or cluster samples based on joint representations of multi-view samples. A multi-modal DBN has emerged in bioinformatics to integrate gene expression, DNA methylation and drug response for tumor subtyping [145], showing superior performance to k-mean clustering in terms of clinical discrepancy (survival time).

Multi-modal deep neural networks have five attractive strengths for data integration. First, when learning model parameters, the sub-networks can be pretrained using different data views, separately, then the parameter of the entire network (including the integrative layers and sub-networks) can be globally fine-tuned. Thus, this component-wise learning can significantly reduce the cost of computation. Second, the heterogeneous information from different views can be jointly considered well in the integrative layers for inference, classification and clustering [146]. Third, multi-modal networks can even learn on samples with missing views [144], which enables the maximal use of available data instead of merely using samples with complete views. Furthermore, a well-trained generative multi-modal network, such as DBM, can be used to infer profiles of missing views given some other observed views from an individual, which is quite interesting, for instance, for predicting the impact of genetic variations and epigenetic changes to gene expression. Last but not least, the flexible and deep structure of multi-modal learning is appropriate to model complex systems, thus has a great potential to make a full use of genomic data observed in various molecular levels.

The consistency of neural networks is well studied. The universal approximation theorem tells us that a feed-forward neural network having one hidden layer with proper parameters is able to approximate any continuous function [147, 148]. The great predictive power of multi-modal deep learning lies in its capabilities of modeling complex probability distributions and capturing high-level semantics. The limits of RBMs and DBMs to approximate probability distributions are still under investigation. Robust deep learning models can be developed by using robust cost functions for data with outliers [149]. Some experimental studies have reported that deep learning models are robust to noise [150, 151]. Overfitting is a problem for deep neural networks owing to their complex structures and large amount of parameters. Regularization (using l_1- and l_2-norms) and model averaging such as dropout [152] are effective techniques to avoid this issue. In addition, how to precisely catch and explicitly interpret inter-view feature interactions remains an open problem. The generalization of shallow multi-view matrix factorization methods, discussed in the 'Feature extractions and matrix factorizations for detecting shared and view-specific components' section, to their deep versions poses a new challenge. Finally, it should be noted that, beside additive integrative deep learning models, other strategies, such as multiplicative and sequential integration, exist [153].

Discussion and conclusion

Recent developments in many bioinformatics topics, such as cancer diagnosis, precision medicine and health-informatics systems have a keen need for integrative machine learning models to incorporate all available data for a better insight into complex biological systems and a more precise solution. In this review, we have investigated a variety of data integration principles from a machine learning perspective. Their basic ideas, structures, asymptotic consistency, robustness, risk of overfitting, strengths and limitations are discussed, respectively. These methods, particularly multi-view matrix factorizations and multi-modal deep learning, will revolutionize the way of using information and play a key role in integrative bioinformatics.

Multi-omics data measured for the same set of patients (or samples) are key to identify disease subtypes and pathways. However, such data are almost unavailable in diseases other than cancers in the current moment. In studies of complex diseases, such as multiple sclerosis, schizophrenia and autism spectrum disorder, there exist some independent RNA-seq and whole genome (or exome) sequencing data, enabling investigations either in transcriptomic level or genetic level. Compared with cancers, the signals in complex diseases might be too weak to be identified. Ideally large-scale multi-omics data of the same batch of patients are necessary for comprehensive analysis in different molecular levels [154]. The difficulty of obtaining tissues and lack of funding and human resources are the main challenges to generate such data. Therefore, researchers in non-cancer studies are eagerly suggested to switch their focus to integrative analysis and work in a coordinated manner to ensure the quality and completeness of multi-platform omics data.

Feature selection methods should be carefully chosen according to the purpose of a specific analysis. Owing to the nature that biological data often have highly correlated features, a set of relevant features selected to computationally optimize the power of prediction may not make sense in biological causalities. If features are selected to solely pursue the highest classification performance, l_1-norm regularized sparse models (e.g. LASSOs), sparse PLS, DFS or random forest should be considered. However, if one wants to globally examine the behavior of all features in multiple classes, BNs, feature clustering or multi-view matrix factorizations for feature pattern discovery should be taken into account.

We list open-source packages and tools for the seven categories of integrative models in Table 2. They are mainly implemented in Python, R and MATLAB, among which Python can serve as a promising platform to realize integrative models because (1) a multi-dimensional array is passed to a function by reference (while array arguments are passed by value in R and MATLAB), which is critical for big data; (2) the friendly object-oriented programming paradigm enables the development of large packages; and (3) the support of machine learning (particularly deep learning) packages facilitates the implementation of intelligent integrative methods. Even though multi-modal neural networks are potentially useful in the fusion of multi-view data, satisfactory software packages are still not available yet. Thus, a generic and comprehensive multi-modal package is eagerly expected in the near future, so that bioinformaticians can conveniently choose suitable types of sub-networks and define model structures.

Finally, we hope this review will provide a guide for bioinformaticians to select suitable tools corresponding to specific problems. We also expect that machine learning engineers and biological data scientists can be inspired by this discussion to develop and share their own novel approaches, to push forward the study of integrative biological data analysis.

Key Points

- We provide a comprehensive review on biological data integration techniques from a machine learning perspective.
- Bayesian models and decision trees are discussed for incorporating prior information and integrating data of mixed data types.
- Tri-matrix factorizations and network-based method are reviewed for two-relational and multi-relational association studies.
- Multi-view matrix factorization models are investigated for detecting ubiquitous and view-specific components from multi-view omics data.
- Multi-modal deep learning approaches are discussed for simultaneous use of multiple data sets in supervised and unsupervised settings.

Acknowledgements

We greatly appreciate the suggestions from the three anonymous reviewers, making this review clear and comprehensive. We would like to thank Drs Youlian Pan (NRC), Alain Tchagang (NRC) and Raymond Ng (UBC) for providing valuable comments in the improvement of this article. We also want to acknowledge Ping Luo (UofS) for searching for potential non-cancer multi-omics data.

Funding

The National Research Council Canada (NRC) and the Natural Sciences and Engineering Research Council of Canada.

References

1. Zhou Z, Chawla N, Jin Y, *et al*. Big data opportunities and challenges: discussions from data analytics perspectives. *IEEE Comput Intell Mag* 2014;**9**(4):62–74.
2. The Cancer Genome Atlas Network. Comprehensive molecular portraits of human breast tumors. *Nature* 2012;**490**(7418):61–70.
3. The ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature* 2012;**489**:57–74.
4. Roadmap Epigenomics Consortium, Kundaje A, Meuleman W, *et al*. Integrative analysis of 111 reference human epigenomes. *Nature* 2015;**518**:317–30.
5. The GTEx Consortium. The genotype-tissue expression (GTEx) project. *Nature Genetics* 2013;**45**(6):580–5.
6. Wheeler D, Srinivasan M, Egholm M, *et al*. The complete genome of an individual by massively parallel DNA sequencing. *Nature* 2008;**452**:872–6.
7. Johnson D, Mortazavi A, Myers R, *et al*. Genome-wide mapping of in vivo protein-DNA interactions. *Science* 2007;**316**(5830):447–55.
8. Boyle A, Davis S, Shulha H, *et al*. High-resolution mapping and characterization of open chromatin across the genome. *Cell* 2008;**132**:311–22.
9. Lister R, Pelizzola M, Dowen R, *et al*. Human DNA methylomes at base resolution show widespread epigenomic differences. *Nature* 2009;**462**:315–22.
10. Nagalakshmi U, Wang Z, Waern K, *et al*. The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science* 2008;**320**(5881):1344–9.
11. Creighton C, Reid J, Gunaratne P. Expression profiling of microRNAs by deep sequencing. *Brief Bioinform* 2009;**10**(5):490–7.
12. Spurrier B, Ramalingam S, Nishizuka S. Reverse-phase protein lysate microarrays for cell signaling analysis. *Nat Protoc* 2008;**3**(11):1796–808.
13. Domon B, Aebersold R. Mass spectrometry and protein analysis. *Science* 2006;**321**(5771):212–7.
14. Dunn W, Erban A, Weber R, *et al*. Mass appeal: Metabolite identification in mass spectrometry-focused untargeted metabolomics. *Metabolomics* 2013;**9**:S44–66.
15. Xu R, Wunsch D. *Clustering*. New Jersey: Wiley-IEEE Press, 2008.
16. Ritchie M, Holzinger E, Li R, *et al*. Methods of integrating data to uncover genotype-phenotype interactions. *Nat Rev Genet* 2015;**15**:85–97.
17. Kristensen V, Lingjarde O, Russnes H, *et al*. Principles and methods of integrative genomic analyses in cancer. *Nat Rev Cancer* 2014;**14**:299–313.
18. Costello J, Heiser L, Georgii E, *et al*. A community effort to assess and improve drug sensitivity prediction algorithms. *Nat Biotechnol* 2014;**32**:1202–12.
19. Li Y, Chen C, Kaye A, *et al*. The identification of *cis*-regulatory elements: a review from a machine learning perspective. *Biosystems* 2015;**138**:6–17.
20. Nobel W, Support vector machine applications in computational biology. In: B Scholkopf, K Tsuda, JP Vert (eds), *Kernel Methods in Computational Biology*, Chap. 3. Cambridge, MA: MIT Press, 2004, 71–92.
21. Li Y, Ngom A. Data integration in machine learning. In: *IEEE International Conference on Bioinformatics and Biomedicine*, IEEE, IEEE Press, Piscataway, NJ, 2015, p. 1665–71.
22. Gligorijevic V, Przulj N. Methods for biological data integration: perspectives and challenges. *J R Soc Interface* 2015;**12**(112):20150571.
23. Ding J, Shi J, Wu F. SVM-RFE based feature selection for tandem mass spectrum quality assessment. *Int J Data Min Bioinform* 2011;**5**(1):73–88.
24. Cortes C, Vapnik V. Support vector networks. *Mach Learn* 1995;**20**:273–97.
25. Tibshirani R. Regression shrinkage and selection via the Lasso. *J R Stat Soc Ser B* 1996;**58**(1):267–88.
26. Zou H, Hastie T. Regularization and variable selection via the elastic net. *J R Stat Soc Ser B* 2005;**67**(2):301–20.
27. Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. *J R Stat Soc Ser B* 2006;**68**(1):49–67.
28. Friedman J, Hastie T, Tibshirani R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* 2008;**9**(3):432–41.
29. Friedman J, Hastie T, Tibshirani R. A note on the group lasso and a sparse group lasso. *arXiv* 2010:arXiv:1001.0736.
30. Jacob L, Obozinski G, Vert JP. Group Lasso with overlap and graph Lasso. In: *International Conference on Machine Learning*, ACM Press, New York, NY, 2009, p. 433–40.
31. Azencott CA, Grimm D, Sugiyama M, *et al*. Efficient network-guided multi-locus association mapping with graph cuts. *Bioinformatics* 2012;**29**(ISMB/ECCB2013):i171–9.
32. Zhao P, Yu B. On model selection consistency of lasso. *J Mach Learn Res* 2006;**7**:2541–63.
33. Zou H. The adaptive LASSO and its oracle property. *J Am Stat Assoc* 2006;**101**:1418–29.
34. Meinshausen U, Buhlmann P. Stability selection. *J R Stat Soc Ser B* 2010;**72**(4):417–73.
35. Wu C, Ma S. A selective review of robust variable selection with applications in bioinformatics. *Brief Bioinform* 2015;**16**(5):873–83.
36. Pique-Regi R, Degner J, Pai A, *et al*. Accurate inference of transcription factor binding from DNA sequence and chromatin accessibility data. *Genome Res* 2011;**21**:447–55.
37. Pearl J, *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufmann, 1988.
38. Chickering D, Learning Bayesian networks is NP-complete. In: D Frisher, HJ Lenz (eds.) *Learning from Data: AI and Statistics V, Lecture Notes in Statistics*, Chap. 12, Springer-Verlag New York, Inc, Secaucus, NJ, 1996, 121–30.
39. Elidan G, Nachman I, Friedman N. "Ideal Parent" structure learning for continuous variable Bayesian networks. *J Mach Learn Res* 2007;**8**:1799–833.
40. Davies S, Moore A. Mix-nets: Factored mixtures of Gaussians in Bayesian networks with mixed continuous and discrete variables. In: *Proceedings of The Sixteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc, San Francisco, CA, 2000, p. 168–75.
41. Cheng J, Greiner R. Comparing Bayesian network classifiers. In: *The Fifteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc, San Francisco, CA, 1999, p. 101–8.
42. Heckerman D, Geiger D, Chickering D. Learning Bayesian networks: the combination of knowledge and statistical data. *Mach Learn* 1995;**20**:197–243.

43. Langley P, Iba W, Thompson K. An analysis of Bayesian classifiers. In: *The Tenth National Conference on Artificial Intelligence*, AAAI Press, Menlo Park, CA, 1992, p. 223–8.
44. Friedman N, Geiger D, Goldszmith M. *Bayesian Network Classifiers*. *Machine Learning* 1997;**29**:103–30.
45. Chickering D, Meek C. Finding optimal Bayesian networks. In: *The Eighteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc., San Francisco, CA, 2002, p. 94–102.
46. Heckerman D, A tutorial on learning with Bayesian networks. In: M Jordan (ed.) *Learning in Graphical Models, Adaptive Computation and Machine Learning series*, Chap. 11. Cambridge, MA: MIT, 1998, 301–54.
47. Breiman L, Friedman J, Stone C, *et al*. *Classification and Regression Trees*. Chapman and Hall/CRC, Boca Raton, FL, 1984.
48. Friedman J, Tibshirani R, Hastie T, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd edn. Springer-Verlag New York, Inc., Secaucus, NJ, 2009.
49. Opitz D, Maclin R. Popular ensemble methods: an empirical study. *J Artif Intell Res* 1999;**11**:169–98.
50. Sewell M. Ensemble Learning, Technical report, Department of Computer Science, University College London, 2011.
51. Breiman L. Bagging predictors. *Machine Learning* 1996;**24**:(3):123–140.
52. Kearns M, Thoughts on hypothesis boosting 1988.
53. Breiman L. Arcing classifiers. *The Ann Stat* 1998;**26**(3):801–49.
54. Gunduz N, Fokoue E. Robust classification of high dimension low sample size data. *arXiv* 2015:arXiv:1501.00592.
55. Biau G, Devroye L, Lugosi G. Consistency of random forests and other averaging classifiers. *J Mach Learn Res* 2008;**9**:2015–33.
56. Scornet E, Biau G, Vert JP. Consistency of random forests. *Ann Stat* 2015;**43**(4):1716–41.
57. Polikar R. Ensemble based systems in decision making. *IEEE Circuits Syst Mag* 2006;**6**(3):21–45.
58. Polikar R. Bootstrap inspired techniques in computational intelligence: Ensemble of classifiers, incremental learning, data fusion and missing features. *IEEE Signal Proc Mag* 2007;**24**(4):59–72.
59. Wozniak M, Grana M, Corchado E. A survey of multiple classifier systems as hybrid systems. *Inf Fusion* 2014;**16**:3–17.
60. Chen C, Liaw A, Breiman L. Using Random Forest to Learn Imbalanced Data. Technical report, Department of Statistics, University of California, Berkeley, 2004.
61. Popovic D, Sifrim A, Davis J, *et al*. Problems with the nested granularity of feature domains in bioinformatics: the eXtasy case. *BMC Bioinformatics* 2015;**16**(Supp. 4):S2.
62. Pittman J, Huang E, Dressman H, *et al*. Integrated modeling of clinical and gene expression information for personalized prediction of disease outcomes. *Proc Natl Acad Sci* 2004;**101**(22):8431–6.
63. Drucker H, Burges C, Kaufman L, *et al*. Support vector regression machines. In: *Advances in Neural Information Processing Systems*. MIT Press, Cambridge, MA, 1996, 155–61.
64. von Luxburg U. A tutorial on spectral clustering. *Stat Comput* 2007;**17**(4):395–416.
65. Li Y, Ngom A. Sparse representation approaches for the classification of high-dimensional biological data. *BMC Syst Biol* 2013;**7**(Suppl 4):S6.
66. Gonen M, Alpaydin E. Multiple kernel learning algorithms. *J Mach Learn Research* 2011;**12**:2211–68.
67. Gonen M, Kaski S. Kernelized Bayesian matrix factorization. *IEEE Trans Pattern Anal Mach Intell* 2014;**36**:2047–60.
68. Wang J, Do HT, Woznica A, *et al*. Metric learning with multiple kernels. In: J Shawe-Taylor, R Zemel, P Bartlett, *et al*. (eds.) *Advances in Neural Information Processing Systems 24*. Curran Associates, Inc., Red Hook, NY, 2011, 1170–78.
69. Xing E, Jordan M, Russell S, *et al*. Distance metric learning with application to clustering with side-information. In: S Becker, S Thrun, K Obermayer (eds.) *Advances in Neural Information Processing Systems* 15. MIT Press, Cambridge, MA, 2003, 521–28.
70. Bellet A, Habrard A, Sebban M. A survey on metric learning for feature vectors and structured data. *arXiv* 2014. arXiv:1306.6709v4.
71. Kulis B. Metric learning: a survey. *Found Trends Mach Learn* 2012;**5**(4):287–364.
72. Li Y, Caron R, Ngom A. A decomposition method for large-scale sparse coding in representation learning. In: *International Joint Conference on Neural Networks (IJCNN/WCCI)*, IEEE, IEEE Press, Piscataway, NJ, 2014, p. 3732–38.
73. Kim SJ, Magnani A, Boyd S. Optimal kernel selection in kernel Fisher discriminant analysis. In: *International Conference on Machine Learning*, ACM Press, New York, NY, 2006, p. 465–72.
74. Rakotomamonjy A, Bach F, Canu S, *et al*. SimpleMKL. *J Mach Learn Res* 2008;**9**:2491–521.
75. Zhang Y, Li A, Peng C, *et al*. Improve glioblastoma multiforme prognosis prediction by using feature selection and multiple kernel learning. *IEEE/ACM Trans Comput Biol Bioinform* 2016;DOI:10.1109/TCBB.2016.2551745.
76. Wang B, Mezlini A, Demir F, *et al*. Similarity network fusion for aggregating data types on a genomic scale. *Nat Methods* 2014;**11**(3):333–7.
77. Shen R, Olshen A, Ladanyi M. Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics* 2009;**25**(22):2906–12.
78. Marbach D, Costello J, Kuffner R, *et al*. Wisdom of crowds for robust gene network inference. *Nat Methods* 2012;**9**(8):796–804.
79. List M, Alcaraz N, Dissing-Hansen M, *et al*. KeyPathwayMinerWeb: online multi-omics network enrichment. *Nucleic Acids Res* 2016;**44**:W98–104. (Web Server):
80. Zitnik M, Zupan B. Gene network inference by fusing data from diverse distributions. *Bioinformatics* 2015;**31**:i230–9.
81. Chen B, Wang J, Li M, *et al*. Identifying disease genes by integrating multiple data sources. *BMC Med Genomics* 2014;**7**(Supp 2):S2.
82. Chen B, Wu F. Identifying protein complexes based on multiple topological structures in PPI networks. *IEEE Trans Nanobiosci* 2013;**12**(3):165–72.
83. Chen B, Wang J, Shang X, *et al*. Identifying individual-cancer-related genes by re-balancing the training samples. *IEEE Trans Nanobiosci* 2016;DOI:10.1109/TNB.2016.2553119.
84. Chen B, Li M, Wang J, *et al*. A fast and high performance algorithm for identifying human disease genes. *BMC Med Genomics* 2015;**8**(Suppl 3):S2.
85. Chen B, Li M, Wang J, *et al*. Disease gene identification by using graph kernels and Markov random fields. *Sci China Life Sci* 2014;**57**(11):1052–63.
86. Ammad-ud-din M, Georgii E, Gonen M, *et al*. Integrative and personalized QSAR analysis in cancer by kernelized Bayesian matrix factorization. *J Chem Inf Model* 2014;**54**(8):2347–59.

87. Lan W, Wang J, Li M, *et al*. Predicting microRNA-disease associations based on microRNA and disease similarity. *IEEE/ACM Trans Comput Biol Bioinform* 2016. DOI:10.1109/TCBB.2016.2586190.
88. Li Y, Patra J. Genome-wide inferring gene-phenotype relationship by walking on the heterogeneous network. *Bioinformatics* 2010;**26**(9):1219–24.
89. Chen X, Liu MX, Yan GY. Drug-target interaction prediction by random walk on the heterogeneous network. *Mol Biosyst* 2012;**8**(7):1970–8.
90. Xuan P, Han K, Guo Y, *et al*. Prediction of potential disease-associated microRNAs based on random walk. *Bioinformatics* 2015;**31**(11):1805–15.
91. Liu Y, Zeng X, He Z, *et al*. Inferring microRNA-disease associations by random walk on a heterogeneous network with multiple data sources. *IEEE/ACM Trans Comput Biol Bioinform* 2016;DOI:10.1109/TCBB.2016.2550432.
92. Huang YF, Yeh HY, Soo VW. Inferring drug-disease associations from integration of chemical, genomic and phenotype data using network propagation. *BMC Med Genomics* 2013;**6(Supp 3)**:S4.
93. Zitnik M, Zupan B. Data fusion by matrix factorization. *IEEE Trans Pattern Anal Mach Intell* 2015;**37**(1):41–53.
94. Zitnik M, Zupan B. Jumping across biomedical contexts using compressive data fusion. *Bioinformatics* 2016;**32**:i90–100.
95. Jolliffe I, *Principal Component Analysis*. Secaucus, NJ: Springer-Verlag New York. Inc., 2002.
96. Wall M, Rechtsteiner A, Rocha L, Singular value decomposition and principal component analysis. In: D Berrar, W Dubitzky, M Granzow (eds.) *A Practical Approach to Microarray Data Analysis*. Norwell, MA: Kluwer, 2003, 91–109.
97. Lawley D. The estimation of factor loadings by the method of maximum likelihood. *Proc R Soc Edinb* 1940;**60**:64–82.
98. West M. Bayesian factor regression models in the "large p, small n" paradigm. *Bayesian Stat* 2003;**7**:723–32.
99. Lee D, Seung S. Learning the parts of objects by non-negative matrix factorization. *Nature* 1999;**401**:788–91.
100. Fertig E, Ding J, Favorov A, *et al*. CoGAPS: an R/C++ package to identify patterns and biological process activity in transcriptomic data. *Bioinformatics* 2010;**26**(21):2792–3.
101. Li Y, Ngom A. The non-negative matrix factorization toolbox for biological data mining. *BMC Source Code Biol Med* 2013;**8**(1):10.
102. Li L, Wu L, Zhang H, *et al*. A fast multiplicative update algorithm for nonnegative matrix factorization and its convergence. *IEEE Trans Neural Netw Learn Syst* 2014;**25**(10):1855–63.
103. Kolda T, Bader B. Tensor decompositions and applications. *SIAM Rev* 2009;**51**(3):455–500.
104. Li Y, Ngom A. Non-negative matrix and tensor factorization based classification of clinical microarray gene expression data. In: *IEEE International Conference on Bioinformatics and Biomedicine*. IEEE, IEEE Press, Piscataway, NJ, 2010, p. 438–43.
105. Hinton G, Salakhutdinov R. Reducing the dimensionality of data with neural networks. *Science* 2006;**313**:504–7.
106. Hinton G. A practical guide to training restricted Boltzmann machines. Technical report., Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, 2010.
107. Li Y, Ngom A. Versatile sparse matrix factorization: theory and applications. *Neurocomputing* 2014;**145**:23–9.
108. Virtanen S, Klami A, Khan S, *et al*. Bayesian group factor analysis. In: *Artificial Intelligence and Statistics Conference*, La Palma, Canary Islands, 2012, p. 1269–77.
109. Klami A, Virtanen S, Leppaaho E, *et al*. Group factor analysis. *IEEE Trans Neural Netw Learn Syst* 2015;**26**(9):2136–47.
110. Klami A, Virtanen S, Kaski S. Bayesian cononical correlation analysis. *J Mach Learn Res* 2013;**14**:965–1003.
111. Bunte K, Leppaaho E, Saarinen I, *et al*. Sparse group factor analysis for biclustering of multiple data sources. *Bioinformatics* 2016;**32**(16):2457–63.
112. Hochreiter S, Bodenhofer U, Heusel M, *et al*. FABIA: Factor analysis for bicluster acquisition. *Bioinformatics* 2010;**26**(12):1520–7.
113. Liu J, Wang C, Gao J, *et al*. Multi-view clustering via joint non-negative matrix factorization. In: *SIAM International Conference on Data Mining*, Austin, USA, 2013, p. 252–60.
114. Hidru D, Goldenberg A. EquiNMF: Graph regularized multi-view nonnegative matrix factorization. In: Workshop on Machine Learning in Computational Biology co-located with NIPS, Montreal, Canada, 2014, p. 1–9.
115. Zhang S, Liu C, Li W, *et al*. Discovery of multi-dimensional modules by integrative analysis of cancer genomic data. *Nucleic Acids Res* 2012;**40**(19):9379–91.
116. Yang Z, Michailidis G. A non-negative matrix factorization methods for detecting modules in heterogeneous omics multi-modal data. *Bioinformatics* 2016;**32**(1):1–8.
117. Hellton K, Thoresen M. Integrative clustering of high-dimensional data with joint and individual clusters. *Biostatistics* 2016;**17**(3):537–48.
118. Lock E, Hoadley K, JS, *et al*. Joint and individual variation explained (JIVE) for integrated analysis of multiple data types. *Ann Appl Stat* 2013;**7**:1:523–42.
119. Wold H, Nonlinear estimation by iterative least square procedures. In: F David (ed.) *Research Papers in Statistics*. New York: John Wiley and Sons Inc., 1966, 411–44.
120. Biancolillo A, Liland K, Mage I, *et al*. Variable selection in multi-block regression. *Chemometr Intell Lab Syst* 2016;**165**:89–101.
121. Trygg J, Wold S. Orthogonal projections to latent structures (O-PLS). *J Chemometr* 2002;**16**:119–28.
122. Trygg J. O2-PLS for qualitative and quantitative analysis in multivariate calibration. *J Chemometr* 2002;**16**:283–93.
123. Trygg J, Wold S. O2-PLS, a two-block (X-Y) latent variable regression (LVR) method with an integral OSC filter. *J Chemometr* 2003;**17**:53–64.
124. Bylesjo M, Eriksson D, Kusano M, *et al*. Data integration in plant biology: the O2PLS method for combined modeling of transcript and metabolite data. *Plant J* 2007;**52**:1181–91.
125. Lofstedt T, Trygg J. OnPLS - A novel multiblock method for the modelling of predictive and orthogonal variation. *J Chemometr* 2011;**25**:441–55.
126. Lofstedt T, Hanafi M, Mazerolles G, *et al*. OnPLS path modelling. *Chemometr Intell Lab Syst* 2012;**118**:139–49.
127. Srivastava V, Obudulu O, Bygdell J, *et al*. OnPLS integration of transcriptomic, proteomic and metabolomic data shows multi-level oxidative stress responses in the cambium of transgenic hipI- superoxide dismutase Populus plants. *BMC Genomics* 2013;**14**:893.
128. Rantalainen M, Bylesjo M, Cloarec O, *et al*. Kernel-based orthogonal projections to latent structures (K-OPLS). *J Chemometr* 2007;**21**:376–85.
129. Omberg L, Golub G, Alter O. A tensor higher-order singular value decomposition for integrative analysis of DNA microarray data from different studies. *Proc Natl Acad Sci USA* 2007;**104**(47):18371–6.
130. Cichocki A, Zdunek R, Phan A, *et al*. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way*

Data Analysis and Blind Source Separation. West Sussex: John Wiley & Sons, 2009.

131. Smilde A, Mage I, Naes T, *et al*. Common and distinct components in data fusion. *ArXiv* 2016. arXiv:1607.02328.
132. Wang Z, Yuan W, Montana G. Sparse multi-view matrix factorization: a multivariate approach to multiple tissue comparisons. *Bioinformatics* 2015;**31**(19):3163–71.
133. Hubert M, Branden K. Robust methods for partial least squares regression. *J Cheometr* 2003;**17**:537–49.
134. Huang J, Nie F, Huang H, *et al*. Robust manifold nonnegative matrix factorization. *ACM Trans Knowl Discov Data* 2014;**8**(3):Article No. 11.
135. Huang H, Ding C. Robust tensor factorization using R1 norm. In: *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Press, Piscataway, NJ, 2008, p. 1–8.
136. Luttinen J, Ilin A. Transformations in variational Bayesian factor analysis to speed up learning. *Neurocomputing* 2010;**73**:1093–102.
137. Cemgil A. Bayesian inference for nonnegative matrix factorization models. *Computat Intell Neurosci* 2009;**2009**:785152.
138. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015;**521**:436–44.
139. Hinton G, Osindero S, Teh Y. A fast learning algorithm for deep belief nets. *Neural Comput* 2006;**18**:1527–54.
140. Salakhutdinov R, Hinton G. Deep Boltzmann machine. In: *International Conference on Artificial Intelligence and Statistics*, Volume 5 of JMLR: W&CP, Microtome Publishing, Brookline, MA, 2009, p. 448–455.
141. LeCun Y, Bengio Y, Convolutional networks for images, speech, and time series. In: M Arbib (ed.) *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA: MIT Press, 1995, 255–8.
142. Graves A, Jaitly N. Towards end-to-end speech recognition with recurrent neural networks. In: *International Conference on Machine Learning (ICML)*, JMLR: W&CP volume 32, Curran Associates, Inc., Red Hook, NY, 2014, p. 1764–72.
143. Li Y, Chen C, Wasserman W. Deep feature selection: theory and application to identify enhancers and promoters. *J Comput Biol* 2016;**23**(5):322–36.
144. Srivastava N, Salakhutdinov R. Multimodal learning with deep Boltzmann machines. *J Mach Learn Res* 2014;**15**:2949–80.
145. Liang M, Li Z, Chen T, *et al*. Integrative data analysis of multi-platform cancer data with a multimodal deep learning approach. *IEEE/ACM Trans Comput Biol Bioinform* 2015;**12**(4):928–37.
146. Bengio IGY, Courville A. *Deep Learning*, 2016. Book in preparation for MIT Press, Cambridge, MA. http://www.deeplearningbook.org
147. Hornik K. Approximation capabilities of multilayer feedforward networks. *Neural Netw* 1991;**4**(2):251–7.
148. Farago A, Lugosi G. Strong universal consistency of neural network classifiers. *IEEE Trans Inf Theory* 1993;**39**(4):1146–51.
149. Liano K. Robust error measure for supervised neural network learning with outliers. *IEEE Trans Neural Netw* 1996;**7**(1):246–50.
150. Seltzer M, Yu D, Wang Y. An investigation of deep neural networks for noise robust speech recognition. In: International Conference on Acoustics, Speech, and Signal Processing, 2013, p. 7398–492.
151. de-la Calle-Silos F, Gallardo-Antoln A, Pelaez-Moreno C. Deep maxout networks applied to noise-robust speech recognition, chap. In: *Advances in Speech and Language Technologies for Iberian Languages*. Springer-Verlag, Berlin Heidelberg, 2014, 109–18.
152. Srivastava N, Hinton G, Krizhevsky A, *et al*. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 2014;**15**:1929–58.
153. Wu Y, Zhang S, Benjio Y, *et al*. On multiplicative integration with recurrent neural networks. *ArXiv* 2016. arXiv:1606.06630.
154. Woods A, Wormwood K, Wetie A, *et al*. Autism spectrum disorder: An omics perspective. *Proteomics* 2015;**9**(1-2):159–68.
155. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. *J Stat Softw* 2010;**33**:1–22.
156. Pedregosa F, Varoquaux G, Gramfort A, *et al*. Scikit-learn: machine learning in Python. *J Mach Learn Res* 2011;**12**:2825–30.
157. Meier L, van de Geer S, Buhlmann P. The group lasso for logistic regression. *J R Stat SocSer B* 2008;**70**(1):53–71.
158. Mairal J, Bach F, Ponce J. Sparse modeling for image and vision processing. *Found Trends Comput Graph Vision* 2014;**8**(2-3):85–283.
159. Scutari M. Learning Bayesian networks with the bnlearn R package. *J Stat Softw* 2010;**35**(3):1–22.
160. Liaw A, Wiener M. Classification and regression by randomForest. *R News* 2002;**2**(3):18–22.
161. Strazar M, Curk T. Learning the kernel matrix via predictive low-rank approximation. *arXiv* 2016. arXiv:1601.04366.
162. Xu X, Tsang I, Xu D. Soft margin multiple kernel learning. *IEEE Trans Neural Netw Learn Syst* 2013;**24**(5):749–61.
163. Varma M, Babu B. More generality in efficient multiple kernel learning. In: International Conference on Machine Learning, 2009, p. 1065–72.
164. Mevik BH, Wehrens R. The pls package: Principal component and partial least squares regression in R. *J Stat Softw* 2007;**18**(2):1–23.
165. Chung D, Keles S. Sparse partial least squares classification for high dimensional data. *Stat Appl Genet Mol Bioinform* 2010;**9**(1):17.
166. Bouhaddani S, Houwing-Duistermaat J, Jongbloed G, *et al*. Evaluation of O2PLS in omics data integration. *BMC Bioinformatics* 2016;**17**(Suppl 2):S11.
167. Bylesjo M, Rantalainen M, Nicholson J, *et al*. K-OPLS package: Kernel-based orthogonal projections to latent structures for prediction and interpretation in feature space. *BMC Bioinformatics* 2008;**9**:106.
168. Bader BW, Kolda TG, *et al*. Matlab Tensor Toolbox Version 2.6. Available online 2015.
169. Andersson C, Bro R. The N-way toolbox for MATLAB. *Chemometr Intell Lab Syst* 2000;**52**(1):1–4.
170. Rasmussen M, Bro RA. tutorial on the LASSO approach to sparse modelling. *Chemometr Intell Lab Syst* 2012;**119**:21–31.
171. Acar E, Papalexakis E, Gurdeniz G, *et al*. Structure-revealing data fusion. *BMC Bioinformatics* 2014;**15**:239.