Dipartimento di Informatica, Bioingegneria,
Robotica ed Ingegneria dei Sistemi

Challenges in biomedical data science: data-driven solutions to clinical questions

by

Samuele Fiorini

Theses Series **DIBRIS-TH-2017-XX**

DIBRIS, Università di Genova

Via Opera Pia, 13 16145 Genova, Italy http://www.dibris.unige.it/

Università degli Studi di Genova

Dipartimento di Informatica, Bioingegneria,
Robotica ed Ingegneria dei Sistemi

Ph.D. Thesis in Computer Science and Systems Engineering
Computer Science Curriculum

Challenges in biomedical data science: data-driven solutions to clinical questions

by

Samuele Fiorini

October, 2017

Dottorato di Ricerca in Informatica ed Ingegneria dei Sistemi
Indirizzo Informatica
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università degli Studi di Genova

DIBRIS, Univ. di Genova
Via Opera Pia, 13
I-16145 Genova, Italy
`http://www.dibris.unige.it/`

Ph.D. Thesis in Computer Science and Systems Engineering
Computer Science Curriculum
(S.S.D. INF/01)

Submitted by Samuele Fiorini
DIBRIS, Univ. di Genova
. . . .

Date of submission: October 5, 2017

Title: Machine Learning 4 healthcare.

Advisor: Annalisa Barla
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università di Genova
. . .

Ext. Reviewers:
Lo Scopriremo
Lo Scopriremo
Lo Scopriremo

Abstract

Abstract

Contents

1 Introduction

Understanding the underlying mechanisms of biological systems can be a challenging task. Different domains can be involved and their interactions can be unknown.

Nowadays, most of the life science research relies upon the extraction of meaningful information from heterogeneous sources of biological data. Thanks to the remarkable technological progresses of the last decades, the dimensions of such data collections is increasing everyday.

This PhD thesis is divided in two parts. Part I presents a thorough description of the multidisciplinary prerequisites that are relevant for the comprehension of Part II, which, in turn, presents the original contributions of my work.

Part I is organized as follows: Chapter 3 introduces the concept of *data science* (Section 3.1) and its declination toward life science studies. It also describes the major challenges of the field (Section 3.2) along with several examples of the most common clinical/biological questions and their translation to data analysis tasks (Section 3.3). Chapter 4 summarizes basic notation and definitions adopted throughout the thesis (Section 2) and presents an overview of the statistical and technological tools that are mostly relevant for this work. In particular, this chapter defines the concept of *machine learning* from a general perspective and provides rigorous description of a selection of supervised and unsupervised learning strategies (Section 4.1). At the end of this chapter, hints on the computational requirements and implementation strategies are also presented (Section 4.2).

Part II describes the main contributions of my PhD work which consisted in the process of translating into data analysis tasks a number of biological questions coming from real-world clinical environments. For each task, this second part shows how the previously introduced tools can be exploited in order to develop statistically sound models that are capable of providing insightful answers to different clinical questions. This part is organized as follows: Chapter 5 introduces ADENINE, an open-source Python framework for large-scale data exploration I developed during my PhD. **{Chapter 6 describes a work I developed in collaboration with Gaslini hospital on biological age estimation from blood samples}**. Chapter 7 describes the development of a temporal model that aims at following the evolution of multiple sclerosis patients exploiting the use of patient-friendly and inexpensive measures such as patient centered outcomes. Chapter 8 describes a machine learning time-series forecasting approach for glucose sensor data collected by type I and type II diabetic patients.

2 Basic notation and definitions

In this thesis, the data are described as input-output pairs, $X \in \mathbb{R}^{n \times d}$ and $Y \in \mathbb{R}^{n \times k}$, respectively. The i-th row of X is a d-dimensional data point $\boldsymbol{x}_i$ belonging to the input space $\mathcal{X} \subseteq \mathbb{R}^d$. The corresponding outputs $\boldsymbol{y}_i$ belong to the output space $\mathcal{Y}$.

The nature of the output space defines the problem as *binary classification* if $\mathcal{Y} = \{a, b\}$ (with $a \neq b$), *multiclass classification* if $\mathcal{Y} = \{\alpha, \beta, \dots, \omega\}$ (with $\alpha \neq \beta \neq \dots \neq \omega$)), *regression* if $\mathcal{Y} \subseteq \mathbb{R}$ and *vector-valued regression* if $\mathcal{Y} \subseteq \mathbb{R}^k$. For binary classification problems common choices for the label encoding are $a = 1, b = -1$ or $a = 0, b = 1$. For multiclass classification problems classes are usually encoded as natural numbers, i.e. $\alpha, \beta, \dots, \omega \in \mathbb{N}$.

Predictive models are functions $f : \mathcal{X} \to \mathcal{Y}$. The number of relevant variables is d^*. In feature selection tasks, the number of selected features is $\tilde{d}$.

A kernel function acting on the elements of the input space is defined as $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$, where $\phi(\boldsymbol{x})$ is a *feature map* from $\mathbb{R}^d \to \mathbb{R}^{d'}$. Feature learning algorithms project the data into a p-dimensional space.

Part I

3 Background

3.1 What is data science and why should we care?

{

- **Data engineering**
- **Data exploration**
- **Machine learning and data understanding**
- **Data visualization**

}

- cross-disciplinary field
- Drew Conway's Data Science Venn Diagram, first published on his blog in September 2010
- data-intensive applications (maybe)

3.2 Challenges in biomedical data science

The process of modeling complex systems often implies collecting large amount of data in the field of life science, where large, multivariate and noisy measurements are typically acquired with the aim of describing multifactorial diseases.

In the era of personalized medicine, biospecimen collection and biological data management is still a challenging and expensive task [Toga and Dinov, 2015]. Only few large-scale research enterprises, such as ENCODE [Consortium et al., 2004], ADNI [Jack et al., 2008], MOPED [Kolker et al., 2012] or TCGA [2] , have sufficient financial and human resources to manage, share and

[1] http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

[2] https://cancergenome.nih.gov

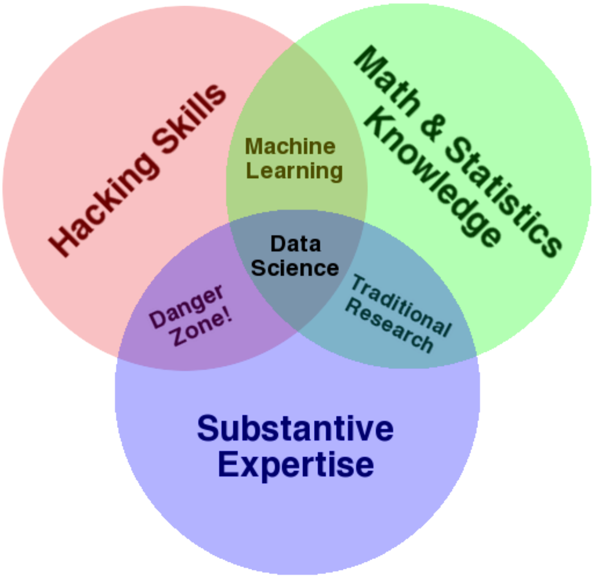


Figure 3.1: Drew Conway's Data Science Venn Diagram[1].

distribute access of heterogeneous types of biological data. To date, many biomedical studies still rely on a small number of collected samples [McNeish and Stapleton, 2016; Button et al., 2013; Yu et al., 2013]. This effect is even worse in case of rare diseases [Garg et al., 2016] or in high-throughput molecular data (e.g. genomics and proteomics) where the dimensionality of the problem can be in the order of hundreds of thousands. The setting where the number of measured variables heavily outnumbers the amount of collected samples is usually referred to as *large p small n* scenario, or simply $n \ll p$. In this case, the main goal of the learning step is often to identify a meaningful subset of relevant variables that are the most representative of the observed phenomenon. In machine learning, this is known as variable/feature selection and several techniques addressing this task were presented so far [Guyon et al., 2002]. Variable selection not only increases the prediction power of the learning machine, but it also promotes model interpretability, that is crucial in biology [Altmann et al., 2010]. Regardless [Okser et al., 2014] of the learning machine, regularization can be introduced in several ways and it is of fundamental use in order to achieve the following desired properties:

- identify models with good generalization properties, even with a limited amount of collected samples;
- achieve solutions that are robust to noise;
- learn the data structure when unknown;
- exploit prior knowledge on the data structure;

- promote interpretability performing variable selection;
- reduce the feasible set in order to help solving inverse problems.

In this paper we illustrate how regularization impacts in finding robust and meaningful models and we clarify how to choose the most suitable regularization scheme according to the biological context. The remainder of the paper is organized as follows: ....

3.3 From clinical questions to data analysis

In applied life science, the biological question at hand usually drives the data collection and therefore the statistical challenge to be solved. In order to achieve meaningful results, thorough data analysis protocol must be followed (see Section 4.1.3). In this section, my goal is to illustrate some of the most recurrent biological questions and how they can be translated into machine learning tasks.

3.3.1 How to predict phenotypes from observed data?

Starting from a collection of input measures that are likely to be related with some known target phenotype, the final goal here is to learn a model that represents the relationship between input and output. Several researches fall in this class, for instance in molecular (e.g. lab tests, gene expression, proteomics, sequencing) [Angermueller et al., 2016; Okser et al., 2014; Abraham et al., 2013] or radiomics/imaging studies (e.g. MRI, PET/SPECT, microscopy) [Min et al., 2016; Helmstaedter et al., 2013]. Biological questions of this class are usually tackled by *supervised learning* models. In particular, when the observed clinical outcome is expressed as a one-dimensional continuous value, as in survival analysis, a *single-output regression* problem is posed. Moreover, if the outcome is vector-valued, as in the case of multiple genetic trait prediction [He et al., 2016], the problem can be cast in a *multiple-output regression* framework [Argyriou et al., 2008; Baldassarre et al., 2012]. Biological studies involving categorical outcomes translate into *classification* problems. In particular, if the clinical outcome assumes only two values, as in the *case-control* scenario, the classification problem is said to be *binary*, whilst, if multiple classes are observed, the classification task becomes *multi-class*.

3.3.2 Which variables are the most significant?

In the above case, a complementary question revolves around the interpretability of the predictive model. In particular, if dealing with high-throughput data, the main goal is to identify a relevant subset of meaningful variables for the observed phenomenon. This problem can be cast into a variable/feature selection problem [Guyon et al., 2002].

A machine learning model is said to be *sparse* when it only contains a small number of non-zero parameters, with respect to the number of features that can be measured on the objects this model represents [Hastie et al., 2015; Meier et al., 2008]. This is closely related to feature selection: if these parameters are weights on the features of the model, then only the features with non-zero weights actually enter the model and can be considered *selected*.

3.3.3 How to stratify the data?

Collecting measures from several samples, the final goal here is to divide them in homogeneous groups, according to some *similarity* criterion. In machine learning, this is usually referred to as *clustering* [Hastie et al., 2009].

3.3.4 How to represent the samples?

In order to formulate a model of some natural phenomenon, it is necessary to design and follow a suitable data collection protocol. A natural question that may arise here is whether the raw collected measures are intrinsically representative of the target phenomenon or if some transformation must be applied in order to achieve a data representation that can be successfully exploited by a learning machine. For instance, it may be plausible to assume that the data lie in a low-dimensional embedding or that they can be better represented by a richer polynomial or Gaussian expansion. A common solution, in this case, is to take advantage of *feature engineering* techniques to obtain hand crafted features. However, this process can be very time-consuming and it may require the help of domain experts. The process of automatically identify suitable representations from the data itself is usually referred to as *(un)supervised feature learning* [Angermueller et al., 2016; Mamoshina et al., 2016].

3.3.5 Are there recurring patterns in the data?

Analyzing data coming from complex domains, one may be interested in understanding whether complex observations can be represented by some combination of simpler events. In machine learning this typically translates into *adaptive sparse coding* or *dictionary learning* problems [Masecchia et al., 2015; Alexandrov et al., 2013].

3.3.6 How to deal with missing values?

Applied life science studies must often deal with the issue of missing data. For instance, peaks can be missed in mass-spectrometry [Jung et al., 2014] or gene expression levels can be impossible to measure due to insufficient array resolution or image corruption [Stekhoven and Bühlmann, 2011; Troyanskaya et al., 2001]. Common strategies, such as discarding the samples with missing entries, or replacing the holes with the mean, median or most represented value, fall short when the missing value rate is high or the number of collected samples is relatively small. In machine learning this task usually translates into a *matrix completion* problem [Candès and Recht, 2009].

4 State of the art

This chapter defines the concept of machine learning and presents a comprehensive overview of the most relevant algorithms and models.

4.1 Machine learning

The term *Machine Learning* (ML) first appeared in the late 50's in the field of computer science and now it is becoming a buzzword used in several contexts spanning from particle physics and astronomy to medicine and social sciences [Service, 2017]. With a simple search on Google Trends[1] it is possible to roughly quantify the pervasiveness of this term on the Internet in the last few years. From Figure 4.1 we can see that the interest toward both the terms *machine learning* and *data science* are growing, with the first consistently superior to the second.

A possible explanation to this phenomenon can be found in a recent article published on Science [Appenzeller, 2017] in which the authors describe a new *scientific revolution* lead by an explosion in the modern data collection abilities. Such massive amounts of data have long overwhelmed human analysis and insights potential and this makes ML a key element for scientists trying to make sense of large-scale observations.

But, what is *machine learning*? And how does it differ from statistics?

A unique answer to this question may not be easy to provide. In fact, ML can be defined in different ways and from several standpoints. Let's see three remarkable examples.

1. Kevin P. Murphy in its *Machine Learning - A Probabilistic Perspective* [Murphy, 2012] defines machine learning as follows.

    > "[...] *a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty* [...]"

2. Trevor Hastie, a well-known applied statistician, in a famous seminar[2], held in October 2015 at the Stanford University, gave the following three definitions.

[1] https://trends.google.com

[2] part of Data Science @ Stanford Seminar series (source: https://goo.gl/UFgqxU).

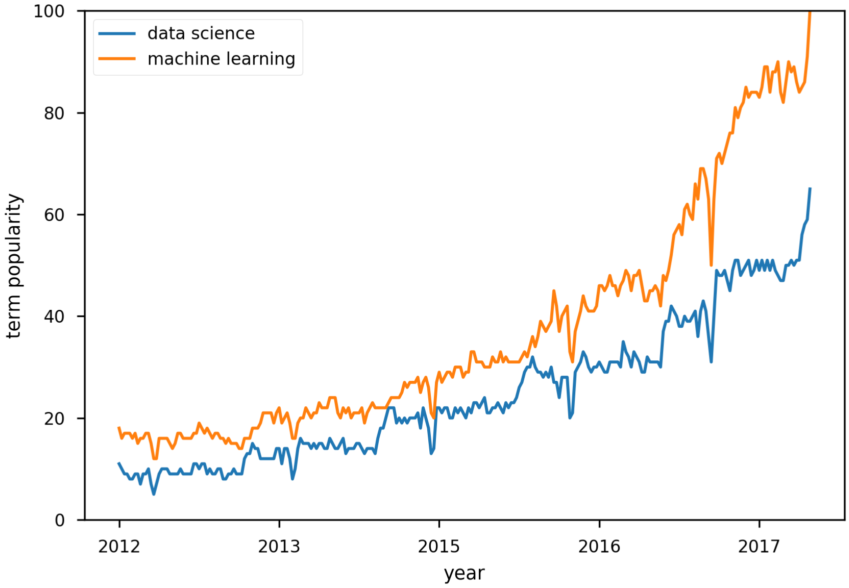


Figure 4.1: The Internet popularity over the past five years of two terms: *data science* and *machine learning*. The vertical axis represents a normalized measure of the number of Google search query of an input term. The trend is normalized with respect to its maximum (source: Google Trends).

> **Machine Learning** *constructs algorithms that can learn from data.*
>
> **Statistical Learning** *is a branch of applied statistics that emerged in response to machine learning, emphasizing statistical models and assessment of uncertainty.*
>
> **Data Science** *is the extraction of knowledge from data, using ideas from mathematics, statistics, machine learning, computer science, engineering...*

3. Carl E. Rasmussen in the preface of its *Gaussian Processes for Machine Learning* [Rasmussen and Williams, 2006] introduces the difference between statistics and ML as follows.

> "*in statistics a prime focus is often in understanding the data and relationships in terms of models giving approximate summaries such as linear relations or independencies. In contrast, the goals in machine learning are primarily to make predictions as accurately as possible and to understand the behaviour of learning algorithms*"

It looks like each author, according to his background, expertise and experiences, provides a slightly different definition of ML. Trying to summarize these three standpoints, we can say that

ML is an interdisciplinary field that borrows the concept of data-driven model from statistics in order to devise algorithms that can exploit hidden patterns in current data and make accurate predictions on future data.

As of today ML is the workhorse of data science.

4.1.1 Supervised learning

Humans are remarkably good at *learning by examples*. When a kid is taught what a pencil looks like, he will be capable of understanding the concept of pencil from a limited number of guided observations. Similarly, when future radiologists are trained to distinguish between healthy tissues from tumors in MRI scans, they will be provided with several annotated biomedical images from which they will be able to generalize. This learning paradigm is characterized by the presence of two key objects: *data* and *labels*. In the last example, we will consider the MRI scans as data, and their annotations (e.g. tumor vs healthy tissue) as labels.

Supervised learning is the branch of ML, in which predictive models are trained on labeled data. In the ML jargon, and in this thesis, one usually refers to data as collections of *samples* described by an arbitrarily large number of *predictors* (or *features*) that are used as *input* in the training process having labels as its *otput*.

In general, input samples x throughout this thesis are represented as d-dimensional vectors in an input space $\mathcal{X}$, where $\mathcal{X} \subseteq \mathbb{R}^d$, while the labels $y \in \mathcal{Y}$. The nature of the output space $\mathcal{Y}$ defines the learning task as *binary classification* if $\mathcal{Y} = \{-1, +1\}$, *multiclass classification* if $\mathcal{Y} = \{1, 2, \dots, k\}$, *regression* if $\mathcal{Y} \subseteq \mathbb{R}$ and *vector-valued regression* if $\mathcal{Y} \subseteq \mathbb{R}^k$. Each one of these learning problems will be faced in the second part of the thesis and for each problem a possible data-driven solution will be proposed. The remainder of this section describes the methods that are most relevant with the adopted models and pipelines.

4.1.1.1 Regularization methods

The process of identifying a model from a real-world data collection can be very hard. Many unwanted and concurrent factors may be misleading and the result may have poor predictive power. For instance, the acquisition devices may introduce random fluctuations in the measures or the amount of collected samples n may be small with respect to the number of observed variables d which, in turn, may not even be representative of the target phenomenon. From a modeling standpoint, every combination of the factors above can be seen as *noise* affecting the data. Precautions in the model formulation process must be taken in order to achieve solutions that are *robust* to the noise effect.

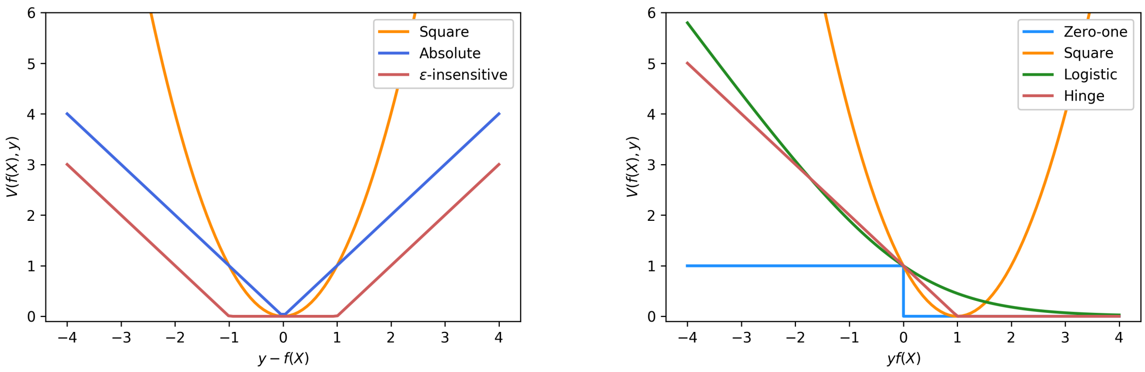
In the field of ML, a common strategy to build predictive models out of noisy data is called *regularization*. As the biomedical world is the the main area of interest of this Thesis (see Section 3.2), on each learning algorithm described, particular emphasis will be put on the relevant regularization strategies.

In its broader definition regularization is the process of introducing additional information in order to solve a possibly ill-posed problem [Tikhonov, 1963; Evgeniou et al., 2000]. The expected result is a function that fits the training data while having good generalization properties, i.e. accurate predictions on previously *unseen* test data [Hastie et al., 2009]. In ML, a model that fits well the training data but performs poorly on new samples is said to be *overfitting* the training set.

Given a set of input-output pairs $\{X, Y\}$, the main objective of *supervised learning* is to find a function of the inputs $f(X)$ that approximates the outputs Y. This translates into the minimization problem defined in Equation (4.1).

$$\min_f \frac{1}{n} \sum_{i=1}^{n} V(f(\boldsymbol{x}_i), y_i) + \lambda R(f) \tag{4.1}$$

The loss function $V(\cdot, \cdot)$ can be seen as a measure of *adherence* to the available training data. Several loss function for regression and classification problems were proposed; in Table 4.1 we define the most commonly adopted in biomedical studies and their visual representation is presented in Figure 4.2. The regularization penalty $R(\cdot)$ imposes stability on the expected function exploiting the available prior knowledge on the problem [Tikhonov, 1963].



(a)

(b)

Figure 4.2: An overview on the most common loss functions for regression (a) and classification (b) problems plotted against the corresponding prediction error.

Table 4.1: Definition of the loss functions for regression (top) and classification (bottom) problems represented in Figure 4.2.

Loss function	$V(f(\boldsymbol{x}), y)$	Learning problem
Square	$(y - f(\boldsymbol{x}))^2$	regression
Absolute	$\lvert y - f(\boldsymbol{x}) \rvert$	regression
ϵ-insensitive	$\begin{cases} 0 & \text{if } \lvert y - f(\boldsymbol{x}) \rvert < \epsilon \\ \lvert y - f(\boldsymbol{x}) \rvert - \epsilon & \text{otherwise} \end{cases}$	regression
Zero-one	$\begin{cases} 0 & \text{if } y = f(\boldsymbol{x}) \\ 1 & \text{otherwise} \end{cases}$	classification
Square	$(1 - yf(\boldsymbol{x}))^2$	classification
Logistic	$\log(1 + e^{-yf(\boldsymbol{x})})$	classification
Hinge	$\lvert 1 - yf(\boldsymbol{x}) \rvert_+$	classification

4.1.1.2 Ensemble methods

4.1.1.3 Deep learning

4.1.2 Unsupervised learning

4.1.2.1 Manifold learning

4.1.2.2 Clustering

4.1.3 Model selection and evalutation

4.1.3.1 Model selection strategies

4.1.3.2 Feature selection stability

4.1.3.3 Performance metrics

4.2 Computational requirements and implementations

- MPI
- GPU and accelerators

Part II

5 ADENINE: a data exploration tool

ADENINE is a machine learning framework designed for biological data exploration and visualization. Its goal is to help bioinformaticians achieving a first and quick overview of the main structures underlying their data. This software tool encompasses state-of-the-art techniques for missing values imputing, data preprocessing, dimensionality reduction and clustering. ADENINE has a scalable architecture which seamlessly work on single workstations as well as on high-performance computing facilities. ADENINE is capable of generating publication-ready plots along with quantitative descriptions of the results. In this paper we provide an example of exploratory analysis on a publicly available gene expression data set of colorectal cancer samples. The software and its documentation are available at `https://github.com/slipguru/adenine` under FreeBSD license.

5.1 What is data exploration?

In biology, as well as in any other scientific domain, exploring and visualizing the collected measures is an insightful starting point for every data analysis process. For instance, the aim of a biomedical study can be detecting groups of patients that respond differently to a given treatment, or inferring possible molecular relationships among all, or a subset, of the measured variables. In both cases, bioinformaticians will be asked to extract meaningful information from collections of complex and high-dimensional measures, such as NGS data.

In these cases, a preliminary Exploratory Data Analysis (EDA) is not only a good practice, but also fundamental before further and deeper investigations can take place. To accomplish this task, several machine learning and data mining techniques were developed over the years. Among those, we recall the combined use of the following classes of methods: (i) missing values imputing, (ii) data preprocessing, (iii) dimensionality reduction and (iv) unsupervised clustering (see Section 5.4).

5.2 Popular data exploration tools

In the last few years, a fair number of data exploration software and libraries were released. Such tools may be grouped in two families: GUI-based and command-line applications. Among the first group we recall Divvy [Lewis et al., 2013], a software tool that performs dimensionality reduction and clustering on input data sets. *Divvy* is a light framework; however, its collection of C/C++ algorithm implementations does not cover common strategies such as kernel principal component analysis [Schölkopf et al., 1997] or hierarchical clustering [Friedman et al., 2001] and it does not offer strategies to perform automatic discovery of the number of clusters. The most notable project that spans between the two families is *Orange* [Demšar et al., 2013], a data mining software suite that offers both visual programming front-end and Python APIs. In the context of data exploration, *Orange* can be successfully employed. However, in order to test different data analysis pipelines, each one must be manually created as it does not support their automatic generation. Moreover, large data sets are difficult to analyze with both *Divvy* and *Orange* as they can run only on a single workstation, lacking of distributed computing support.

5.3 ADENINE **overview**

Step	**Algorithm**	**Reference**
Imputing	mean	
	median	
	most frequent	
	k-nearest neighbors	[Troyanskaya et al., 2001]
Preprocessing	recentering	
	standardize	
	normalize	
	min-max	
Dimensionality reduction	principal component analysis (PCA)	[Jolliffe, 2002]
	incremental PCA	[Ross et al., 2008]
	randomized PCA	[Halko et al., 2011]
	kernel PCA	[Schölkopf et al., 1997]
	isomap	[Tenenbaum et al., 2000]
	locally linear embedding	[Roweis and Saul, 2000]
	spectral embedding	[Ng et al., 2002]
	multidimensional scaling	[Borg and Groenen, 2005]
	t-distributed stochastic neighbor embedding	[Van der Maaten and Hinton, 2008]
Clustering	k-means	[Bishop, 2006]
	affinity propagation	[Frey and Dueck, 2007]
	mean shift	[Comaniciu and Meer, 2002]
	spectral	[Shi and Malik, 2000]
	hierarchical	[Friedman et al., 2001]
	DBSCAN	[Ester et al., 1996]

Table 5.1: Pipeline building blocks available in ADENINE.

In this paper, we present ADENINE, a command-line Python tool for biological data exploration and visualization that, starting from a set of unsupervised algorithms, creates textual and graphical reports of an arbitrary number of pipelines. Missing data imputing, preprocessing, dimensionality reduction and clustering strategies are considered as building blocks for constructing data analysis pipelines. The user is simply required to specify the input data and to select the desired blocks. ADENINE, then, takes care of generating and running the pipelines obtained by all possible combinations of the selected blocks. Every algorithm implementation of the presented software tool is inherited, or extended, from SCIKIT-LEARN [Pedregosa et al., 2011] which is, to the best of our knowledge, the most complete machine learning open source Python library available online.

ADENINE natively supports data integration with the NCBI Gene Expression Omnibus (GEO) archive [Barrett et al., 2013], which data sets can be retrieved specifying their GEO accession number.

Thanks to its scalable architecture, ADENINE pipelines can seamlessly run in parallel as separate Python processes on single workstations or MPI[1] tasks in high-performance computing (HPC) cluster facilities. This remarkable feature allows to explore and visualize massive amounts of data in a reasonable computational time. Moreover, as ADENINE makes large use of NUMPY and SCIPY, it automatically benefits from their bindings with optimized linear algebra libraries (such as OpenBLAS or Intel® MKL).

5.4 ADENINE **implementation**

ADENINE is developed around the data analysis concept of *pipeline*. A pipeline is a sequence of the following fundamental steps: (i) missing values imputing, (ii) data preprocessing, (iii) dimensionality reduction and (iv) unsupervised clustering. For each task, different off-the-shelf algorithms are available (see Table 5.1).

Data collected in biomedical research studies often present missing values. Devising imputing strategies is a common practice [De Souto et al., 2015] to deal with such issue. ADENINE offers an improved version of the `Imputer` class provided by SCIKIT-LEARN. In addition to the pre-existent feature-wise *mean*, *median* and *most frequent* strategies, this extension presents the k-nearest neighbors imputing method proposed for microarray data in [Troyanskaya et al., 2001].

Collecting data from heterogeneous sources may imply dealing with variables lying in very different numerical ranges and this could have a negative influence on the behavior of data analysis techniques. To tackle this issue ADENINE offers different strategies to preprocess data, such as recentering, standardizing or rescaling.

The presented software includes a set of linear and nonlinear dimensionality reduction and manifold learning algorithms that are particularly suited for exploration and visualization of

[1] http://mpi-forum.org/

high-dimensional data. Such techniques rely on the fact that it is often possible to *decrease* the dimensionality of the problem estimating a low-dimensional embedding in which the data lie.

Besides offering a wide range of clustering techniques, ADENINE implements strategies and heuristics to automatically estimate parameters that yield the most suitable cluster separation. The optimal parameter selection of centroid-based algorithms follows the B-fold cross-validation strategy presented in Algorithm 1, where $\mathcal{S}(X, y)$ is the mean silhouette coefficient [Rousseeuw, 1987] for all input samples.

Algorithm 1 Automatic discovery of the optimal clustering parameter.

```
for clustering parameter $k$ in $k_1 \dots k_K$ do
    for cross-validation split $b$ in $1 \dots B$ do
        $X_b^{tr}, X_b^{vld} \leftarrow$ $b$-th training, validation set
        $\hat{m} \leftarrow$ fit model on $X_b^{tr}$
        $\hat{y} \leftarrow$ predict labels of $X_b^{vld}$ according to $\hat{m}$
        $s_b \leftarrow$ evaluate silhouette score $\mathcal{S}(X_b^{vld}, \hat{y})$
    end for
    $\bar{S}_k = \frac{1}{B} \sum_{i=1}^{B} s_i$
end for
$k_{opt} = \arg\max_{k} \, (\bar{S}_k)$
```

For affinity propagation [Frey and Dueck, 2007] and k-means [Bishop, 2006] clustering parameters can be automatically defined (*preference* and *number of clusters*, respectively). Mean shift [Comaniciu and Meer, 2002] and DBSCAN [Ester et al., 1996] offer an implicit cluster discovery. For hierarchical [Friedman et al., 2001] and spectral clustering [Shi and Malik, 2000], no automatic discovery of clustering parameters is offered. However, graphical aids are generated to evaluate clustering performance such as dendrogram tree and eigenvalues of the Laplacian of the affinity matrix plots, respectively.

5.5 Usage example

In this section we show how ADENINE can be used to perform two EDAs on a gene expression microarray data set obtained from the GEO repository (accession number GSE87211). This data set was collected in a recent medical study that aimed at understanding the underlying mechanism of colorectal cancer (CRC) as well as identifying molecular biomarkers, fundamental for the disease prognostication. It is composed of 203 colorectal cancer samples and 160 matched mucosa controls. The adopted platform was the Agilent-026652 Whole Human Genome Microarray, which was used to measure the expression of 34127 probe sets.

ADENINE offers a handy tool to automatically download the data set from the GEO repository given only its accession name. It also let the user select phenotypes and/or probe sets of interest.

Given these preferences, ADENINE automatically converts the data set from the *SOFT* format to a comma-separated values text file. To download the remote GEO data set specifying the tissue type as phenotype of interest we used the following command.

```
$ ade_GEO2csv.py GSE87211 --label_field characteristics_ch1.3.tissue
```

This automatically creates `GSE87211_data.csv` and `GSE87211_labels.csv` which contain gene expression levels and tissue type of each sample, respectively.

The first EDA aims at stratifying the samples according to their tissue type (mucosa or rectal tumor) this can be performed by executing the following command.

```
$ ade_run.py ade_config.py
```

Where `ade_config.py` is a configuration file which should look like the snippet below.

{config here}

Each `step` variable refers to a dictionary having the name of the building block as key and a list as value. Each list has a *on*$\backslash$*off* trigger in first position followed by a dictionary of keyword arguments for the class implementing the corresponding method. When more than one method is specified in a single step, or more than a single parameter is passed as list, ADENINE generates the pipelines composed of all possible combinations.

The configuration snippet above generates eight pipelines with similar structure. The first and the second halves have recentered and ℓ_2-normalized samples, respectively. Each sample is then projected on a 2D space by isomap or by linear, Gaussian or polynomial kernel PCA. k-means clustering with automatic cluster discovery is eventually performed on each dimensionality-reduced data set, as in Algorithm 1. Results of such pipelines are all stored in a single output folder. Once this process is completed, plots and reports can be automatically generated running the following command.

```
  $ ade_analysis.py results/ade_output_folder_YYYY-MM-DD_hh:mm:ss
```

The aim of the second EDA is to uncover the relationships among a set of genes known from the literature to be strongly associated with CRC. Specifically this signature is composed of the following genes: APC, KRAS, CTNNB1, TP53, MSH2, MLH1, PMS2, PTEN, SMAD4, STK11, GSK3B and AXIN2 [Schulz, 2005]. We also considered probe sets measuring expression level of the same gene, and we labelled them with a progressive number. Three partially overlapping sublists compose this signature.

S1) Genes fundamental for the progression of CRC (i.e. APC, KRAS, CTNNB1, TP53).

S2) Genes relevant in the *Wnt signaling pathway*, which is strongly activated in the first phases of CRC (i.e. APC, CTNNB1, GSK3B, AXIN2).

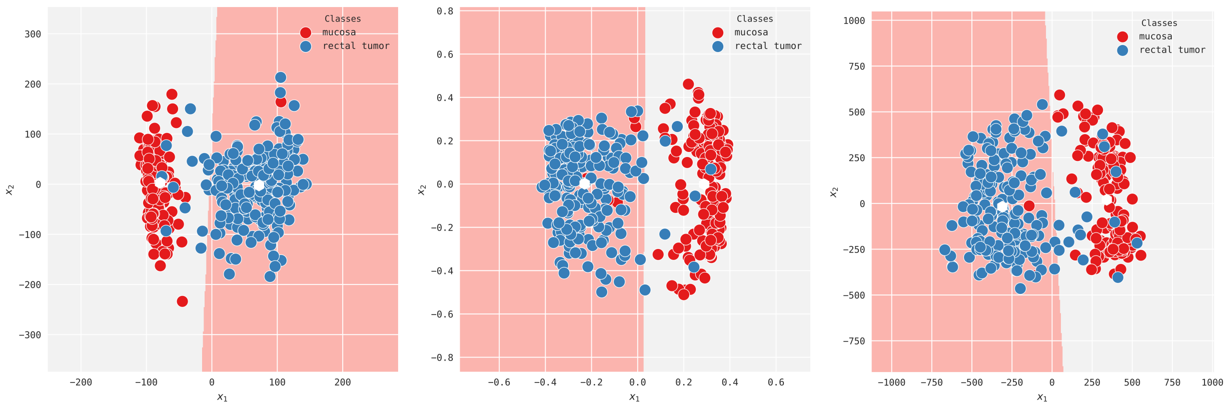
S3) Genes involved in hereditary syndromes which predispose to CRC (i.e. APC, MSH2, MLH1, PMS2, PTEN, SMAD4, STK11) [Schulz, 2005].

A reduced version of the GEO data set that comprises only such genes can be easily created calling `ade_GEO2csv.py` with the option `--signature GENE_1,GENE_2,...,GENE_N`. On the same line, the option `--phenotypes P_1,P_2,...,P_M` can be used to keep only mucosa or rectal tumor samples. To run such experiment, one simply needs to select and activate the hierarchical clustering building block and to follow the same steps presented above.

For ADENINE installation instructions and for a comprehensive description of all the options available in the configuration file we refer to the online documentation and tutorials[2].

5.6 Results

In the first EDA, we compared the clustering performance achieved by the eight ADENINE pipelines and we reported in Figure 5.1 an intuitive visualization of the results achieved by the top three, evaluated in terms of silhouette score [Rousseeuw, 1987]. As expected, the top performing pipelines show a clear separation between the two sample groups, as the k-means algorithm devises a domain partitioning that is consistent with the actual tissue types.



(a) (b) (c)

Figure 5.1: Three different 2D projections of the samples of the GEO gene expression data set used in this work. Projections on the left (a), middle (b) and right (c) panes are obtained via linear PCA, Gaussian PCA and isomap, respectively. The color of each point corresponds to the actual tissue type, while the background color is automatically learned by the k-means clustering algorithm. White hexagons correspond to cluster centroids.

[2] http://slipguru.github.io/adenine

For the second EDA, the relationships among the probe sets corresponding to the genes of the signature are separately explored learning a different hierarchical clustering [Friedman et al., 2001] tree for mucosa (Figures 5.2a) and CRC samples (Figure 5.2b), separately. The two trees are learned from different tissues, nevertheless they show some remarkable similarities. For instance, the pairs TP53-TP53.1 and MSH2-PMS2.1 always share a common parent. Interestingly, the first is a relationship between probe sets of the same gene, and the second is confirmed in literature, as MSH2 and PMS2 are both involved in hereditary non-polyposis CRC, a syndrome that predisposes for CRC. Moreover, two probe sets of the the genes of *S1*, namely APC and CTNNB1, are consistently close to the root of the two trees. This suggest that the expression level of these two genes highly differs from the others. Two interesting differences between the two trees can also be noticed. First, most of the elements of the sublist *S3*, which contains genes that enhance the risk of developing CRC, tend to be grouped together in Figure 5.2b, while the same observation cannot be done for Figure 5.2a. Secondly, probe sets of the genes belonging to sublists *S2* and *S3* tend more to more closely connected in Figure 5.2b than in Figure 5.2a.

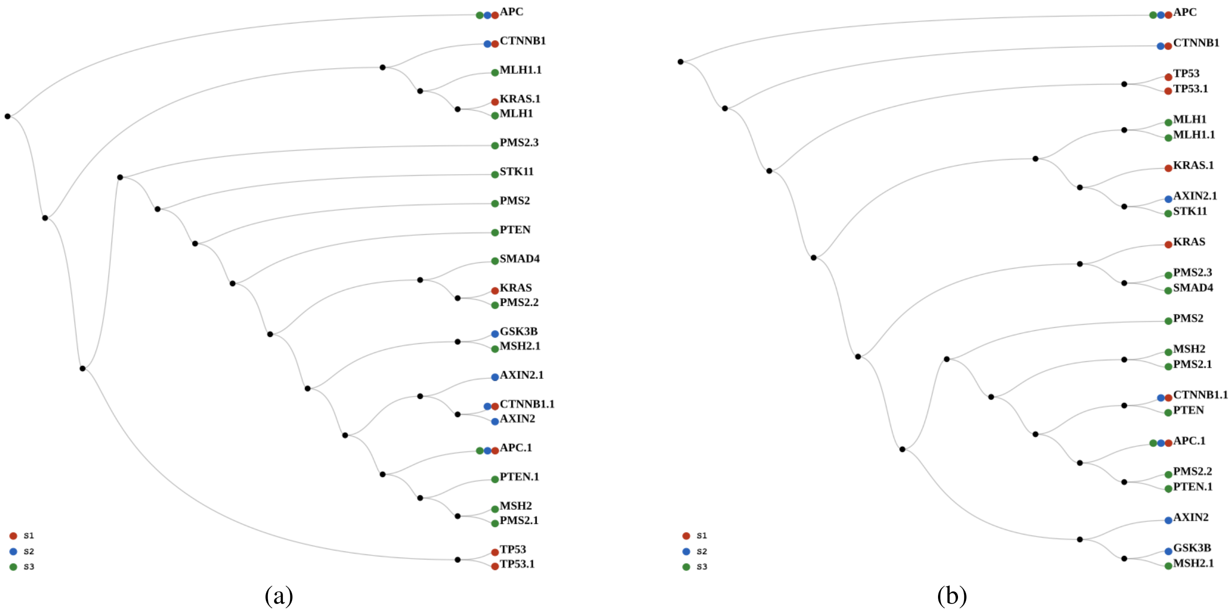


Figure 5.2: An example of hierarchical trees visualization learned by two ADENINE pipelines on mucosa (a) and CRC (b) samples. Each probe set is color coded according to the corresponding sublist. This visualization provides insights on the underlying structure of the measured gene expression level.

{...} In this paper we presented ADENINE, a biomedical data exploration and visualization tool that can seamlessly run on single workstations as well as on HPC clusters. Thanks to its scalable architecture, ADENINE is suitable for the analysis of large and high-dimensional data collections, that are nowadays ubiquitous in biomedicine.

ADENINE natively supports the integration with the GEO repository. Therefore, a user provided with the accession number of the data set of interest can select target phenotypes and genotypes and ADENINE takes care of automatically downloading the data and plugging them into the computational framework. ADENINE offers a wide range of missing values imputing, data preprocessing, dimensionality reduction and clustering techniques that can be easily selected and applied to any input data.

In this paper we showed ADENINE capabilities performing two EDAs on a CRC gene expression data set. From the obtained results we can observe that a clear discrimination between CRC and control samples can be achieved by unsupervised data analysis pipeline. Moreover, a meaningful description of the relationships among the group of genes strongly associated with CRC can be represented as hierarchical trees.

6 Model for metabolic age prediction

7 Temporal model for multiple sclerosis evolution

8 Temporal model for glucose predictions

9 Conclusion

Bibliography

Abraham, G., Kowalczyk, A., Zobel, J., and Inouye, M. (2013). Performance and robustness of penalized and unpenalized methods for genetic prediction of complex human disease. *Genetic Epidemiology*, 37(2):184–195. [Cited on page 12.]

Alexandrov, L. B., Nik-Zainal, S., Wedge, D. C., Aparicio, S. A., Behjati, S., Biankin, A. V., Bignell, G. R., Bolli, N., Borg, A., Børresen-Dale, A.-L., et al. (2013). Signatures of mutational processes in human cancer. *Nature*, 500(7463):415–421. [Cited on page 13.]

Altmann, A., Toloşi, L., Sander, O., and Lengauer, T. (2010). Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347. [Cited on page 11.]

Angermueller, C., Pärnamaa, T., Parts, L., and Stegle, O. (2016). Deep learning for computational biology. *Molecular systems biology*, 12(7):878. [Cited on pages 12 and 13.]

Appenzeller, T. (2017). The ai revolution in science. *Science*. [Cited on page 15.]

Argyriou, A., Evgeniou, T., and Pontil, M. (2008). Convex multi-task feature learning. *Machine Learning*, 73(3):243–272. [Cited on page 12.]

Baldassarre, L., Rosasco, L., Barla, A., and Verri, A. (2012). Multi-output learning via spectral filtering. *Machine learning*, 87(3):259–301. [Cited on page 12.]

Barrett, T., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Holko, M., et al. (2013). Ncbi geo: archive for functional genomics data sets—update. *Nucleic acids research*, 41(D1):D991–D995. [Cited on page 23.]

Bishop, C. M. (2006). Pattern recognition. *Machine Learning*. [Cited on pages 22 and 24.]

Borg, I. and Groenen, P. J. (2005). *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media. [Cited on page 22.]

Button, K. S., Ioannidis, J. P., Mokrysz, C., Nosek, B. A., Flint, J., Robinson, E. S., and Munafò, M. R. (2013). Power failure: why small sample size undermines the reliability of neuroscience. *Nature Reviews Neuroscience*, 14(5):365–376. [Cited on page 11.]

Candès, E. J. and Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717. [Cited on page 14.]

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619. [Cited on pages 22 and 24.]

Consortium, E. P. et al. (2004). The encode (encyclopedia of dna elements) project. *Science*, 306(5696):636–640. [Cited on page 10.]

De Souto, M. C., Jaskowiak, P. A., and Costa, I. G. (2015). Impact of missing data imputation methods on gene expression clustering and classification. *BMC bioinformatics*, 16(1):64. [Cited on page 23.]

Demšar, J., Curk, T., Erjavec, A., Gorup, Č., Hočevar, T., Milutinovič, M., Možina, M., Polajnar, M., Toplak, M., Starič, A., et al. (2013). Orange: data mining toolbox in python. *The Journal of Machine Learning Research*, 14(1):2349–2353. [Cited on page 22.]

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231. [Cited on pages 22 and 24.]

Evgeniou, T., Pontil, M., and Poggio, T. (2000). Regularization networks and support vector machines. *Advances in computational mathematics*, 13(1):1–50. [Cited on page 18.]

Frey, B. J. and Dueck, D. (2007). Clustering by passing messages between data points. *science*, 315(5814):972–976. [Cited on pages 22 and 24.]

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin. [Cited on pages 22, 24, and 27.]

Garg, R. P., Dong, S., Shah, S. J., and Jonnalagadda, S. R. (2016). A bootstrap machine learning approach to identify rare disease patients from electronic health records. *CoRR*, abs/1609.01586. [Cited on page 11.]

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422. [Cited on pages 11 and 13.]

Halko, N., Martinsson, P.-G., and Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288. [Cited on page 22.]

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning*, volume 2. Springer. [Cited on pages 13 and 18.]

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical learning with sparsity: the lasso and generalizations*. CRC Press. [Cited on page 13.]

He, D., Kuhn, D., and Parida, L. (2016). Novel applications of multitask learning and multiple output regression to multiple genetic trait prediction. *Bioinformatics*, 32(12):i37–i43. [Cited on page 12.]

Helmstaedter, M., Briggman, K. L., Turaga, S. C., Jain, V., Seung, H. S., and Denk, W. (2013). Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature*, 500(7461):168–174. [Cited on page 12.]

Jack, C. R., Bernstein, M. A., Fox, N. C., Thompson, P., Alexander, G., Harvey, D., Borowski, B., Britson, P. J., L Whitwell, J., Ward, C., et al. (2008). The alzheimer's disease neuroimaging initiative (adni): Mri methods. *Journal of magnetic resonance imaging*, 27(4):685–691. [Cited on page 10.]

Jolliffe, I. (2002). *Principal component analysis*. Wiley Online Library. [Cited on page 22.]

Jung, K., Dihazi, H., Bibi, A., Dihazi, G. H., and Beißbarth, T. (2014). Adaption of the global test idea to proteomics data with missing values. *Bioinformatics*, 30(10):1424–1430. [Cited on page 14.]

Kolker, E., Higdon, R., Haynes, W., Welch, D., Broomall, W., Lancet, D., Stanberry, L., and Kolker, N. (2012). Moped: model organism protein expression database. *Nucleic acids research*, 40(D1):D1093–D1099. [Cited on page 10.]

Lewis, J. M., De Sa, V. R., and Van Der Maaten, L. (2013). Divvy: fast and intuitive exploratory data analysis. *The Journal of Machine Learning Research*, 14(1):3159–3163. [Cited on page 21.]

Mamoshina, P., Vieira, A., Putin, E., and Zhavoronkov, A. (2016). Applications of deep learning in biomedicine. *Molecular pharmaceutics*, 13(5):1445–1454. [Cited on page 13.]

Masecchia, S., Coco, S., Barla, A., Verri, A., and Tonini, G. P. (2015). Genome iny model of metastatic neuroblastoma tumorigenesis by a dictionary learning algorithm. *BMC medical genomics*, 8(1):57. [Cited on page 13.]

McNeish, D. M. and Stapleton, L. M. (2016). The effect of small sample size on two-level model estimates: A review and illustration. *Educational Psychology Review*, 28(2):295–314. [Cited on page 11.]

Meier, L., Van De Geer, S., and Bühlmann, P. (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71. [Cited on page 13.]

Min, S., Lee, B., and Yoon, S. (2016). Deep learning in bioinformatics. *arXiv preprint arXiv:1603.06430*. [Cited on page 12.]

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press. [Cited on page 15.]

Ng, A. Y., Jordan, M. I., Weiss, Y., et al. (2002). On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856. [Cited on page 22.]

Okser, S., Pahikkala, T., Airola, A., Salakoski, T., Ripatti, S., and Aittokallio, T. (2014). Regularized machine learning in the genetic prediction of complex traits. *PLoS Genet*, 10(11):e1004754. [Cited on pages 11 and 12.]

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. [Cited on page 23.]

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian processes for machine learning*, volume 1. MIT press Cambridge. [Cited on page 16.]

Ross, D. A., Lim, J., Lin, R.-S., and Yang, M.-H. (2008). Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141. [Cited on page 22.]

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65. [Cited on pages 24 and 26.]

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326. [Cited on page 22.]

Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In *Artificial Neural Networks—ICANN'97*, pages 583–588. Springer. [Cited on page 22.]

Schulz, W. (2005). *Molecular biology of human cancers: an advanced student's textbook*. Springer Science & Business Media. [Cited on pages 25 and 26.]

Service, R. F. (2017). Ai is changing how we do science. get a glimpse. *Science*. [Cited on page 15.]

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905. [Cited on pages 22 and 24.]

Stekhoven, D. J. and Bühlmann, P. (2011). Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118. [Cited on page 14.]

Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323. [Cited on page 22.]

Tikhonov, A. (1963). Solution of incorrectly formulated problems and the regularization method. In *Soviet Math. Dokl.*, volume 5, pages 1035–1038. [Cited on page 18.]

Toga, A. W. and Dinov, I. D. (2015). Sharing big biomedical data. *Journal of big data*, 2(1):7. [Cited on page 10.]

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. (2001). Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525. [Cited on pages 14, 22, and 23.]

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85. [Cited on page 22.]

Yu, D., Huber, W., and Vitek, O. (2013). Shrinkage estimation of dispersion in negative binomial models for rna-seq experiments with small sample size. *Bioinformatics*, 29(10):1275–1282. [Cited on page 11.]