

POLITECNICO DI MILANO
DEPARTMENT of Electronics, Information, and Bioengineering
DOCTORAL PROGRAMME IN Information Technology

Novel Genome-Scale Data Models and Algorithms for Molecular Medicine and Biomedical Research

Doctoral Dissertation of:
Fernando Palluzzi

Supervisor:
Prof. Ivan Gaetano Dellino

Tutor:
Prof. Barbara Pernici

The Chair of the Doctoral Program:
Prof. Andrea Bonarini

Year 2016/2017 – Cycle XXVIII

All human beings are members of one frame,
Since all, at first, from the same essence came.
When time afflicts a limb with pain
The other limbs at rest cannot remain.
If thou feel not for other’s misery
A human being is no name for thee

~

سعدی شیرازی
Saadi Shirazi

Abstract

Genomes are **complex systems**, and not just static and monolithic entities. They represent the ensemble of the **hereditary information** of an **organism** or, in evolutionary terms, of a **species**. **Sequencing sciences** offered through the last two decades the possibility to dive into this complexity. This scientific revolution led to a series of paradigm changes, and the collateral need of efficient technical and methodological innovations. However, this process was not gradual, exposing genome research to an explosive **data deluge**, thus causing a series of **bottlenecks**: **technological**, **computational**, and **methodological**, due to the existence of efficient data reduction and analysis algorithm, capable of capturing the **informative** portion of genomic complexity. In the last years, there have been huge efforts towards the generation of **unified standards** and **databases**, **ontologies**, **integrated platforms**, **integrative methodologies**, to provide a strong theoretical background, capable of capturing **true biological variation**, often masked behind **marginal synergic processes**. The goal of this thesis was to shed light on this last problem, trying to expose a **unifying theoretical foundation** needed to approach the study of genome complexity.

Summary

In the present work, we investigated the fundamental modelling strategies in genome research, to expose a **unifying theoretical foundation** needed to approach the study of genome complexity. We showed that this unifying theory exists, and it is represented by the **Kullback-Leibler Divergence** (KLD). The KLD is further supported by the **sampling theory**, that allowed us to **produce** and **rank** different representations of reality. We gave a generic definition of our modeling paradigm as the **Genomic Interaction Model** (GIM), that is deeply founded over data, and manipulated through the generic **Genomic Vector Space** (GVS) data structure, composed by a **sampling space** *S*, a **variable space** *X* (that is defined by data itself, represented by sequencing **tag density profiles**), an **architecture** over data *H(X)*, and a monotonic variable *T* defining its **evolution**. In other terms, GVS = { *S*, *H(X)* }$_T$. The architecture *H* can be represented in different ways, from the more traditional **Generalized Linear Models** (GLM), to the more complex **interaction networks**, to the less used **Structural Equation Models** (SEM). Responding to the need for parallel computational solutions, we developed the **GenoMetric Query Language** (GMQL) as an **Apache/Pig-based** language for genomic cloud computing, ensuring clear and compact user coding for: (i) searching for relevant biological data from several sources; (ii) formalize biological concepts into rules for genomic data and region association; (iii) integrate heterogeneous kinds of genomic data; (iv) integrate publicly available and custom data. However, similarly to other available methods for genomic data modelling and algebra implementation (such as BEDTools, BEDOPS, GQL, and GROK), a declarative-like approach may underperform in modelling complex interactions. In general, feature definition may involve multiple combinations of variables, including **distance**, a (continuous) **dynamic range** of **binding**, **transcriptional**, and **epigenetic signals**, with **interaction terms** connecting all or some of these variables.

Here we come to one fundamental definition of this work: a ***genomic feature*** *is represented by a specific* ***chromatin state****, defined through* ***context-specific variables*** *that can be* ***profiled using sequencing technologies****, and which relationships can be* ***modeled and tested through an interaction model***. In this way, we can represent every probable chromatin state (as a **functional element**) and possible alteration (i.e. **physiological and pathological functional change**).

There are at least five main conceptual improvements in this definition:

- **Genomic features are not fixed entities**.
- **Genomic features are knowledge-independent**.
- **Unknown genomic features are included in the modeling framework**.
- **Time-variant interactions are part of the feature definition**.
- **Genomic coordinates identify an instance of a chromatin state**.

Our approach demonstrated its flexibility and efficacy in two different human complex disorders. Specifically, our interaction-based statistical modeling strategy allowed us to investigate the basal mechanisms of DNA damage occurrence in **breast cancer** (BC), and to discover new, subtle and synergic biomarkers in the etiology of **frontotemporal dementia** (FTD).

In the former, we applied an **ensemble logistic-RFC model** (based on Random Forest Classifiers), that evidenced the accumulation and causal effect of gene length and RNA Polymerase II in promoters of damaged genes, significantly associated with the occurrence of translocations and gene fusions in breast cancer patients. Our methodology enabled us also to quantitatively measure thresholds for single and interacting predictors, excluding other potential causes of DNA damage (including gene expression and the presence of common fragile sites). The overall predictive accuracy of our algorithms was ~83% (i.e. we correctly predicted more than 8 damaged promoters every 10 trials, based only on functional signals at promoters, including gene length, RNA Polymerase II, histone H3K4me1, and gene expression). We also demonstrated that there are specific genomic regions, i.e. intron-exon junctions, may be prone to pathogenic translocation events in BC, in which the probability of a DNA double-strand break (DSB) to cause a gene fusion is directly proportional to the distance between the DSB event and the junction.

In the latter project, we identified novel FTD biomarkers, and characterized the perturbation of their interactions, to generate a novel oxidative stress-based theory for the etiology of this complex neurological and behavioral disorder. In this case, we instantiated the genomic interaction model into a **SEM-based network**, coupled with a weighted shortest-path Steiner’s tree-based heuristic algorithm.

Firstly, the molecular basis of this theory includes Calcium/cAMP homeostasis and energetic metabolism impairments as the primary causes of loss of neuroprotection and neural cell damage, respectively. Secondly, the altered postsynaptic membrane potentiation due to the activation of stress-induced Serine/Threonine kinases leads to neurodegeneration. Our study investigated the molecular underpinnings of these processes, evidencing key genes and gene interactions that may account for a significant fraction of unexplained FTD etiology.

Contents

1. Chapter 1: Introduction to Sequencing Sciences
1.1. Preface
1.2. From DNA to Genomes
1.3. Chromatin State and Dynamics
1.4. DNA Sequencing Technologies and Historical Background
1.5. DNA Sequencing Standards and Bio-Ontologies

2. Chapter 2: Biology and Medicine as Big Data Sciences
2.1. Data Complexity and Genomics as a Hypothesis-driven science.
2.2. NGS Technology Perspectives.

3. Chapter 3: Genomic Data Modeling
3.1. Conceptual Genomic Data Modeling.
3.2. The Region-based Genomic Data Model.
3.3. Biochemical Signatures and the Genomic Interaction Model.
3.4. The Nature of Biological Interactions.
3.5. Modelling and Testing Biological Interactions.
3.5.1. Models for Tag Density Peak Detection.
3.5.2. Over-Dispersion, Zero-Inflation, and Negative Binomial Model.
3.5.3. Modelling Count Data Using Generalized Linear Models.
3.5.4. Generalized Linear Models for Modelling (Latent) Categorical Features.
3.5.5. Model Complexity, Bias-Variance Tradeoff, and Model Selection.

4. Chapter 4: Applications in Genome Research
4.1. Genomic instability in Breast Cancer.
4.1.1. Modeling the Association Between Risk Factors and DNA Damage.
4.1.2. Modeling Non-Linear Interactions: Fusion-DSB Association.
4.1.3. Modeling Double Strand Brakes.
4.2. Discovering the biomarkers of Frontotemporal Dementia.
4.2.1. Modelling Genetic Variability Through Genomic Vector Spaces.
4.2.2. Modelling Perturbation in a Structural Model.

Concluding Remarks
References

CHAPTER 1

Introduction to Sequencing Sciences

And this I believe:
that the free, exploring mind of the individual human
is the most valuable thing in the world.
And this I would fight for: the freedom of the mind
to take any direction it wishes, undirected.
And this I must fight against: any idea, religion,
or government which limits or destroys the individual.
This is what I am and what I am about.

John Steinbeck
East of Eden, 1952

1.1. Preface

Sequencing sciences radically changed the way researchers conceive genomes and genome-associated processes, construct new hypotheses and develop the experimental procedures that replaced many old-fashioned technologies, including microarrays. Generally, current sequencing technologies are called **High-Throughput Sequencing** (**HTS**), to highlight the high quality of the resulting massively parallel synthesis/scanning process, or **Next Generation Sequencing** (**NGS**), to underline the improvement respect to the classical low-throughput capillary sequencing. The real potential of HTS is represented by its versatility: the capability to answer to radically different biological questions with relatively few changes to the basic

protocols, at reduced costs. This led to an explosion of data production and data types, with a plethora of (often slightly) different methodologies to store, index, retrieve and hopefully distribute and analyze them. Data production is no longer a problem, also thanks to platforms allowing external data and knowledge integration. Until few years ago, the main limitation to the production of new knowledge from HTS data was the availability of efficient methodological solutions: a condition known as the **computational bottleneck** [88, 90]. The development of novel **heuristic algorithms** and **cloud computing** platforms allowed data analysts to overcome this problem, revealing the true **genome complexity** (and, in general, biological systems complexity). Since the advent of high throughput technologies (including microarrays and NGS), sequencing sciences become more and more important also in already well-established studies, such as the simultaneous genotyping and analysis of millions of **sequence polymorphisms**, that were previously analyzed only through SNP arrays in GWAS. Hence, in relatively few years, sequencing sciences witnessed **multiple paradigm shifts**: from genome-wide and low-coverage sequencing to **single-cell sequencing**, from sequence-based analyses to **epigenetics**, from simple hypothesis testing in local environments to **integrated heuristic methods** running over **distributed systems**. Although technological solutions play a critical role in enabling genome research, understanding the true complexity of human genotypes and phenotypes cannot be afforded without an efficient theoretical background. This prepares ground for a last sequencing sciences revolution: from multiple method competition, to an integrated and comprehensive theory for modeling genome complexity. Although generating such a theory is far beyond the purposes of this thesis, the following discussions will dive into genome complexity and find the theoretical core that could best represent it. In

other terms, the subject of this work can be stated through the following question:

Given a formal definition of genome G and a biological process $f \in G$, is it always possible to define a valid model $\hat{f}$ capable of efficiently represent f?

Here, an **efficient representation** identifies a model (i.e. an approximation) that is **arbitrarily close** to **reality** (i.e. G). This implies having a correct and computable definition for G, an exhaustive **description** of f and a principle for assessing the absolute **distance** between $\hat{f}$ and f. Of course, if we think about a specific biological process (e.g. transcription), this is the issue underlying almost every algorithm for genome analysis. But here the question is more generally expressed, and can be better stated by rephrasing the problem:

Given a set of observations X, that we assume to contain the full description of a genomic process $f(X)$, is there any theory capable of generating a model $\hat{f}(X)$ sufficiently close to it?

The first formulation of the problem implies having a correct description of reality (G). This is often the assumption of many conceptual models (e.g. the relational model) and statistical approaches (e.g. the Bayesian approach). Specifically, these approaches assume that the true description of reality lies within the set of possible models. Although describing reality is a fascinating concept, the idea of a synthetic genome (i.e. a model of the whole reality) is not necessary (as an input). Rather, the description of reality should be the desired output. Similarly, the description of a biological process f is necessary only to define the biological question, but it has

no role in generating the model. In the second formulation, we only have two working assumptions:

(i) data contain the information that is necessary to describe the true biological phenomenon f;

(ii) there exists a theory that allows us to extract this information and build a model $\hat{f}(X)$ of the true biological process $f(X)$;

As a corollary of the second assumption, the theory should also provide a way to assess the divergence between the real process $f(X)$ and the related model $\hat{f}(X)$.

Currently, genomes are described in terms of their **features** and the **time-dependent multi-level interactions** characterizing them [36-38, 90, 125-132]. This means that the genome is an evolving entity, organized in hierarchical levels of complexity. The problem of representing this system is that there is no way to assess neither its actual complexity, nor the exact state it will reach at a given time. In other terms, uncertainty is part of the model. Indeed, the most common representation of the genome is static and determined. We will call this representation the **Region-Based Model** (RBM). In the RBM, genome is a collection of segments, each having the length of a chromosome. Every genomic feature is a segment of a chromosome, called **locus**. This is basically the structure of what is defined a karyotype. This basic structure is obviously not sufficient to account for the complex interactions among genomic loci. The RBM, instead, is a simple and useful way to visualize the status of the knowledge, with poor performances in terms of computation. If we now go back to the initial question, in its second formulation, we do not need the representation of the universe (i.e. the genome) to efficiently describe part of it (i.e. the biological process). What we need, is to have a description of the interactions among genomic features, possibly

through time. By describing these interactions, we will be able also to define genomic features and their properties. This interaction-oriented modeling strategy will be called the **Genomic Interaction Model** (GIM). It will be shown that the unifying theory underlying this proposed model is the well-known **Kullback-Leibler Divergence** (KLD)

$$I(f,\hat{f}) = \int f(X)\ln\frac{f(X)}{\hat{f}(X)}dX$$

defining the information-loss we experience passing from reality (f) to the representation of it ($\hat{f}$). In this case, the **sample** (S) will define our **feature space**, and the measurement we make over them through HTS experiments, will define the **variable** set (X) describing the features. Possible data values are called **profiles**, that are basically density functions with the genome as a support. The aim of interaction modeling is to provide an efficient computable representation M_j to find the **interaction architecture** (η) that minimizes the KLD. In general, the space of possible models will be defined as $M = \{S,\eta(X)\}_t$, where t describes the model evolution through time. Model architecture is the critical element that must be tested, while the feature space S is generally represented by known structural genomic elements, such as promoters, transcripts, or enhancers, that are chosen *a priori*. In alternative, also features can be derived from data, following a process defined **feature calling**.

Since the interaction model is basically data-driven, every theory describing X will be also supportive of the interaction model. These supportive theories include the **likelihood theory**, the **sampling theory**, and in general, the **information theory**. Chapter 1 introduces the concepts of genome and genome sequencing, while Chapter 2 will

enfold this discipline in the context of **big data sciences**. Chapter 3 will describe genomic models to handle feature definition and to test complex hypotheses over their interactions, with an exposition of the general theoretical framework, in Paragraphs 3.5.5 and 3.5.6. Finally, Chapter 4 will present different applications in two main genomic contexts: **cancer research** and **complex neurodegenerative disorders**.

1.2. From DNA to genomes

Deoxyribonucleic Acid (DNA) is a **biological macro-molecule**, carrying **inheritable information** in living systems, including mono- and multicellular organisms, and viruses. **Ribonucleic Acid** (RNA) may also carry genetic information, instead of DNA, in RNA-based viruses. RNA is also responsible for safe-copying and amplifying genomic information (**messenger RNA**, mRNA), translating this information into amino acids (**transfer RNA**, tRNA), combining tRNAs to generate polypeptides (**ribosomal RNA**, rRNA), silencing translation (**micro-RNA**, miRNA), generating transcript variability through splicing (**small nuclear RNA**, snRNA; see below), post-processing other RNAs (**small nucleolar RNA**, snoRNA), and other. DNA and RNA are called **Nucleic Acids**, i.e. **biopolymers** constituted by five basic **nucleosides**: **Guanosine** (G), **Cytidine** (C), **Adenosine** (A) and **Thymidine** (T; DNA only) or **Uridine** (U; RNA only). Once incorporated in a nucleic acid chain, nucleosides are called **nucleotide residues**, or simply nucleotides. **Hydrogen bonding** between nucleotides lead to G≡C pairing in both DNA and RNA, and A=T or A=U pairing in DNA or RNA, respectively (Figure 1). Every nucleotide contains a corresponding **nucleobase**: **Guanine** (G), **Cytosine** (C), **Adenine** (A), **Thymine** (T) and **Uracil** (U). G≡C interactions (3 hydrogen bonds) are stronger than A=T/A=U ones (2 hydrogen bonds), giving to the DNA/RNA sequence

peculiar biophysical and biochemical properties. Pairs of nucleobases interact each other in a plane with a characteristic electron density, given by the aromatic cycles they contain. While hydrogen bonds (the strongest non-covalent chemical bonding) keep two nucleotide strands together, **stacking interactions** between base-base planes (weak non-polar interactions) give the **DNA double helix** its characteristic and well known structure (Figure 2). Although DNA was isolated in 1869 by Friedrich Mieschner for the first time, DNA double-helix structure was first resolved in 1953 by **James Watson and Francis Crick** [1], thanks to the crystallized x-ray structures studied by **Rosalind Franklin**. This double-helix structure is not symmetrical respect to the central axis, generating two **grooves**: the **minor** and **major groove**, fundamental for **DNA-protein interactions**. While the base-dependent non-polar stacking interactions are placed internally the DNA structure, a **phosphate-group backbone** is exposed outwards, in the aqueous medium.

Figure 1. Nucleobase G≡C and A=T pairing in a DNA chain.

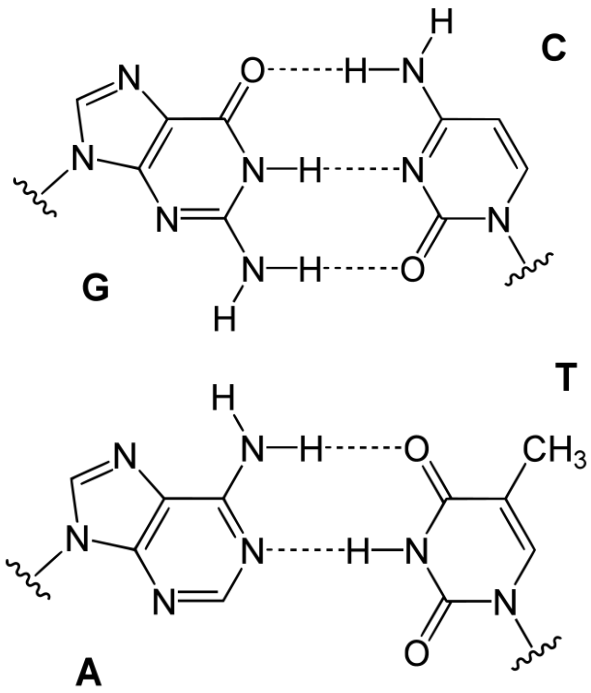
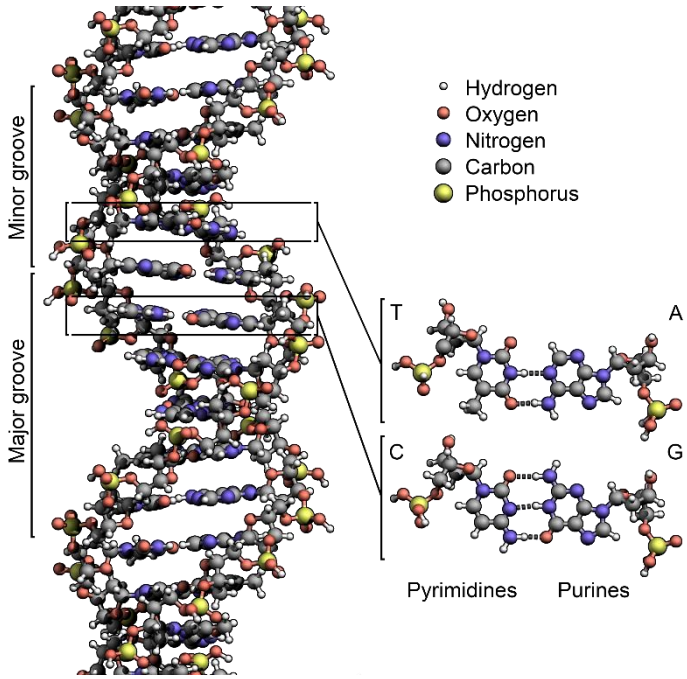


Figure 2. Typical DNA double helix structure (from http://www.richardwheeler.net/contentpages/index.php).



The negative electron density generated by the phosphate groups surrounding DNA determines its acidic nature and enhance its capability of interacting with proteins. DNA-protein interaction keeps DNA in the **nucleoid** of **prokaryotes** (eubacteria and archaea) and **chromatin** inside the **nucleus** of **eucaryotes**, including protozoa, molds, yeasts, fungi, algae, plants, and animals. The double-helix backbone is composed by an alternation of *n*·(nucleobase-phosphate) bindings. Therefore, every deoxyribose molecule belonging to one nucleobase is bound to two phosphate groups (**phosphodiester bond**): one at the 3'-carbon and the other at the 5'-carbon. The terminal nucleobases have either a free phosphate group (5' end) or a free –OH (i.e. hydroxyl) group (3' end). Since the two strands are complementary they must have an anti-parallel orientation. By convention, the **forward** (i.e. **positive**) strand is represented with 5' on the left (gene start) and the 3' on the right (gene end). The **reverse** (i.e.

negative) strand, that is the forward’s **reverse complement**, is represented with opposite direction.

Genomic DNA is the hardware support used to backup what we define the **genome**. The genome represents the **inheritable portion of an organism variation**; in other words, its **hereditary information**. Indeed, large part of this inheritable variability does not depend on the DNA sequence itself, but on the set of biochemical modifications to the DNA (e.g., the addition or removal of methyl groups to the cytosine of CpG dinucleotides), or to a class of structural DNA-binding proteins, called **histones**. The layer of biochemical modifications that do not alter the DNA sequence in response to the interaction between the genome, the physiological context and the external environment is called the **epigenome** [36, 38, 40]. Due to **intrinsic instability** and **environmental pressure**, the genome is not a fixed entity. Genome variation may occur by **spontaneous mutation**, **replication/transcription-associated DNA damage**, and **DNA repair errors**. When a DNA sequence variant is rare (i.e. < 1% in the population), it is classified as a **rare variant**, that is the primary source of genomic variation. For instance, Activation-induced Cytidine Deaminases (AID) are enzymes that induce mutations by deamination of cytosine into uracil, causing C→T transitions during DNA replication. These kind of activity is used in lymph nodes to increase B cell genome mutation rate in regions that are important for the antibody-antigen interaction, to efficiently react to a wide spectrum of pathogens. By contrast, the same mechanism may lead to B cell lymphoma in susceptible individuals [13]. Another example is the increased **DNA damage rate** during normal transcriptional activity. This DNA damage is generally repaired, but in case of stress it might lead to mutations and chromatin **double strand breaks** (DSBs) [14-16].

The genome is a complex system and its variability can be explained only through the relationships among its components. The first level of complexity is represented by the hierarchical structure of its linear sequence, measured in **base-pairs** (bp). **Genes** are genomic elements containing *the entire **nucleic acid sequence** needed to produce a **functional RNA***. The typical structure of a **coding gene** (i.e. genes encoding for proteins) consist of different elements (Figure 3), including the actual **coding DNA sequence** (CDS), representing only a minor portion of the entire gene length. The CDS is a discontinuous stretch of DNA, composed by segments called **Exons**. One gene usually have multiple exons that can be alternatively excluded, generating different **mRNA**s or **non-coding RNA**s from a single gene. Exons are spaced each other through **introns**. Introns are non-coding sequences needed to assemble exons together during **splicing**, but then discarded. Some introns may also contain functional RNAs, **mobile genetic elements**, and **enhancers**. The 5’ of a gene contains its **transcription start site** (TSS), corresponding to the first base of a transcript. The TSS is included in a regulatory region called **promoter**. The promoter region contains consensus binding sequences upstream the TSS for the **RNA Polymerase** binding, and the **5’UTR** sequence downstream the TSS. The promoter is not hard-encoded as a discrete DNA segment, but is rather detected by the genes 5’ regulatory region, identified by transcription factor binding, histone markers, and other functional properties. Promoter extension is conventionally and reasonably defined by the observer (that is usually, no larger than TSS ± 2.5kbp). The 5’UTR contains regulatory regions and the **ribosome binding site** (RBS), needed for **translation** (i.e. mRNA to polypeptide conversion). Translation starts from the ATG **codon** (encoding for the

amino acid Methionine), in eukaryotes. A codon is a sequence of 3 nucleotides that are converted into a single amino acid. There are also three **stop codons** (i.e. codons TAA, TAG, and TGA), that cause translation termination. The **3’UTR** is a non-translated region at the end of a gene. It contains regulatory sequences that are necessary to determine mRNA integrity, including **miRNA** target sites. The 3’-end of a gene terminates with the **Transcription Termination Site** (TTS). Before translation, every mRNA is **capped at 5’** and possibly a **poly-A tail** is added at the 3’. These modifications are needed to determine mRNA lifespan. One single gene may be converted in more transcripts through **alternative splicing**. Therefore, one gene can give rise to different proteins called **isoforms**. Different transcripts of a gene may have different TSS, some of which may produce non-coding RNAs. Transcriptional activity would not be efficient enough to sustain life without **enhancers** [21]. An enhancer is a genomic region capable of binding promoters and increasing the transcription rate, by connecting transcription factors with the RNA Polymerase II (Figure 4).

Figure 3. Coding gene elements and processing.

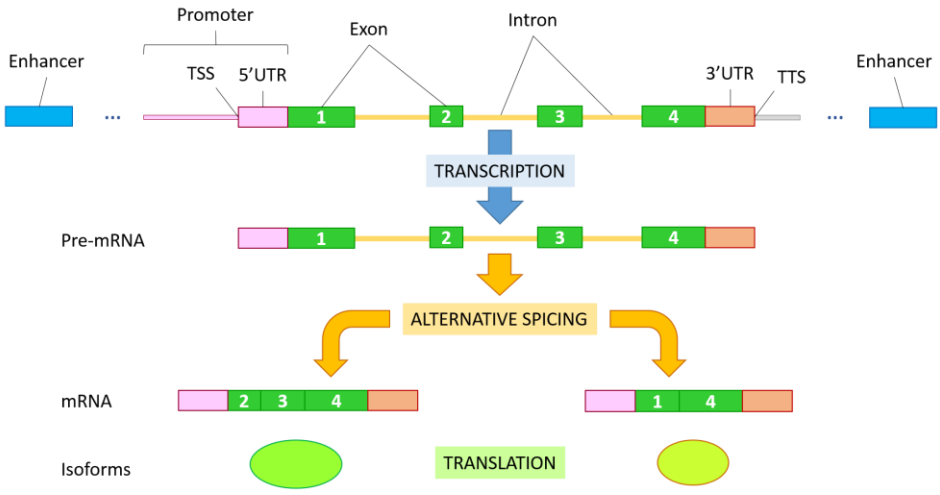
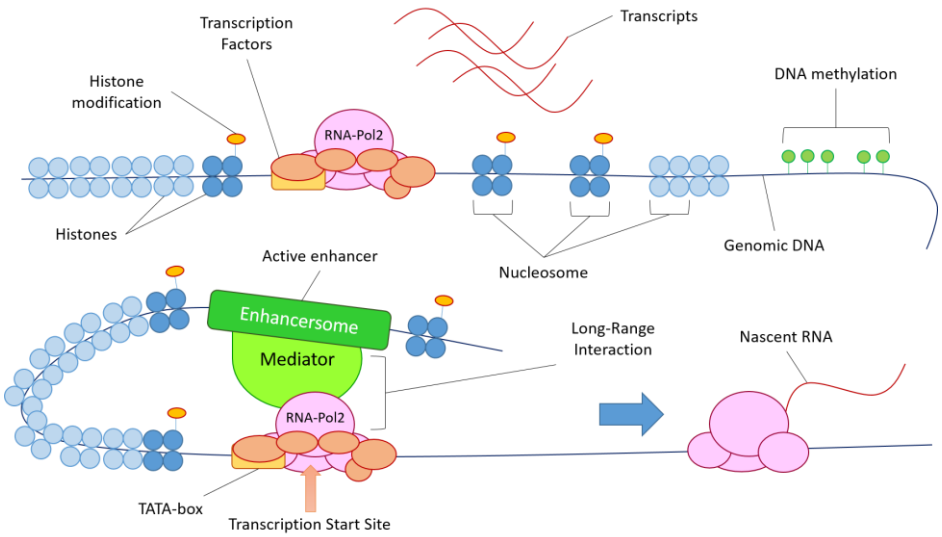


Figure 4. Schematic representation of the basic chromatin elements.



For this reason, enhancers are called **distal** genomic elements (although some enhancers may reside inside introns), and their interaction with promoters (i.e. **proximal** genomic elements) is a kind of **long-range chromatin interaction** [22-25].

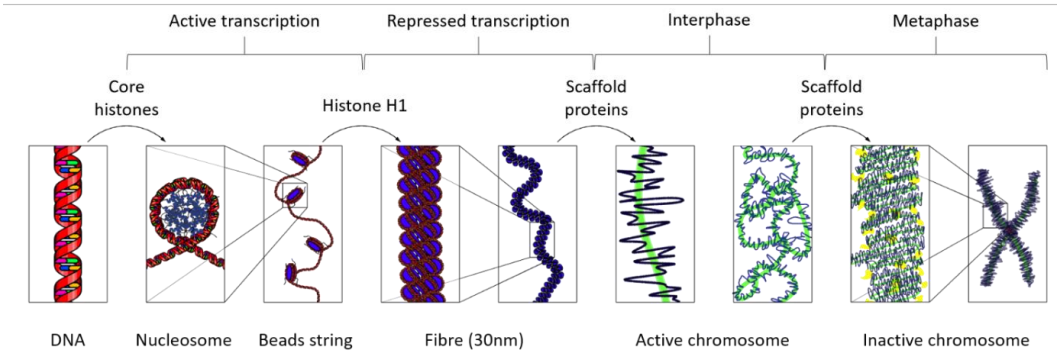
1.3 Chromatin state and dynamics

In contrast to Prokaryotes, Eukaryotic DNA is located inside the nucleus double membrane. Nuclear DNA is wrapped around **histones** and other **scaffold proteins**, depending on the specific **cell cycle** stage (Figure 5). This DNA-protein ensemble is called **chromatin**. During chromosome segregation (**metaphase**), DNA is condensed into an insoluble form, that is the classical “X-shape” visible to the optical microscope. During the first phase of cell division (**interphase**) chromatin is more relaxed, but still condensed. In these conditions both transcription and replication are repressed, because they require **accessible DNA**. Seen from electronic microscope, nuclear DNA shape appear as a bead string. These beads, called **nucleosomes**, are constituted by ~147bp of DNA (about 2 twists and half) wrapped

around a core **histone octamer** (2 copies of histones H2A, H2B, H3, and H4). In this shape, nucleosomes are spaced enough to allow **transcription factors**, **enzymes**, and **repair factors**, to access the DNA. Enzymes include **RNA Polymerases** (transcription-specific), **DNA Polymerases** (replication-specific), and **Topoisomerases I and II**, capable of breaking DNA in a controlled way, and thus reducing the tension due to double-strand **super-coiling**, during transcription and replication. However, chromatin can be repressed in the form of 30nm fibers, characterized by the presence of histone H1. This type of chromatin, tightly wrapped around nucleosomes, is not actively transcribed or replicated.

Chromatin conformational changes depend on a variety of endogenous and environmental stimuli. For instance, metabolic stress may induce cell and DNA damage due to the production of reactive oxygen species (ROS). ROS stimulate protein kinase cascades that activate the transcription of repair genes, thus modifying their activity. The same happens to immune-related genes during infections. Transcriptional and replicative modulation, as well as DNA damage, leave a precise chemical trace onto chromatin. Specifically, **chromatin dynamics** is related to precise **DNA** and **histone modification profiles**. Histone modifications influence both **chromatin contacts** and the **recruitment of other non-histone proteins**. These modifications occur at the N-terminus of histones, and include **Acetylation**, **Methylation**, **Phosphorylation**, **Ubiquitination**, **Sumoilation** of lysine (K), arginine (R), Serine (S), or Threonine (T) residues.

Figure 5. Chromatin remodeling in different genomic contexts.
See reference [U3].



For example, **H3K4me3** is the tri-methylation of lysine 4 of the histone H3, and is associated to promoters of actively transcribed genes [35-38]. Other common histone modifications are the **H3K27ac** (acetylation of lysine 27 of H3), associated to open chromatin at promoters and enhancers; **H3K36me3**, associated with transcriptional elongation in active genes; **H3K27me3**, associated with **Polycomb domains** (protein complexes binding inactive chromatin); and **γH2AX**, associated with DNA damage. Table 1 reports the most informative histone modifications, to date. **Signal profile** distribution, as well as **signal intensity** and **distance** from other genomic features, is as important as the nature of the modification itself. For example, histone mark **H3K4me1** is both present at active enhancers and promoters, but in the latter case it has a bimodal profile, with the valley centered at the TSS. Therefore, to describe the **dynamics** and the **biological function** of a specific chromatin modification, we need to specify also its genomic **context** and the possible **interaction** with other histone marks and DNA-binding proteins.

Table 1. Most informative histone modifications to date. *List of different chromatin signatures found in various genomic contexts and associated to specific chromatin states. Naming code for the Localization field: [P] promoter, [E] enhancer, [H] constitutive heterochromatin, [R] repressed chromatin, [B] gene body, [A] accessible chromatin.*

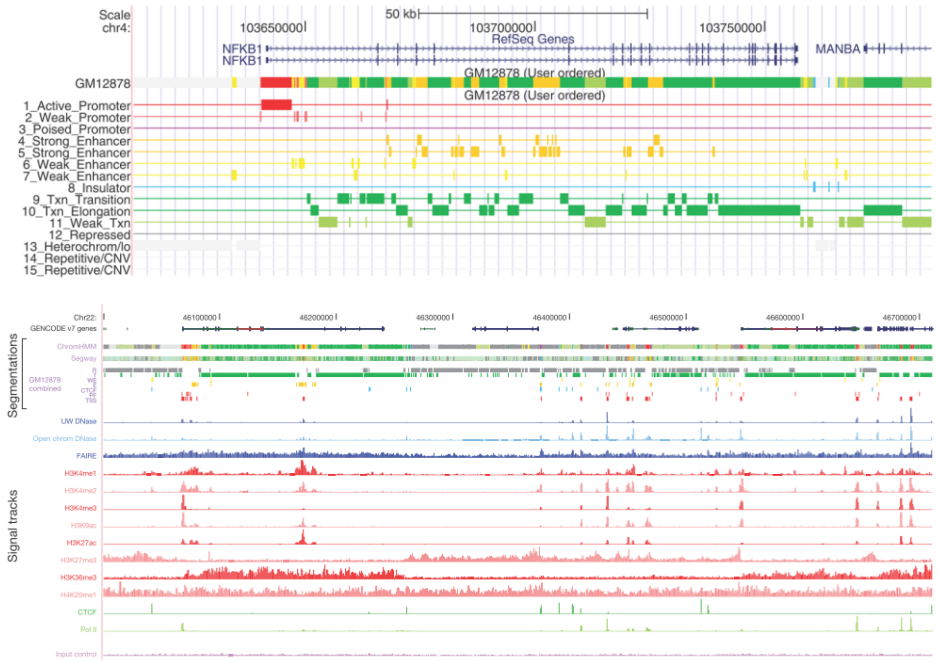
Histone mark	Localization	Effect
H3K4me1	P, E	Active transcription / Active enhancer
H3K4me2	P	Active transcription
H3K4me3	P	Active transcription
H3K9ac	P	Active transcription
H3K27ac	P, E	Open chromatin / Active enhancer
H3K9me3	H, R	Repressed / Inactive chromatin
H3K27me3	P, B, R	Polycomb-associated repressed chromatin
H3K36me3	B	Transcriptional elongation
H3K79me2	B	Active transcription
gammaH2AX	A	Damaged chromatin

Thanks to **ChIP-seq** technology (Chromatin Immunoprecipitation followed by sequencing) [26-34], these aspects have been extensively studied, producing histone modifications and binding protein **signatures** (Table 1), enabling researchers to clarify the interplay between chromatin dynamics and histone modifications. In 2003, the National Human Genome Research Institute (NHRGI) launched the **ENCODE project** (Encyclopedia Of DNA Elements), with the aim of systematically map regions of transcription, transcription factor binding, histone modification, and chromatin structure [35, 36]. According to the ENCODE definition, a **genomic functional element** is a **discrete genome segment** that encodes for a product (e.g. a protein or non-coding RNA), or displays a **reproducible biochemical signature** (i.e. a specific combination of biological signals).

Based on biochemical signatures, the project listed 7 different element types: (i) **CTCF enriched elements** (insulator), (ii) **predicted enhancers**, (iii) **weak enhancer / open chromatin**, (iv) **predicted promoters including TSS**, (v) **predicted promoter-flanking region**, (vi) **predicted transcribed region**, and (vii) **predicted repressed region** [35, 36]. These elements are generically known as **chromatin states**, a term that recalls the *state* of a Markov Model. The ENCODE project started on a dataset of 147 different cell lines (organized in 3 tiers and 1640

datasets), applying for the first time the problem of genome modelling and classification to large-scale genomic data. This first huge effort to learn chromatin state using large-scale functional datasets was based on the ChromHMM method, a multivariate Hidden Markov Model (HMM), modeling the observed combination of histone markers as a product of independent Bernoulli random variables [37]. The input for the ChromHMM system is a list of aligned reads (see Paragraph 1.5) that is automatically converted into a discrete genomic segment for each mark across the genome, through genome-wide signal enrichment testing (see Paragraph 3.5.2).

Figure 6. Example of ChromHMM classification. *See reference [37].*



The need for a comprehensive reference in epigenomics, human variation and diseases, gave rise in 2007 to the **Roadmap Epigenome project**, based on the integrative profiling of histone modifications, DNA methylation, chromatin accessibility, and RNA expression from 111 different cell lines, to date [38]. Althorugh the DNA sequence is almost completely conserved among cell lines, epigenomic signatures may deeply differ. This study revealed that ~5% of the whole

epigenome, on average, is marked by promoter- or enhancer-specific signatures, which showed a greater association with gene expression and increased conservation. On the other hand, two-thirds of the epigenome are, on average, quiescent and associated with gene-poor or nuclear lamina-associated (i.e. the fibrillar network on the internal surface of the nucleus) repressed chromatin. Furthermore, cell lineage-specific promoter- and enhancer-specific signatures are informative enough to allow a data-driven approach to **cell line and tissue classification**. Integration with the **GWAS catalog** [39] revealed that complex traits are significantly enriched in epigenomic markers, suggesting a tight **association between epigenome and compex disorders,** including **autoimmune diseases**, **neurological disorders**, and **cancer** [38]. Concurrent evidences demonstrated the interplay between histone modifications, DNA methylation, and mutations [40]. DNA can be directly modified by the addition or removal of a methyl group to che cytosine residues of a CpG dinucleotide (here the “p” in CpG indicates the phosphate group between C and G residues in the same strand).

1.4. DNA sequencing technologies and historical background

Formally, the term *DNA sequencing* defines the manual or automated process to determine the precise order of **nucleotides** in a **DNA molecule**. Beyond this basic outdated definition, sequencing technologies have been extended to **sequence** and **quantify** different RNA species, **DNA-protein binding and occupancy**, **chromatin accessibility**, **epigenetic chromatin modifications**, and DNA-specific modifications and alterations, including the DNA **methylation profile**, sequence **polymorphisms** and **mutations**, and huge **chromatin**

rearrangements, such as **gene fusions**. More recently, DNA sequencing technologies have been used to determine the **3D chromatin structure** and thus the **topological rearrangements** occurring in physio-pathological genome regulation processes, especially in **cancer** [16, 41].

Although the first DNA sequencing method dates to 1970, the so-called “sequencing revolution” began in 1977, with the publication of two milestone articles, one from **Allan Maxam and Walter Gilbert** [2], and the other from **Frederick Sanger** and colleagues [3]. Maxam and Gilbert presented a method in which terminally labeled DNA fragments were subjected to specific chemical cleavage and then separated by electrophoresis. Sanger’s group described the **chain-termination method**, employing nucleotide analogs to prematurely stop DNA synthesis and produce fragments separable by electrophoresis. The latter was more suitable for large-scale data production and commercialization. Sanger’s method was used to sequence the first human genome, through the ***Human Genome Project*** [42], and represented the gold standard for DNA sequencing in the past 30 years, until the advent of **Next-Generation Sequencing** (**NGS**) **technologies**.

Sanger-based sequencing technology is what we now call **First-Generation Sequencing** (FGS), characterized by high costs and low throughput. In this technique, DNA fragments are generated starting from the same location, but terminating at different template points. These differently sized fragments are distributed in the order of their length via **capillary electrophoresis**. The terminal residue of each fragment is marked with one of the four fluorescent dyes used, corresponding to a specific base. This information is used to reconstruct the original sequence. An automated Sanger platform may produce reads with an average length of ~800 bases, up to a maximum

of 1Mb. By contrast, NGS sequencing technologies are characterized by high throughput at relatively low costs (Tables 2 and 3). Traditional **NGS methods** (Figure 7), often referred as **second-generation sequencing** (SGS) techniques, use tens of thousands of identical DNA strands, produced by **PCR amplification**. Amplification details depend on the technology employed, however the two most common methods are **emulsion PCR** (emPCR) and **solid-phase amplification** (Figure 7). DNA strands are first annealed with primers to start the DNA synthesis needed for sequencing. These technologies use marked nucleotides that allow the machine to individuate which bases have been incorporated at each cycle (**base calling**). When a nucleotide is incorporated in the new strand, the support is washed and scanned, in a process called sample **imaging**. After imaging, dyes are cleaved and then the support is exposed again to marked nucleotides for another cycle. These **washing-scanning cycles** are repeated until the reaction is practicable. Due to the huge number of cycles required, the time elapsed per run could range from hours to days (Table 2). A common problem in SGS technologies is represented by **dephasing**, a phenomenon that occurs when growing primers become asynchronous for any given cycle. Dephasing increases signal noise, causing base-calling errors as the read extends, and thus limiting the read length produced by most SGS platforms to less than the average read lengths achieved by Sanger method [3]. Traditionally, we distinguish SGS reads in **short reads** ($\leq$ 200 bases/read) and **long reads** (> 200 bases/read) – values per platform are reported in Table 2. These technological details influence every aspect of the study: the experimental settings and design, the statistical analyses used to pre-process data, and the algorithms used for further data processing. Dephasing is not the only length-related issue. A general quality decay near read end (3' terminus) is caused by a set of circumstances,

including reduced polymerase efficiency, loss of polymerase and incomplete dye removal. Furthermore, PCR has intrinsic pitfalls, considering the relatively high amount of initial DNA needed, the polymerase reading errors, and the biases due to GC-rich and low-complexity regions in the template. Currently, the paradigm of SGS technology is represented by **pyrosequencing** (the first NGS technology; purchased by Roche / Life Sciences), **ion semiconductor sequencing** (by Ion Torrent / Life Sciences), **reversible terminator sequencing** (by Illumina/Solexa) and **oligonucleotide probe ligation sequencing** (by AB SOLiD). Since the imaging systems these platforms use, they are not capable of detecting single fluorescent events and template amplification is needed to increase the signal. While the mentioned SGS technologies are commutated by the PCR amplification step, they differ by the way in which PCR is performed.

In the framework of SGS, ion semiconductor sequencing (Ion Torrent) represent a step ahead respect to the other technologies. In this technology, DNA sequencing is based on the detection of electric field variations, caused by the production of hydrogen ions during DNA synthesis. DNA synthesis is realized in high density micro-wells arrays, endowed with an ion sensor, that eliminates the need of reagents for producing light signals, scanners and cameras to monitor the synthesis process, thus accelerating the whole sequencing process and considerably lowering costs. Nevertheless, Ion Torrent protocol still needs PCR (emPCR), washing-scanning cycles and thus halting the synthesis process between a cycle and the following one, to monitor the incorporated bases. So, despite of time and cost reductions, read length and throughput are comparable to the other SGS technologies (Table 2). **Single molecule sequencing** (SMS) is the key for **third-generation sequencing** (TGS) technologies, to avoid PCR pitfalls. While discussing about TGS technologies, we must consider that they have

pitfalls as well (first, the errors due to the weak signal coming from a single DNA molecule), that most of them have not been released yet and finally new and more accurate SGS platforms are currently in progress. Consequently, the name TGS should not be connected necessarily to better overall platform performances, respect to SGS. Anyway, there is neither a neat boundary between SGS and TGS technologies nor a perfect accordance amongst authors about the denotation of TGS. Here we will follow the definition given by Schadt et al. (2010) [11]: TGS platforms are those performing SMS and avoiding system halt while reading signals. Nonetheless, given this definition, there is still a technology that stays in between: the Helicos platform. It has been the first NGS platform capable of SMS, although it still relies on fluorescent dyes and so the washing-scanning step. The innovation introduced by Helicos consisted in avoiding PCR and the possibility of carrying out sequencing reactions asynchronously (a typical feature of TGS platforms). More recent technologies aim to monitor DNA polymerase activity in real time. The main problem in observing a DNA polymerase in real time is to detect the incorporation of a single nucleotide in a large pool of other potential nucleotides. Pacific Biosciences addressed this problem using **zero-mode waveguide** (ZMW) technology [11, 12]. This technology had been previously used in microwave ovens doors, that are covered by a holed screen, whose holes (i.e. the ZWMs) have a diameter of tens of nanometers. ZMWs allow shorter light waves to pass, but prevent larger microwaves to pass as well. In **single-molecule real time** (SMRT) sequencing methods, employed by PacBio RS platform, ZMWs are used to filter laser light. Within each ZMW, a single DNA polymerase is anchored on the bottom surface of the glass support. Nucleotides (each type labeled with a different dye) are flooded above the ZMW array. As no laser light penetrates up through the holes to excite the

fluorescent labels, the labeled nucleotides above the ZMWs do not contribute to the measured signals. Once captured by the polymerase, the dye emits colored light and the instrument can detect and associate it to a specific base. After incorporation, the signal returns to baseline (the background noise level) and the process can restart.

Table 2. NGS platforms: costs and throughputs (§). *Technology color code: first-generation sequencing in green, second-generation sequencing in blue and third generation sequencing in purple. Single-pass error rates are reported in percent (in round brackets the final error percent) and error type is reported in square brackets: [S] substitutions, [D] deletions, [ID] indels, [GC] GC deletions, [AT] A-T bias. Reagent costs are in US dollars (in round brackets the cost per run).*

Instrument	Run time (**)	Avg.Bases/read	Error rate %	ReagentCost/Mb
AB Sanger 3730xl	2h	650	0.1-1 (0.1-1) [S]	2308 (144)
454 GS Jr. Titanium	10h	400	1 (1) [ID]	22 (1,100)
454 FLX Titanium	10h	400	1 (1) [ID]	12 (6,200)
454 FLX +	20h	650	1 (1) [ID]	7 (6,200)
Ion Torrent 314 chip	2.5h	200	~1 (~1) [ID]	7 (350)
Ion Torrent 316 chip	3h	200	~1 (~1) [ID]	2 (550)
Ion Torrent 318 chip	4.5h	200	~1 (~1) [ID]	1 (750)
*Ion Torrent Proton I	4h	200	~1 (~1) [ID]	0.09 (1,000)
*Ion Torrent Proton II	4h	400	~1 (~1) [ID]	0.01 (1,000) (#)
Illumina MiSeq	26h	150 + 150	~0.1 (~0.1) [S]	1 (1,160)
Illumina iScanSQ	8.5d	100 + 100	~0.1 (~0.1) [S]	0.09 (12,750)
Illumina GAIIx	14d	150 + 150	~0.1 (~0.1) [S]	0.19 (17,575)
Illumina HiSeq 1000	8.5d	100 + 100	~0.1 (~0.1) [S]	0.04 (10,220)
Illumina HiSeq 2000	11.5d	100 + 100	~0.1 (~0.1) [S]	≥0.04 (23,470)
*Illumina HiSeq 2500	40h	150 + 150	~0.1 (~0.1) [S]	? (data not available)
SOLiD 5500xl	8d	75 + 35	~5 (≤0.1) [AT]	0.07 (10,503)
PacBio RS	2h	860 to 1,500	~15 (≤15) [GC]	7-38 (300-1,700)
*‡ O. N. Minlon	6h	9,000	≥4 (4) [D]	1 (≤900)
*‡ O. N. Gridlon 2000	?	10,000	≥4 (4) [D]	0.03-0.04 (varies)
*‡ O. N. Gridlon 8000	5h	10,000	≥4 (4) [D]	0.01-0.02 (varies)

‡ Oxford Nanopore

(*) Not released yet

(#) Extrapolated by author speculation

(**) Run time at maximum read length

(§) T.C. Glenn, 2011; Reference [10].

Table 3. Selection (in alphabetic order) of open-source/free read alignment algorithms. *Different algorithms are used for reference-based mapping and de novo assembly, but in general, read alignment algorithms can be roughly divided into those suited for long (> 200 bases) and short (≤ 200 bases) reads. Other general features are indicated as follows: [M] allow multi-threading, [P] allow paired-end data, [gap] allow gapped alignments, [SAM] SAM-friendly output, [G] suited for genome assembly, [LG] suited for large genome assembly, [EST] suited for ESTs assembly. Platforms are reported with the following nomenclature: [C] Sanger (capillary), [454] Roche 454, [T] Ion Torrent, [I] Illumina/Solexa, [S] SOLiD, [PB] Pacific Biosciences. Many long reads aligners were ideated before Ion Torrent or PacBio release, but Sanger- and 454-suited algorithms should work also for these platforms.*

Reference-based	Read type [options]	Platform	License	URL
BLAST	Long	(*)(**)	Public domain	[U4]
BLAT	Long [M][SAM]‡	(*)	Free for academic use	[U5]
BFAST	Short [M]	[I][S]	Open-source	[U6]
Bowtie	Short [M][SAM]	[I]	Open-source	[U7]
BWA	All [M][P][gap][SAM]	[I][S][454]	Open-source	[U8]
Maq	Short [SAM]	[I][S]	Open-source	[U9]
MrFAST	Short [P][gap][SAM]	[I]	Open-source	[U10]
SHRiMP	Short [M][P][gap]	[I][S]	Open-source	[U11]
SMALT	All [M][P][gap][SAM]	[I][C][454][T][PB]	Open-source	[U12]
SOAP	Short [M][gap]	[I]	Open-source	[U13]
SSAHA2	All [SAM]	[I][C][454]	Free for academic use	[U14]
***De novo* assembly**	**Read type [options]**	**Platform**	**License**	**URL**
ABySS	Short [LG]	[I][S]	Free for academic use	[U15]
AMOS	Long [G]	[C][454]	Open-source	[U16]
MIRA	All [G][EST]	[I][C][454]	Open-source	[U17]
SHARCGS	Short [G]	[I]	Open-source	[U18]
SOAPdenovo	Short [G]	[I]	Open-source	[U19]
SSAKE	Short [G]	[I]	Open-source	[U20]
VCAKE	Short [G]	[I]	Open-source	[U21]
Velvet	All [G]	[I][S][C][454]	Open-source	[U22]

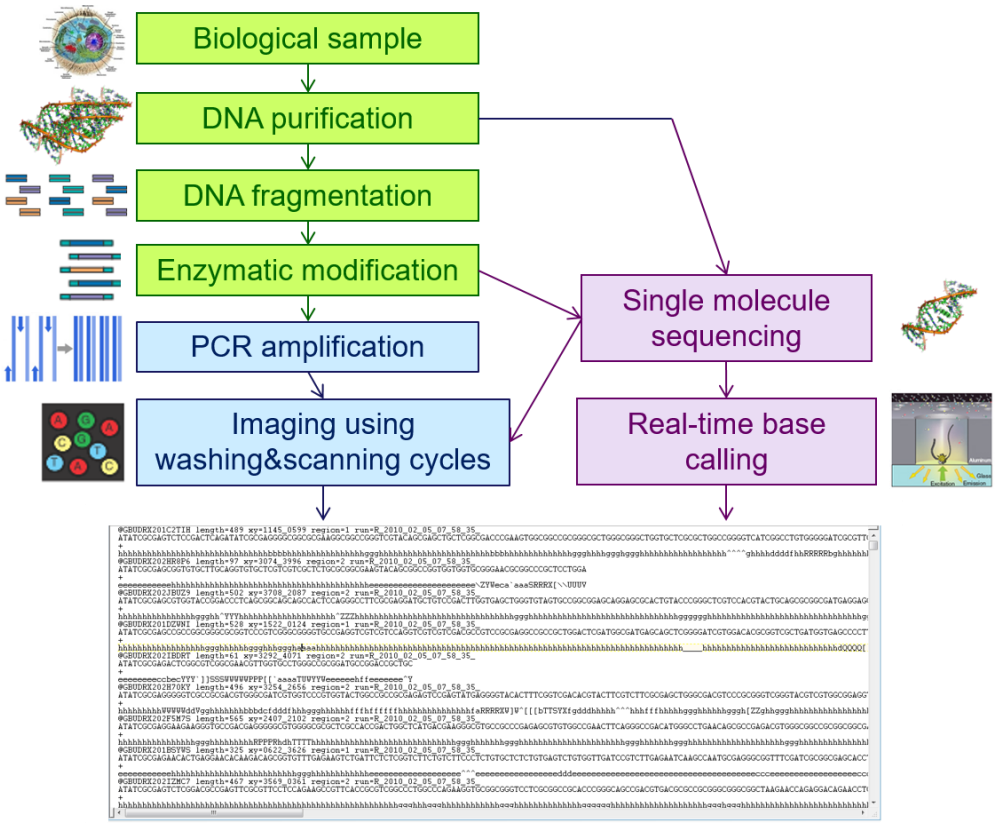
(*) Not designed for any particular platform (**) Uses a database rather than a reference sequence

‡ PSL to SAM format conversion could raise issues, e.g. due to CIGAR string absence in PSL

Due the incorporation event, mediated by the polymerase, the process employs milliseconds, that is about three orders of magnitude longer than simple diffusion. This difference in time allow a higher signal intensity and a consequent huge signal/noise ratio [11, 12]. In SMRT sequencing, dye is anchored to the γ-phosphate, hence its removal is part of the natural synthesis process. Due to the minimal amount of DNA required, SMRT sequencing do not need PCR amplification. This technique has other interesting qualities, such as the possibility of collecting kinetic information about polymerase activity. **Nanopore sequencing** is one of the most advanced sequencing technologies.

Nanopores can be either natural or synthetic pores, with internal diameter of 1nm. At this scale, DNA could be forced to pass inside the pore one base at time. Base detection can be realized measuring the specific effect that each different base type has either on an electric current or an optical signal, while passing through the pore. Oxford Nanopore Technologies will soon commercialize a system for DNA sequencing based on an engineered type of alpha hemolysin (αHL), found in nature as a bacterial nanopore. A synthetic cyclodextrin sensor is covalently attached to the inside surface of the nanopore and the system is included into a synthetic lipid bilayer. Single nucleotides are thus detected during cleavage from the DNA strand on the base of the characteristic ionic current disturbance they cause through the pore. A reliable throughput at high multiplex could be difficult to obtain using natural lipid bilayers, although synthetic membranes, together with solid-state nanopores, could help to overcome this problem in the future [11, 12]. Currently, other nanopore approaches are being studied. Briefly, one of them employs a modified version of *Mycobacterium smegmatis* Porin A (MspA), to perform sequencing directly on intact DNA, by measuring the effect of a single strand DNA (ssDNA) on the electric current through the pore [11]. The first parallel nanopore-based with optical readout has been recently developed, using solid state nanopores and fluorescence [12].

Figure 7. Generic experimental steps in NGS technologies. *Several biological sample preparation steps precede base calling and data (pre)processing for NGS. These steps can be grouped into: (i) library preparation (green boxes); (ii) washing-scanning cycles, typical of second-generation sequencing (cyan boxes); (iii) single-molecule sequencing, typical of third-generation sequencing (purple boxes).*



1.5. Sequencing standards and bio-ontologies

The primary source of inheritable information is the DNA sequence, although several inheritable epigenetic factors, affect genome architecture, regulation, and expression. This primary information is serialized in the simplest possible form into FASTA format files [U1]. FASTA are plain text format files characterized by a header line followed by the nucleotide or amino acid sequence. By convention, each sequence line should not be longer than 70 characters, to prevent formatting error in any kind of text editor. The header line should start with a > character and can be filled with free-text descriptions,

although often a pipe character (i.e. |) is used text separator. Multiple headers, followed by the respective sequence, can be concatenated inside a single FASTA file (i.e. “multiple FASTA file”). This is usually the input for multiple alignment algorithms (e.g. ClustalW, MAFFT, MUSCLE, T-Coffee). Since experimental and base calling procedures may introduce sequence errors, reads produced by sequencing machines are stored together with base call quality information. To assess read quality, FASTA files containing reads are accompanied with a QUAL file, containing the quality **Phred scores**. Phred quality scores were initially used in the Human Genome Project (Table 4) and assigned to assess quality during base calling, using the Phred program [43, 44]. The quality score *q* is connected to the probability *P* for a base to be miscalled, in the following way:

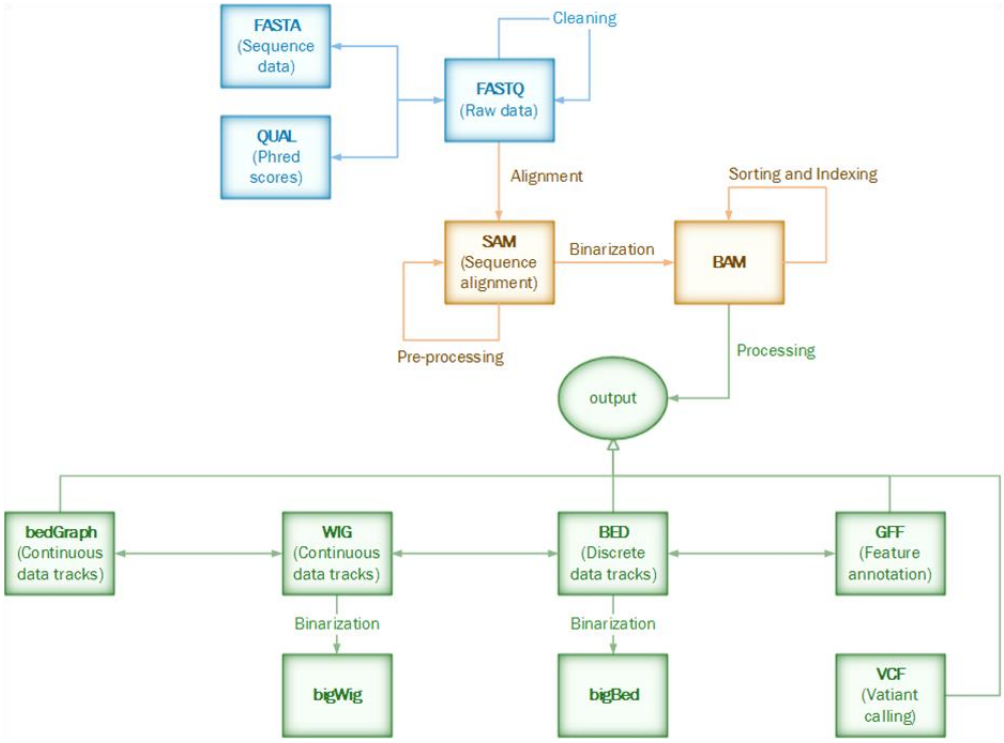
$$q = -10\log P \qquad \text{or alternatively} \qquad P = 10^{-\frac{q}{10}}$$

In early Solexa instruments, quality scores were calculated using the odds-ratio $P/(1 - P)$, but the current standard employs *P*. In this way, if a base comes with a Phred score of 20, it has a probability of 0.01 to be miscalled. A quality score of 20 is a standard threshold for many tasks in NGS analysis). From now on, the nomenclature Q*n* (e.g. Q20) will be used to designate a quality threshold of *n*. A QUAL file is the same as a FASTA file, with the difference that, in the former, base codes (A, G, C, T) are replaced by a sequence of quality scores, separated by a space. This implies that a quality file is much larger than the correspondent FASTA file. The FASTQ [U2] format was developed by the Sanger Institute to solve the problem of having two different files, by encoding the quality information in a single ASCII string, together with the nucleotide sequence. Every read sequence in the FASTQ file starts with a @, equivalent to the > symbol of a FASTA file, reserved to the header. The second line is the nucleotide sequence.

The third line starts with a + followed by either the sequence header or nothing, and the fourth line is the ASCII quality string. Each ASCII character correspond to a nucleotide. Therefore, a single read in a FASTQ file is represented by 4 lines. A FASTQ file can be generated starting from FASTA and QUAL files and vice versa, but different instruments could use diverse ASCII encoding. The current universal standard is the Sanger one, that use ASCII characters from 33 to 126 (! to ~) to encode quality scores from 0 to 93. So a quality score of 20 is encoded by the ASCII(20 + 33) = 5. Although newer Illumina standards (1.8+) use the normal Sanger encoding, previous Illumina instruments used own encodings. The **Sequence Read Archive** (SRA) [45] is a large repository of publicly available (compressed) FASTQ files, supported by the NCBI, from high-throughput sequencing platforms, including Roche 454 GS System, Illumina Genome Analyzer, Applied Biosystems SOLiD System, Helicos Heliscope, Complete Genomics, and Pacific Biosciences SMRT. Raw (i.e. unaligned) reads in FASTQ files can be then aligned either to profile the biological signal onto the reference genome or to build a new reference. A variety of read alignment algorithms are available, most of which are free software (Table 3). Standard file formats and the analysis flow connecting them is shown Figure 8. Before alignment, FASTQ files are cleaned to avoid alignment errors. Cleaning procedures may involve different steps, including file format conversion (for instance, 454 and Ion Torrent platforms produce SFF files), removal of adapter or multiplexing sequences, trimming low-quality bases from read end, and discarding low-quality reads (i.e. reads having too many low-quality bases). All these steps can be performed with the open-source FASTX toolkit [46]. Read alignment algorithms can be divided in two groups: **reference-based mapping** algorithms, and ***de novo* assemblers**. Reference-based algorithms can align reads against an existent reference genome, while *de novo*

assemblers merge different reads into a contiguous sequence, in absence of a reference guide. As one may expect, *de novo* assembly is more computationally intensive than reference-based mapping, with a computational complexity that increases as reads length and number decrease. Considering this, one of the strong points of TGS platforms is to produce longer reads than SGS ones: up to 1,5Mb for PacBio RS platform and 9-10Mb for Oxford Nanopore technologies. However, SGS technologies still have a major impact on data production respect to TGS, specifically on health and cancer research [6, 17-19], being used on a wide range of whole-genome experiments, from microbes to humans. Traditionally, 454 platforms have been favored by the capacity of producing longer reads respect to Illumina and SOLiD platforms, facilitating *de novo* assembly. This capability has been now implemented also by Ion Torrent, challenging the use of 454, mainly due to its lowered costs.

Figure 8. Schematic Input/Output data processing stream for NGS. *NGS data manipulation can be divided into three main blocks: (i) data pre-processing (cyan boxes); alignment (brown boxes); and (iii) post-processing (green boxes). Each block encompasses own standards and algorithms, as described in the main text.*



Although capable of producing longer reads, 454 and Ion Torrent platforms suffer the same limitation: high error rates when dealing with long **homopolymers**. A homopolymer is a stretch of DNA composed by the same base repeated a number of times. In both 454 and Ion Torrent platforms, the solution containing the four nucleotides flows over the support at each cycle, with the only difference that Ion Torrent does not use any marked or modified nucleotide. In both cases, stretches of sequence composed by the same base (e.g. AAAAA) are read as a single signal, deriving from the summation of the contributes of all the bases. Therefore, homopolymer length is calculated as a function of signal intensity. Following this reasoning, for example, the homopolymer AAAA should have a 4-fold higher signal intensity than a single base A. This rule keeps with few errors until we stay under a length of 6 bases. However, above this threshold, there is more uncertainty about the exact homopolymer length, increasing the probability of indel errors in the homopolymer flanking regions [5, 6]. For example, 454 platforms have a 3.3% error rate for 5-bases homopolymers. For 8-beses homopolymers , the error rate increase to 50%. Instead, in Illumina and SOLiD platforms, the homopolymer issue is solved through reversible terminators and two-base encoding. Illumina platforms use modified fluorescent bases capped at 3’-OH allowing only one incorporation event for each cycle. In this way homopolymers are read by position, instead of signal intensity, without a relevant increasing in error rate. This 454 and Ion Torrent weak point is the reason why short reads are often preferred and used also for *de novo* assembly [53]. Furthermore, the possibility of producing paired-end libraries (possible in all SGS technologies) can compensate for short read lengths and offer the possibility to observe large genome rearrangements (i.e. discovering structural variants).

This is possible because, in paired-end sequencing, each template strand is sequenced at both ends. Suppose that the template DNA fragment is 200 bases long. If the two mates of paired-end sequencing maps against the reference genome at a distance larger than 200 bases, we are in presence of a deletion in our sample genome. Vice versa, if the distance is shorter than 200 bases, we will notice an insertion in the sample genome. A similar reasoning can be done for long repetitive regions, that can be unmappable with a single read. In fact, if one read in a pair maps in a non-repetitive region, its mate should map at the distance of 200 bases. Therefore, this information can be used to map a paired read onto a highly repetitive region. Whole genome sequencing allows us to have a “soft copy” of the cell hereditary information. Indeed, only clones share the same genomic DNA sequence. Natural or induced DNA modifications and the environmental effect will shape DNA sequence in a unique way. Furthermore, technological limits prevent us to know the whole genomic sequence of a species. However, these obscure genomic portions (i.e. gaps) are gradually revealed technological evolution, generating updated versions of an organism’s genome, called **assembly**. Therefore, a genome assembly has a specific code indicating the source organism and the release. For instance, genome assembly GRCh38/hg38 is the Homo sapiens reference genome released in December 2013. Whole genome sequencing has the advantage of providing huge information about rare variants and structural variation present in an individual genome, highlighting the genomic basis for several diseases; especially in cancer research [17-19]. Anyway, at present time, whole genome sequencing experiments could be still too expensive leaving as a valid alternative the targeted sequencing of a limited genome region. This procedure is currently being used for the analysis of whole exomes, specific gene families involved in drug

response or genomic regions implicated in diseases or pharmacogenetics effects through GWAS. Targeting is a common practice in PCR, but it would lead to unpractical solutions, as thousands of different primers, individually or multiplexed, would be required for a single instrument [9]. Overlapping long-range PCR amplicons could be used for regions in the order of hundreds of kbp, although this method is not suitable for larger genomic regions. Alternative solutions, based on oligonucleotide microarrays and solution-based hybridization strategies have also been used for targeting genomic regions. For example, Roche/NimbleGen oligonucleotide microarrays have been employed to provide complementary sequences for solid phase hybridization to enrich for exons or contiguous stretches of genomic regions [47, 48]. Other researches employed techniques to capture specific genomic regions by using solution-based hybridization methods, such as molecular inversion probes (MIPs) and biotinylated RNA capture sequences [49].

Algorithmically, there are two major groups of read aligners: (i) based on **hash tables**, including MAQ, BFAST and SMALT, and those based on **Burrow-Wheeler Transform** (BWT), such as BWA, Bowtie and SOAP2. In hash table-based methods, the reference (or less often reads) is **indexed** in short multiple spaced words of fixed length. Firstly, exact matches between reads and reference are identified by searching specific ***k*-mers** of the read in the reference hash index. This process produces candidate alignment regions (i.e. **seeds**), that are used to produce **pairwise alignments** between the read and the reference, using the **Smith**-**Waterman algorithm** [50]. In BWT-based methods instead, a transform of the reference sequence is created. The transform contains the same information as the original reference, and so can be converted back to it, but is arranged in a way that allows fast subsequences lookup. BWT offers a very efficient data compression,

especially for repetitive substrings, allowing much higher speeds than hash-based methods. The cost for this speed is represented by a lower mapping accuracy, although BWT-based methods are generally more sensitive in variable regions. In hash-based alignment, two parameters are crucial: the word length (the stretch of identical nucleotides between query read and reference needed to define a candidate alignment region) and the step size, that is the space between a word and the next one (e.g. with a step of 3, every three words in the reference, only one is kept). Both parameters can be set manually, but there is the possibility to tune them on the base of read lengths, source organism and library type (i.e. single-end or paired-end). Generally, short reads ($\leq$ 100 bases) require a small step size (2-3), otherwise one could obtain reduced sensitivity and too many mapping errors. For long reads (> 100 bases) and/or paired-end data, step size can be increased (up to the word length default value, i.e. 13). The choice of word length is less critical, and it can be generally increased from its default value (i.e. 13) only for long ($\geq$ 100 bases) Illumina reads (remember that long illumina reads suffer less error rates than 454 or Ion Torrent reads). Also the reference genome influence parameters tuning. For example, small bacterial genomes require less memory, so they can be indexed at minimum step (= 1), aiming to maximum accuracy. Logically, all these parameters should consider memory requirements. The conventional alignment format is called **SAM** (Sequence Alignment/Map format) [51]. This format is now the standard for many aligners, although there is still some exception, such as BLAT that produces alignments in PSL format. Converting from PSL is possible, but could cause problems due to the lack of information (e.g. the CIGAR string) in the PSL format.

All these alignment-specific statistics can be computed using **Samtools** [51]. An important quality parameter is **coverage depth**. Coverage depth is the number of mapping reads at a given alignment position. During alignment, it is possible to have **duplicated reads**, due to PCR amplification, resulting into artificially increased coverage. In Illumina reads there are also optical duplicates, caused by a single cluster being read twice (optical duplicates can be detected because they will appear very close in the slide). The problem of a low of coverage can be only balanced upstream, by producing high quality reads (this will reduce the number of unmapped reads). Monitoring coverage distribution and using uniquely mapped reads solve this problem, unless there is sufficient coverage depth. This problem can badly influence variant calling by weighting more some regions respect to others and could be even more severe in quantitative experiments. However, duplicate removal efficiency depends strongly on the library quality, since even PCR duplicates can have small sequence differences. On the other hand, reads mapping at the same coordinates could still come from different templates, and the user must be aware that duplicate removal could result in a massive loss of data. For instance, when we are interested in single-cell signal (i.e. reads from every clonal population are marked with a specific tag), duplicate removal effectively prevents cell-specific quantification. Consequently, even if there is no straightforward method to perform duplicate removal, the user should adopt some caution. For example, coverage profiling may help in spotting signal spikes (i.e. read distribution outliers) and remove them. Another key issue raised by variant calling is **realignment**. This is needed because the mapping process is a pairwise, instead of multiple, alignment between each read and the reference, implying that different positions could come with high entropy (i.e. could be wrongly aligned). The problem,

involving especially alignment gaps, is that SNPs and indels called in misaligned regions are not reliable. Solving this problem by performing multiple alignment is still computationally too intensive, therefore many algorithms adopt local realignment. This strategy is implemented in currently available variant calling software, including Samtools [51] and GATK [87]. After realignment, bases having suspect positions in the alignment must be recalibrated, to avoid calling SNPs from them. Recalibration consist in re-computing mapping quality after realignment.

Reads aligned onto the reference generate a “tag” (i.e. aligned reads) distribution profile. Density profiles can be also used to discretize biological signals into **tag density regions**, called **peaks** (see Paragraph 3.5.2). Discrete signals are serialized in **BED** files. BED files have 6 compulsory fields for feature definition: *chromosome*, *start*, *end*, *name*, *score*, *strand*; and other 6 fields for visualization. These can be followed by any other supplementary field. The ENCODE project use a modified BED version for either broad-source or point-source peaks (corresponding to different signal profiles, see Paragraph 3.5.2), called **broadPeak** and **narrowPeak**, respectively. The broadPeak format has the canonical 6 first fields, followed by *signal value*, *p-value*, and *q-value* fields. The narrowPeak format is equal to broadeak, plus a supplementary *summit* field. These two files format are used to represent regions in which tag density is significantly higher than the background noise (see Paragraph 3.5.2). However, in many cases, it turns to be more convenient working with continuous data (in this case “continuous” means “per-base”), given the number of true biological signals very close to statistical noise. Continuous signal distributions can be serialized into *ad hoc* file formats, including **wiggle**, **bigwig**, and **bedgraph**.

Alignment-related issues make genome alignment a computationally intensive task. The state-of-the-art in genome computing offers a wide range of resources to speed up analyses. Different whole genome sequencing projects took advantage from large scale parallelization using huge hardware facilities [52]. For instance, Jeckson et al. assembled *Drosophila melanogaster* genome on a 512-node BlueGene/L supercomputer in less than 4 hours, starting from simulated short reads [53]. Despite the excellent results obtained, large-scale parallelization is not always applicable, due to the high hardware resources needed and the low reusability of the software designed for a specific cluster. As an alternative, an increasing number of web services is offering **cloud-computing** resources for genome analyses [54-61, 87]. The advantage of using a distributed system is that the user could chunk the input file (e.g. a reference genome) in partitions, and perform a job (e.g. reads mapping) against the partitioned data. Many distributed systems are based on **Hadoop** [56, 57], an open source implementation of the programming model **MapReduce** [55], used to process large datasets. In Hadoop, each job instance is almost independent and workload deployment in highly efficient, allowing a quasi-linear wall-clock time reduction with the number of CPUs employed. Several distributed web services use Hadoop, making it almost a standard in cloud computing. For example, Amazon provide preconfigured disk images and scripts to initialize Hadoop clusters. To have an idea of the potential advantage of this approach, the Crossbow platform resequenced the whole genome of a Han Chinese male at 38X coverage in 3 hours, using a 320-core cluster provided by Amazon EC2 system at the cost of $85 [61].

The sequencing costs drop, together with technological and methodological advancements, made available large datasets, allowing data integration and knowledge discovery. These large datasets,

including also non-sequencing biological data (Table 4), can be integrated with structural annotations, including transcripts, genes, enhancers, transcription factor binding sites, protein complexes, histone modifications, open chromatin regions, CpG islands, repeats, common and disease-associated variants, CNVs, and others. These features can be retrieved from large databases, knowledgebases, and genome browsers, and serialized in **BED**, **GTF**, **GFF**, or **VCF** formats. **GTF** (General Transfer Format), otherwise called **GFF2** (General Feature Format 2), is a tab-separated file format developed as a GMOD [62] standard, composed by 9 fields: *chromosome identifier* (called *seqid*), *source* (database or project name), *type* (type name, e.g. transcript, gene, variation, etc.), *start*, *end*, *score* (i.e. a floating point number), *strand* (+, -, or . for unstranded features), *phase* (0, 1, or 2, depending on the reading frame), and a semicolon-separated list of *attribute-value pairs* (i.e. attribute1=value1, attribute2=value2, …). GTF/GFF2 is still largely used although deprecated in the GMOD standard that encourages the use of the **GFF3** format. This new version allows to model **nested features** (i.e. parent-descendant relationships among features), and **discontinuous features** (i.e. features composed by multiple genomic segments). These new components in the GFF3 standards allow to model complex features, including coding genes with all their components (i.e. UTRs, CDS, introns, exons, and alternative transcripts), alignments, and quantitative data (e.g. peaks, RNA-seq, or microarrays). Genetic variants are annotated using their own format: **VCF** (Variant Call Format), designed as the standard for the 1000 Genome Project [63]. Every VCF contains both meta-data, encoded by lines starting with ##, and data, in 8 tab-separated fields: *chromosome identifier*, *position* (corresponding to the start in a BED file), *ID* (variant identifier, that sould be the **dbSNP** [64] identifier), *alt* (i.e. comma separated list of alternative, non-reference, alleles), *qual*

(Phred quality score for the probability that the alternative called allele is wrong), *filter* (equal to PASS if the variant passed quality check), *info*. The last VCF field contains additional information about variants, encoded as *attribute*=*value* pairs. Although it is possible to define any attribute, some of them are reserved. For instance, AA stands for *ancestral allele*, AC is the *allele count* in the genotype, AF is the allele frequency, CIGAR is the CIGAR string in a SAM alignment (see SAM format specification [51]), SOMATIC indicates if the variant is a somatic mutation (used in cancer genomics), VALIDATED if the variant has been validated in a follow-up experiment, and so on.

Every standard listed here has been developed by a specific institution (e.g. NCBI), database (e.g. SRA [45], GEO [65, 66], the UCSC Genome Browser [67]), or project (e.g. the 1000 Genome Project [63], ENCODE [35, 36]). Table 4 shows a list of institutions and projects adopting these standards. However, there have been efforts to create independent and general standards for sequencing data format, annotation, and modeling. We already pointed out the **GMOD** (Generic Model Organism Database) project, that is a collection of open-source software tools for managing, visualizing and storing genomic data [62]. This project promotes GFF/GTF file formats, as well as standards for genomic data modeling, that are currently employed by several widespread databases, genome browsers, and analysis platforms, including **BioMart** [68, 69] and **Galaxy** [58]. These standards however do not cover basic experimental protocol specifications to be reported, or controlled vocabularies for generic metadata (including reagents, experimental conditions, statistical methods, and algorithms), clinical and phenotype data description. For this purpose, the MGED Society [70] developed the **MINSEQE** [71], that is the minimum required information about a high-throughput sequencing experiment. MINSEQE standard is the analogous of MIAME standard

for microarray data, a methodology that has been almost completely replaced by sequencing technologies. NCBI GEO and EBI ArrayExpress are finalizing their compliance with MINSEQE. Furthermore, structured genetic and genomic annotations are almost everywhere implemented as domain-specific **bio**-**ontologies**. A bio-ontology has generally a **directed acyclic graph** (DAG) structure, that is often subdivided into multiple roots. **Functional annotations** are collected in the **Gene Ontology** [72], further subdivided into **molecular function**, **biological process**, and **cellular component** roots. Human diseases are instead encoded inside the **Disease Ontology** (DO) [73], whereas a generic controlled vocabulary for high-throughput sequencing is provided by the **Sequence Ontology** (SO) [74]. All these bio-ontologies are part of the **OBO** (Open Biomedical Ontology) **Foundry** [75].

The vast majority of publicly accessible biological information is still unstructured, especially from the clinical side. Molecular databases (e.g. **KEGG** [76], **Reactome** [77], **UniProt** [78] or **STRING** [79]), disease databases (e.g. **OMIM** [80]), cancer databases (e.g. **TCGA** [81], or **ICGC** [82]), and large genomics consortia (e.g. **ENCODE** [35, 36], **Roadmap Epignomics** [38], the **1000 Genome Project** [63], or the **HapMap Project** [83]) do not provide structured access to their data and metadata. Although many instruments provide user-friendly interfaces for interoperability (e.g. **Galaxy** [58], **BioMart** [68, 69], **Cytoscape** [84], **IGV** [85], or **IGB** [86]), they must account for multiple or even proprietary biological data/metadata licensing.

CHAPTER 2

Biology and Medicine as Big Data Sciences

And now there come both mist and snow,
And it grew wondrous cold:
And ice, mast-high, came floating by,
As green as emerald

Samuel Taylor Coleridge
The Rime of the Ancient Mariner, 1798

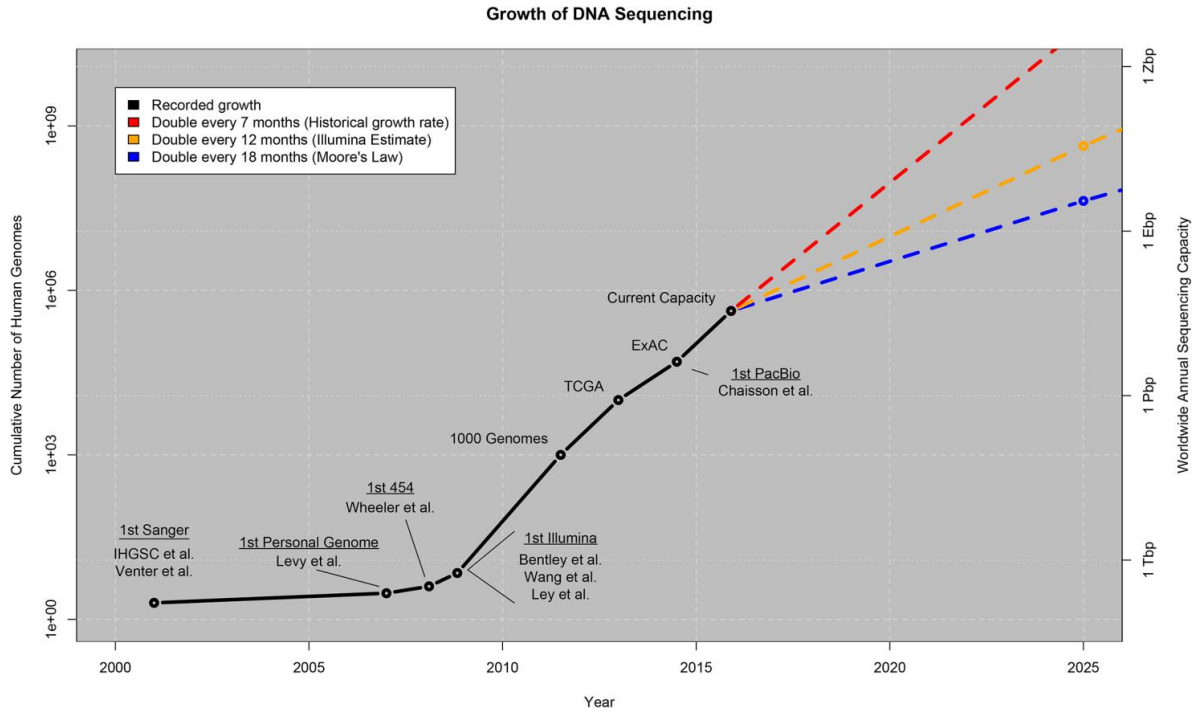
Data deluge is a phenomenon faced in many scientific and social fields, much before the advent of big genomic data. Astronomy and particle physics are typical examples of big-data sciences. More recently, social network websites and applications, such as YouTube and Twitter, have produced comparable amounts of data in a highly distributed way. However, scientific and social fields deeply differ in four crucial data management aspects: **acquisition**, **storage**, **distribution**, and **analysis**. The variable nature of the data is the key reason why this four-faced paradigm shifts accordingly to the specific field of application, making data management, modeling, and analysis the new challenges of bio-medical disciplines. Three fundamental research aspects have been raised as cornerstones for disclosing biological complexity through sequencing technologies: **experimental**, **technological**, and **methodological** advancements. **Sequencing flexibility** and **costs reduction** enhanced the creation of new protocols for many different

applications, from single-cell sequencing to quantification nascent transcript abundance and DNA damage in cancer. Collaterally, information technology developed new systems and interfaces to ease **data retrieval**, provide **parallel computing** solutions, and guarantee **interoperability** across different platforms to speed up turnaround times in data analysis and clinical environments. Finally, new analysis methods are continuously being generated to provide **efficient heuristic solutions** and modeling strategies to overcome the bottleneck between data production and knowledge discovery, in research, or classification and diagnosis, in medicine.

2.1. Data complexity and genomics as a hypothesis-driven science

Genomic data is intrinsically **complex** and **heterogeneous** [4, 6, 17-19, 88-90]. Recent estimations [88] quantified the increase in genomic data production to 1Zbp/year (where 1 Zbp = 1 zetta-base-pair = 10^{21} bp) by 2025, corresponding to 2 orders of magnitude increase in the next decade. To have a better idea of the current genomic data growth rate, this correspond to doubling the data amount every 7 months, compared to the expected growth rate (by Moore’s law) of doubling every 18 months (Figure 9).

Figure 9. Current NGS data growth rate and projections by 2025 (see reference 88).



At the present growth rate, the first technological bottleneck is represented by data storage. Currently, publicly available genomes from most published works are stored in the Sequence Read Archive (SRA) in the form of raw (i.e. unaligned) sequencing reads [45]. The archive is maintained by the United States National Institutes of Health – National Center for Biotechnology Information (NIH/NCBI), or one of the counterparts of the International Nucleotide Sequence Database Collaboration (INSDC; URL: http://www.insdc.org/). However, sequencing data in the SRA is only a small fraction of the total amount produced for research and clinical purposes. This scenario is further complicated by the possibility of sequencing **single-cell** and **cancer genomes** [6, 17-19]. It has been estimated that the required capacity for these data would be 2 to 40 exabytes (i.e. 10^{18} byte) by 2025. From the experimental side, "third-generation" **single molecule sequencing** technologies [11, 12] could help to reduce the quantity of DNA sample to be sequenced and stored; although this cannot be taken as a long-

term solution. Online catalogues of variants, such as the **EBI GWAS catalog** [39], the **Database of Genomic Variants** (DGV) [91], or the **NCBI dbVar** [92], and large shared open-access repositories, including **TCGA** [81], **ICGC** [82], and **NCBI GEO** [65, 66], may offer centralized data sources, avoiding redundant data storage and providing guidelines, protocols, and interfaces for data submission and retrieval.

These solutions together facilitate data storage and distribution, but the ultimate solution and the second, much larger, bottleneck is data analysis [88-90]. When data is big, analysis methods are not only mining tools, but they must consider **computational complexity**, the possibility of efficiently apply **compression**, **indexing**, and **data reduction** strategies, minimizing data loss, and the **biological validity** of models and results. The key data analysis problem is to cope with **biological complexity**, rather than data dimension. In biomedical research, biological complexity arises from different aspects. This is particularly evident when the subject is the **causal relationship** between **biological variation** (including **DNA sequence variation**, **gene expression**, **epigenetic regulation**, **protein structure and function**, and **metabolic reactions**) and **phenotype** (i.e. the set of an organism’s observable traits, including **morphology**, **molecular function**, **development**, **behaviour**, and **diseases**) [4, 6, 18, 19, 89]. To his end, two concepts gain a fundamental value: data **integration** and data **interaction** [4]. Data integration is the process by which different data types are combined to have a better description and comprehensive modeling of a phenotype. In a single-scale model, one single data type (e.g. gene expression) is used to describe a biological trait (e.g. the phenotypical differences between healthy and diseased groups of people). In **multi-dimensional modelling**, instead, different data types (see above) are combined according to a **multivariate** model, at different **genomic scales**. Here, the term genomic scale identifies the

entire set of **genomic elements** (e.g. single nucleotides, DNA sequence motifs, genes, enhancers, topological domains, up to entire chromosomes) and their possible **interactions**. Hence, data integration encompasses the concept of interaction, that is the fundament of biological complexity [90].

In a recent article [4], Ritchie *et al.* (2015) exposed two main working hypotheses in genetics and genomics. In the first one, a **multi-staged data analysis** approach is used, where models are built using only two data types at a time, in a stepwise manner. At each step, the output of one analysis is used as an input for the following one. This approach give rise to the so-called **pipelines**, a name deriving from UNIX-based scripting that employs the “pipe” (i.e. |) character to pass the output of one process as an input for the next one. Traditional pipelines are the result of a procedural approach that follows a simplistic and outdated paradigm of molecular biology, in which genetic variability (almost completely reduced to genes) is the only variable influencing gene expression (transcription) and thus protein structure and function (proteome), causing a specific phenotype. However, this linear working hypothesis do not explain the remarkable phenotypical variability observed in many complex diseases, including behavioral disorders and cancer.

In the alternative working hypothesis, complex phenotypical traits arise from the **simultaneous effect** of **genetic variability**, **epigenetic modifications**, **transcriptional levels**, and **proteomic variability**. In this case, both **model topology** and the direction of **causal relationships** is **not fixed** *a priori*. This implies that the ***causal relationships among variables must be tested, on the base of a model (under the null hypothesis of no significant interaction)***. Given this new paradigm, a sequential modelling strategy is no more appropriate, and **integrative causal models** provide a more efficient solution. Integration can be

ensured, for instance, using **network-based** modelling strategies. Networks are efficient representations since they can incorporate the concepts of: (a) **interaction**, where nodes (i.e. genomic elements) are connected through edges (i.e. their interactions); (b) **integration** (e.g. network connections can be weighed combining different data types); (c) **causality**, by testing or estimating effect intensity and/or directionality; and (d) **simultaneity**, i.e. testing multiple effects simultaneously. Network models also allow **time-dependency**, a critical aspect in system biology, where model interactions may change through time. Naturally, there is no single solution that guarantee the most accurate estimate or prediction, due to several aspects, including:

(i) **Model fitting** and proportion of **explained variability**. Both the input data and the model could be not sufficient to explain the **outcome variability** (e.g. the observed difference between a population of diseased people versus and healthy one), leading to a poor model fitting.

(ii) **Multi-collineaity**. Model variables (i.e. **predictors**) are redundant (highly correlated), leading to non-reliable parameter estimations.

(iii) **Overfitting**. When classification or prediction performs extremely well on the data used to build the model, but has poor accuracy with unrelated test data sets.

(iv) **Validation**. One single model is usually not enough to faithfully describe a complex phenotype. Multiple, integrated and orthogonal methodologies (i.e. **ensemble methods**) usually provide a more accurate result.

All these aspects, deeply connected to biological interaction complexity, together with possible parallel computing paradigms, will be discussed in chapter 3. As a general conclusion, the advent of big data in genomics did not simply enhance low-cost and massive data production, but started a technological and methodological revolution that disclosed a deeper understanding of complex biological phenomena, offering new possibilities for data sciences, personalized medicine, and basic research.

If cost and runtime reduction will keep the present trend, an increasing demand of personal genome sequencing, also in healthy people (e.g. for prevention purposes) will cause an unbelievable explosion of personal genomic data [88]. This is obviously a chance to collect large amounts of genomic information, potentially useful in medicine, human biology and evolution. Genomic variability accounts for large part of the observable phenotypical variation, including disease risk and treatment effectiveness. Genome sequencing and exome sequencing enhanced our understanding of the genetic bases of Mendelian (i.e. monogenic) diseases, providing precious insights in neurological disorders and cancer, including intellectual disability, autism, schizophrenia, and breast cancer [4-6, 14-19]. Routine characterization of complex (i.e. polygenic and environmental) and rare disorders in clinical practice is more difficult. One main limitation that is gradually being removed is represented by costs. The recent drop in sequencing and capital costs for NGS facilities, allows the sequencing of personal genomes. Several genome projects have been working in this direction for many years, such as the **1000 Genome project**, **The Cancer Genome Atlas** (**TCGA**), the **ENCODE project**, the **Roadmap Epigenome** project, the **UK10K project**, the **Personal Genome Project**, and many others (see Table 4 for references). In this framework, the development of databases and web-based open-

source analysis platforms provides a wide range of benefits in personalized medicine and public healthcare, and a solid base for a new translational approach that directly use genomic information to influence clinical decisions [17-19, 93]. Therefore, new algorithms, analysis software and web services are being created, responding to an increasing demand for computational power.

2.2. NGS technology perspectives

The fast-evolving landscape of HTS allowed researchers to identify a wide range of sequence polymorphisms and mutations, with relatively high accuracy at a sufficient coverage [4, 6]. Hence, HTS is gradually replacing less sensitive methodologies, including microarrays. HTS technology offers the possibility of detecting rare variants, Copy Number Variants (CNVs), gene fusions, and non-annotated transcripts [6]. In general, sequencing sciences, including evolutionary biology, molecular biology, medicine, and forensic sciences, are now facing the study of biological systems at almost any scale of complexity, including:

(i) Organism- and cell-specific **whole genome sequencing**.

(ii) Between and Within **population genetic** variability characterization.

(iii) Single-individual variability, involving:

a. **Epigenomics**

b. **Immunogenomics**

c. **Cancer genomics**

d. **Metagenomics**

(iv) **Single-cell genomics**, including **DNA-methylation**, **histone modification** and **DNA-Protein binding** profiling, genome **3D**

structure resolution, **transcriptional activity** quantification, transcript **post-processing**, and **DNA Replication**.

This new experimental paradigm revealed the actual complexity underlying biological variability. We are now able to describe the different regulatory states among different cell types (Single-cell sequencing), define the somatic variation at the basis of the immune response (Immunogenomics), discover genomic differences between normal and cancer cells and explaining their impact on cell functioning (Cancer Genomics), and understanding the genetic variability characterizing the human microbiome (Metagenomics). Besides DNA sequence, there is another source of biological variability: the **epigenome**. The epigenome includes *the set of chemical modifications of DNA and histones, that alter DNA structure and processing, without changing its nucleotide sequence*. These inheritable modifications may involve the DNA itself (e.g. DNA methylation), or the proteins responsible for DNA packing (i.e. the histones), thus changing chromatin accessibility, transcription factor binding, and then transcription and replication rates. If altered, the epigenetic modifications may affect also **DNA sequence** (mutation), **cell cycle** (transcription and replication timing), and the whole **genome stability** (chromatin damage and topological rearrangements).

An **NGS protocol** is defined as a sequence of specific wet lab operations to transform one or more **cell populations** defined by an **experimental design** into nucleic acids suitable for analysis at the molecular level by a sequencer [6]. The conventional approach consists

in isolating DNA or RNA, shearing the nucleic acid sample into fragments with either physical methods (e.g. sonic waves) or enzymes (e.g. DNase or MNase), and enzymatically adding adapter sequences to make them suitable for the specific sequencing platform. A collection of purified and processed fragments is called a **library**, and the **library size** is the total number of fragments for each library. Newer protocols include **targeting**. Targeting consist in selecting a subset of cellular DNA or RNA from the original sample. This approach may also include the isolation of a subpopulation of cells from the initial population. At various steps of library preparation, a sequence **tag** can be added to the fragments to recognize it during data analysis. Tagging can be used to distinguish different cell subpopulations or even single cells. Although sequencing technologies reached an enormous flexibility in few years, there is still ground for improvements and future developments [4, 6, 17-19, 88-90, 93]:

- **Genome assembly technologies and algorithms**. Many genomic regions remain unresolved, due to specific types of variations, repetitive regions, or nucleotide composition.
- ***In situ* single-cell sequencing**. In situ sequencing, may enable detailed study of mitochondrial and cancer genomes, and provide sub-cellular transcript quantification. This technology could also include **multiplexing**; in a similar way, it is done for individuals' genome. Furthermore, single-cell sequencing could be used to quantitatively assay several aspects of signaling cascades, for

instance involved in inflammation, immune response, metabolism, and cancer.

- **Biological interaction maps**. Given the **complexity of biological systems**, NGS technology are a powerful instrument to examine the source of such a complexity: biological and biochemical **interactions**. Potentially, sequencing sciences can study biological interactions at any level, including massive parallel discovery of **protein-protein interactions** (PPI), **neuronal connectivity** maps, and **developmental fate** maps, i.e. the massively-parallel assessment of cell lineage development of multicellular organisms.

Table 4. List of consortia and bioinformatics projects. *List of some important projects and databases (in chronological order) involved in genomics and metagenomics [G], epigenomics [E], proteomics [P], metabolomics [M], pharmacogenomics [F], human variability [V], healthcare and disease characterization [H].*

Project	Launced	URL	Field
RCSB – PDB database **	1971	http://www.rcsb.org/pdb/home/home.do	P
OMIM *	1987	http://www.omim.org/	H
Human Genome Project	1990	http://www.genome.gov/10001772	G
KEGG database	1995	http://www.genome.jp/kegg/	M
Gene Ontology	2000	http://geneontology.org/	G
GEO – NCBI database	2000	http://www.ncbi.nlm.nih.gov/geo/	GE
PharmGKB	2000	https://www.pharmgkb.org/	F
STRING	2000	http://string-db.org/	P
Chimpanzee Genome Project	2002	http://genome.wustl.edu/genomes/view/pan_troglodytes/	G
HapMap project	2002	http://hapmap.ncbi.nlm.nih.gov/	V
UniProt knowledgebase	2002	http://www.uniprot.org/	P
Disease Ontology	2003	http://www.disease-ontology.org/	H
ENCODE project	2003	http://www.genome.gov/10005107	G
Human Epigenome Project	2003	http://www.epigenome.org/	E
Reactome database	2003	http://www.reactome.org/	M
Sequence Ontology	2005	http://www.sequenceontology.org/	G
The Cancer Genome Atlas	2005	http://cancergenome.nih.gov/	VH
Human Variome Project	2006	http://www.humanvariomeproject.org/	V
1000 Genomes Project	2008	http://www.1000genomes.org/	G
GWAS catalog	2008	https://www.ebi.ac.uk/gwas/	VH
Human Microbiome project	2008	http://commonfund.nih.gov/hmp/	G
ICGC ***	2008	http://www.icgc.org/	VH
Roadmap Epigenomics	2008	http://www.roadmapepigenomics.org/	E
GeneMANIA	2010	http://genemania.org/	P
Neanderthal Genome Project	2010	http://www.eva.mpg.de/neandertal/	G
UK10K	2010	http://www.uk10k.org/	V

* OMIM = Online Mendelian Inheritance in Man; ** PDB = Protein Data Bank;
*** ICGC = International Cancer Genome Consortium

CHAPTER 3

Genomic Data Modelling

Thus strangely are our souls constructed,
And by such slight ligaments are we bound
to prosperity or ruin

Mary Wallstonecraft Shelley
Frankenstein, 1818

In the previous chapters, we faced the impressive growth rate of genomic data production, accelerated by the drop of sequencing costs, and the possibilities offered in biomedical research. Genomic data deluge disclosed the actual **complexity** of the human genome and accompanied the technological and methodological revolution that allowed researchers to use it to understand the molecular basis of human disorders. We understood how complex phenotypes are determined by a dense network of **heterogeneous interactions** between molecular components, that can be represented through **network models and dynamics**, key aspects of computational and systems biology. Genomic heterogeneity and complexity cannot be handled without efficient modelling and data reduction methods, given also the high inter- and intra-sample variability [4, 6, 17, 18, 88-90]. For this reason, genomics is a **hypothesis-driven science**, in which informative interactions are not given *a priori*, and must be tested at

each analysis step. In this scenario, data **quality assessment** and method **validation** become necessary tools. **Knowledge**, represented by controlled **annotations** provided by domain-specific **knowledgebases** and **ontologies**, may help in **model construction** and **missing data** integration, although it may hide intrinsic biases and/or not reflect the actual experimental variability. In the following paragraphs, we will discuss **data modelling**, **data reduction** and **analysis methods**, with the final goal of describing principles and methods for building a comprehensive **genomic interaction model** (GIM), capable of integrating several aspects of genomic data modelling and mining (Figures 10 and 13).

Figure 10. Schematic representation of the key concepts supporting the idea of Genomic Interaction Model. *From the theoretical point of view, conceptual models and data analysis methods tend to be separated. Typically, knowledge is used to build static representations of genomic features, as entities included in a network of fixed semantic relationships. On the other hand, data is represented as a black box from which to extract new information and, possibly, new knowledge. In Genomic Interaction Modeling, complex feature interactions are used as building blocks for (theoretical) model construction. Data integration, is used to assess the significance of genomic interactions, and the existence of new genomic features. In this context, data analysis is considered an instantiation of the theoretical model, in which data enters as an input, and the improved model (endowed with new knowledge) is the output.*

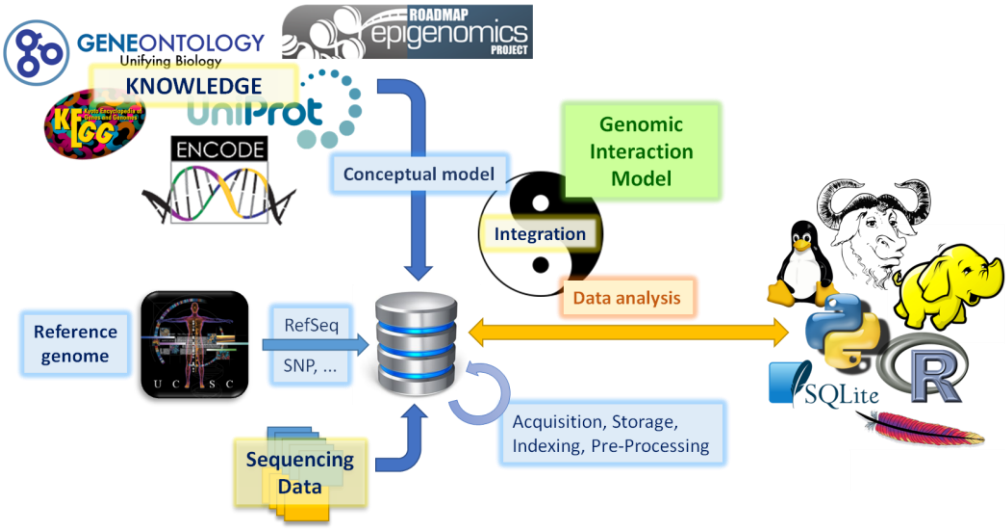


Table 5. List of Bioinformatics-relevant projects, powered by MapReduce/Hadoop or allowing parallel computing.

Project	URL	Ref.
Galaxy	https://galaxyproject.org/	[58]
CloudBurst	https://sourceforge.net/projects/cloudburst-bio/	[59]
Crossbow	http://bowtie-bio.sourceforge.net/crossbow/index.shtml	[60]
GATK	https://software.broadinstitute.org/gatk/	[61]
HBase	http://hbase.apache.org/	[94]
Pig	https://pig.apache.org/	[95]
BioPig	https://github.com/JGI-Bioinformatics/biopig	[96]
SeqPig	https://sourceforge.net/projects/seqpig/	[97]
ApacheSpark	http://spark.apache.org/	[98]
SparkSeq	https://bitbucket.org/mwiewiorka/sparkseq/wiki/Home	[99]
Mahout	http://mahout.apache.org/	[100]
R Project	https://www.r-project.org	[101]
Python	https://www.python.org/	[102, 103, 104]
Biopython	https://github.com/biopython/biopython.github.io/	[103, 104]

3.1. Conceptual Genomic Data Modelling

The ultimate purpose of a **conceptual model** is to provide concepts to generate a **logical data structure** (LDS), that could be faithfully used as a reference for data storage, management, retrieval, and analysis. An efficient LDS should provide: (i) basic, precise, and non-redundant **definitions** that can be reused as building blocks to generate complex entities; (ii) entity and relationship types should be traced through a **controlled vocabulary** to reflect a unique and standard notation; and (iii) an easily **modifiable architecture and hierarchy** to be adapted to fast evolving protocols and methodologies. However, genomic data models present more complications due to the intrinsic **domain complexity**, reflecting the presence of many kinds of relationships, that could be **non-monotonic** in nature (i.e. properties from sub-classes could be used for super-class definition), and **fast evolving** in time, where many interactions can be changed in relation to the **context** (e.g. the cellular compartment), or having **transient** but **significant** effect.

Many **conceptual genomic models** (CGM) are built within the basis of traditional data modeling techniques, including the **Relational Data Model** (RDM), the **Object-Oriented Data Model** (OODM), and the **Object Relational Model** (ORM) [105-107]. A relational database requires a **data base management system** (DBMS) for data acquisition and update, and the housed data must be modeled compliant with a specific **database schema**, that is a **computable description** of the **data domain**. Data schemas are usually generated in collaboration between domain experts and modelers. Users interact with the database via software applications and user interfaces. Generally, CGMs model qualitative information, including definitions, functions, processes, and interactions taken from molecular biology. To this end, many genomic-oriented systems use biological ontologies, such as the **Sequence Ontology** (SO) [74] and the **Gene Ontology** (GO) [72], providing standards for sequencing technologies and functional genomics, respectively. An ontology is a representation of the different types of entity that exist in the world, and the relationships that hold between these entities. An example of ontology-based relational data schema is CHADO [105], used to manage biological knowledge for a variety of organisms, from human to pathogens, as part of the **Generic Model Organism Database** (GMOD) toolkit [62]. In CHADO, ontologies are used as **typing entities** and for **metadata**, used as extensible data properties. This can be a winning strategy, given the increasing number of available open ontologies. Some notable examples are Disease Ontology (DO) [73] for diseases and disease-related genes, CHEBI [108] for chemical species of biological interest, PATO [109] for phenotypic qualities, and other ontologies from the **Open Biomedical Ontology** (OBO) foundry [75]. Knowledgebases may also help in offering a set of updated entities and relationships of biological relevance, with a set of associated unique terms and a hierarchical

structure that may guide model construction. Common examples are the Kyoto Encyclopedia of Genes and Genomes (KEGG) [76] and Reactome [77] for metabolic reactions, MINT [110] and IntAct [111] for molecular interactions, InterPro [112] for protein domains, and CATH [113] for protein architectures. However, the hierarchies provided by these tools hide another limitation: their architecture may not reflect the actual data-driven topology, leading to biased or misleading results. For example, several commonly used ontologies have a **Directed Acyclic Graph** (DAG) structure, that do not match many real biological interactions. One well-known effect is the attraction of computational predictions by **hubs** (i.e. well-characterized genes), obscuring other less-characterized and, possibly, more specific genes. Furthermore, many databases could be biased towards terms belonging to the same domain, leading to a misleading domain-specific over-representation [114].

Most CGMs are based on the concept of **genomic feature**, including every genetically encoded or transmitted entity, such as genes, transcripts, proteins, variants, and so on. These features come generally with necessary attributes, including **feature location**, often referred as **genomic coordinates** or **locus**, feature **provenance** and **database cross-reference**, **ontology**, and a series of optional and **extendible attributes**. Alongside CHADO, some notable genomic data schemas include: ACeDB, ArkDB, Ensembl, Genomics Unified Schemas (GUS), and Gadfly [105]. Modelers often use **Unified Modeling Languages** (UML) [115] as a tool to produce implementations from these data schemas. However, given the variety and complexity of biological entities and interactions, more flexible XML-based languages are often adopted, such as the **Systems Biology Markup Language** (SBML) [116], a free and open interchange format for computer models of biological processes, and **GraphML** [117], a

general XML-based file format for graph representations. More recently, a growing number of analysis tools support parallelization, typically based on **MapReduce** [55], a programming paradigm proposed by Google, and **Hadoop** [56, 57], its open-source implementation for single nodes and clusters. Notable projects in **bioinformatics** and **cloud computing** comprise distributed file systems, high-level and scripting languages tools, and services, including HBase, Pig, Spark, Mahout, R, Python, CloudBurst, Crossbow, GATK, and Galaxy (Table 5). Compared to other **parallel processing** paradigms, including grid processing and Graphical Processing Unit (GPU), MapReduce/Hadoop have two main advantages: (i) **fault-tolerant storage** and reliable data processing due to replicated computing tasks and data chunks, and (ii) **high-throughput data processing** via batch processing and the **Hadoop Distributed File System** (HDFS), by which data are stored and made available to slave nodes for computation. This paradigm respond to a frequent data science description of big-data, i.e. the **4-V definition**: volume, variety, velocity, and value. New parallel programming frameworks are designed for working with large collections of hardware facilities, including connected processors (i.e. cluster nodes), using a **Distributed File System** (DFS), that employs much larger units than the disk blocks in a conventional operating system [54].

3.2. The region-based Genomic Data Model

Responding to the need for parallel computational solutions, we developed the **GenoMetric Query Language** (GMQL) [118] as an **Apache/Pig-based** language for genomic cloud computing, ensuring clear and compact user coding for: (i) searching for relevant biological data from several sources; (ii) formalize biological concepts into rules for genomic data and region association; (iii) integrate heterogeneous kinds of genomic data; (iv) integrate publicly available and custom data. GMQL was developed upon a region-based **Genomic Data Model** (GDM), where each genomic region is modeled as a quadruple r_i = < *chrom, left, right, strand* >, defining a genomic segment coordinates, where *chrom* is the chromosome identifier, *left* and *right* are the starting and ending positions along the chromosome, and *strand* is a strand value in the set *{+, -, .}*, indicating forward, reverse, and unstranded, respectively (some features, such as protein binding, are often mapped indifferently on the two strands). Genomic intervals follow the UCSC notation, using a 0-based, half-open interbase coordinate system, corresponding to *[left, right)*. The GDM allows the description of a **biological sample** (*S*) as a set of homogeneous (i.e. same NGS protocol) genomic regions, each one having region-associated data (v_i). Every sample is characterized by a unique identifier (*id*) and a set of arbitrary *attribute-value pairs* (m_i) describing the sample **metadata**, usually containing information about the source organism (e.g. *Homo sapiens*), NGS experiment type (ChIP-seq, RNA-seq, HiC, etc.), the molecular bio-marker used for the experiment (e.g. the antibody used in a ChIP-seq experiment), the tissue or cell line from which the DNA/RNA sample was extracted, and a phenotype description. Metadata can be also standardized using controlled vocabularies and ontologies. Therefore, the generic sample schema in

the GDM is defined as the quadruple: $S = < id, \{ r_i , v_i \}, m_i >$. Another implicit rule is that data must come from the **same genome assembly**, to avoid chromosome ID and coordinate inconsistencies. However, even different assemblies could be uniformed, with few possible errors, by the UCSC LiftOver tool [67, 119].

In GMQL, **collections of NGS data samples** are represented as **variables**. Each variable can be transformed and combined, according to a wide range of functions, to define and create genomic objects with increasing complexity. NGS data and annotations can be combined with previously defined genomic objects and then serialized or reused to make new queries. Genomic objects can be homogeneous samples, with related data and metadata, or complex objects, such as enhancers, long-range interactions, or custom structures, deriving from a combination of heterogeneous data types. A GMQL query is a sequence of operations having the general schema: $v = operator(T) V$; where v is the output variable, V is the set of input variables required by the *operator*, and T the set of its parameters. GMQL operators allow metadata manipulation (*SELECT, AGGREGATE, ORDER*), region manipulation (*PROJECT, COVER, SUMMIT*), and multi-sample manipulation (*MAP, JOIN, UNION, DIFFERENCE*). The *SELECT* operator is used to create initial GMQL variables, by filtering out data from an initial dataset (e.g. the ENCODE hg19 ChIP-seq data), using their metadata attributes. Serialization, for example into a GTF file, can be done using the *MATERIALIZE v* operation, where the data samples inside the variable v are sent to a file.

To illustrate GMQL flexibility, we will go through a step-by-step example. Let us generate the GMQL code for the following task (modified from Masseroli *et al*. 2015, Supplementary Material: *GMQL Functional Comparison with BEDTools and BEDOPS*):

Find all enriched regions (peaks) in the CTCF transcription factor (TF) ChIP-seq sample from the K562 human cell line which are the nearest regions farther than 100 kb from a transcription start site (TSS). For the same cell line, find also all peaks for the H3K4me1 histone modifications (HM) which are also the nearest regions farther than 100 kb from a TSS. Then, out of the TF and HM peaks found in the same cell line, return all TF peaks that overlap with both HM peaks and known enhancer (EN) regions.

```
# Task 1. DATA COLLECTION.
# From ENCODE hg19 data (stored in the dataset
# HG19_PEAK), we extract ChIP-seq samples for the TF
# CTCF in the form of “peaks” (see paragraph
# 3.3) and for histone modification H3K4me1

TF = SELECT(dataType == 'ChipSeq' AND view == 'Peaks'
     AND cell == 'K562' AND antibody_target == 'CTCF'
     AND lab == 'Broad') HG19_PEAK;

HM = SELECT(dataType == 'ChipSeq' AND view == 'Peaks'
     AND cell == 'K562'
     AND antibody_target == 'H3K4me1'
     AND lab == 'Broad') HG19_PEAK;

# The TF and HM samples are now homogeneous
# collections.
# Annotations (TSS and enhancers) can be collected in
# the same way, from the hg19 annotation dataset
# (HG19_ANN):

TSS = SELECT(ann_type == 'TSS' AND provider == 'UCSC')
      HG19_ANN;

EN = SELECT(ann_type == 'enhancer'
     AND provider == 'UCSC') HG19_ANN;
```

```
# Task 2. DATA PROCESSING.
# Extract only TF and HM peaks farther than 100 kb fro
# a TSS

TF2 = JOIN(first(1) after distance 100000,
           right_distinct) TSS1 TF1;

HM2 = JOIN(first(1) after distance 100000,
           right_distinct) TSS1 HM1;

# Extract only farther TF peaks overlapping with
# farther HM peaks and enhancer regions

HM3 = JOIN(distance < 0, int) EN1 HM2;

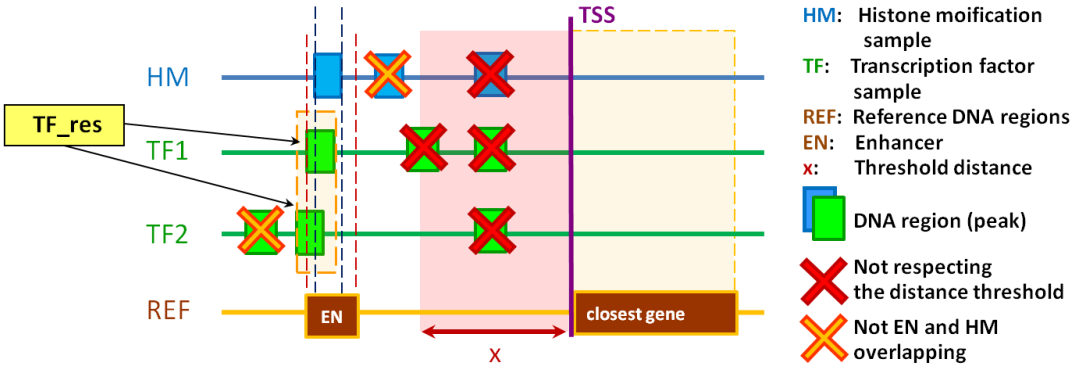
TF_res = JOIN(distance < 0, right) HM3 TF2:

# Variable TF_res is now an heterogeneous collection of
# TF and HM regions


# Task 3. SERIALIZATION.

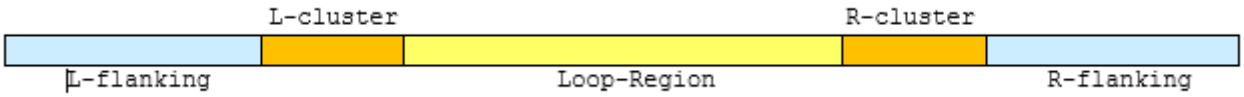
MATERIALIZE TF_res;
```

Figure 11. Graphical representation of the sample GMQL query.



GMQL is flexible enough to model different kind of genomic features, generating heterogeneous genomic objects, by processing many samples at a time with good scalability. However, similarly to other available methods for genomic data modelling and algebra implementation (such as BEDTools [120], BEDOPS [121], GQL [122], and GROK [123]), a declarative-like approach may underperform in modelling complex interactions. To have a clear example, we will now walk through the conceptual steps needed by

GMQL to model **enhancer-promoter interactions** (EPI). EPIs are fundamental genomic interactions, needed to enhance transcriptional activity at a level that is compatible with life. These interactions modulate a cell RNA synthesis by physically bending genomic DNA to put enhancer and promoter into contact through a series of proteins, including **CTCF**, **Cohesin**, the **Mediator Complex**, and **Transcription Factors** (TF) [124]. Different NGS technologies can be used to detect these interactions, such as 5C, HiC, and ChIA-PET [21-25]. As an example, we will consider the last one. ChIA-PET (pre-processed) data can be considered as a collection of **distal** genomic **region pairs**, that may belong to the same (**intra-chromosome**) or different chromosomes (**inter-chromosome**). For simplicity, the example will consider only intra-chromosome interactions. In these interaction data, every entry is a pair of interacting regions generating a **loop**, that can be schematically represented as follows:



where the left-cluster and the right-cluster of mapped reads define the interacting regions, that will be placed one against the other (formally, we may consider them as collapsed to a coincident genomic location). In the following example, GMQL will be used to represent this structure as an enhancer interacting with the associated promoter in a ChIA-PET experiment in *Mus musculus* MEF cells (assembly mm9). We must load data into variables, first:

```
# Loading ChIA-PET data and annotations

CHIAPET = SELECT( dataType == 'ChIA-PET' ) MM9_CHIAPET;

PROMOTERS = PROJECT (true; start = start - 2000,
                    stop = start + 1000) REF;

ENHANCERS = SELECT( cell == 'MEF' ) MM9_ENHANCERS;
```

We then start data manipulation to model the interaction of enhancers with the associated promoters. The initial ChIA-PET data has the following schema:

```
<chromosome, loopStart, loopEnd> read1, read2, score
```

where the attributes in < . > denote the genomic region being processed, while the other attributes are its associated data values. Values `read1` and `read2` store the coordinates of the left and right cluster, respectively.

```
# Initial ChIA-PET data:
#     <chromosome, loopStart, loopEnd> read1, read2, score
# Storing coordinates as data:
#     <chromosome, loopStart, loopEnd> read1, read2, score,
#     loopStart, loopEnd

JUNCTION_L_tmp = PROJECT(true; loopStart as left,
                         loopEnd as right) CHIAPET;

# Moving to left junction:
#     <chromosome, loopStart, read1> read1, read2, score,
#     loopStart, loopEnd

JUNCTION_L = PROJECT(true; right = read1) JUNCTION_L_tmp;

# Annotating PE-overlapping left-junctions:
#    <chromosome, peStart, peEnd> read1, read2, score,
#    loopStart, loopEnd

PEL_tmp = JOIN(distance < 0, left) ENHANCER JUNCTION_L;

# Storing coordinates as data:
#     <chromosome, peStart, peEnd> read1, read2, score,
#     loopStart, loopEnd, peStart, peEnd

PEL = PROJECT(true; peStart as left, peEnd as right) PEL_tmp;

# Moving to right junction:
#     <chromosome, read2, loopEnd> read1, read2, score,
#     loopStart, loopEnd, peStart, peEnd

PRJ_L = PROJECT(true; left = read2 AND right = loopEnd) PEL;

# Annotating promoter-overlapping right-junctions:
#    <chromosome, promStart, promEnd> read1, read2, score,
#    loopStart, loopEnd, peStart, peEnd

PRJ_L_tmp = JOIN(distance < 0, left) PROMOTERS PRJ_L;
```

```
# Storing coordinates as data:
#    <chromosome, promStart, promEnd> read1, read2, score,
#    loopStart, loopEnd, peStart, peEnd, promStart, promEnd

PEG_L_tmp = PROJECT(true; promStart as left, promEnd as right)
PRJ_L_tmp;

# Restoring original loop coordinates:
#    <chromosome, loopStart, loopEnd> read1, read2, score,
#    loopStart, loopEnd, peStart, peEnd, promStart, promEnd

PEG_L = PROJECT(true; left = loopStart AND right = loopEnd)
PEG_L_tmp;
```

Variable `PEG_L` contains the desired biological structure, when the enhancer is upstream the promoter.

The length of the code, the quantity of data manipulation passages, and the inefficient modelling (only upstream promoters are modelled after many passages), are a consequence of the region-based paradigm followed by GMQL. One main limitation in the GDM, and the other conceptual models discussed in the previous paragraph, is the lack of a comprehensive **genomic interaction model** (GIM). As we will see in the next paragraph, biological interactions may differ in type (e.g. protein-protein interaction, signalling, metabolic, etc.) and mode (e.g. short-range, long-range, undirected, causal, time-dependent, non-canonical, non-linear, etc.). The vast majority of CGMs neglect many of these aspects, especially because modelling complex interactions with a conceptual relational-like and/or ontology-based model is not practical, and in many cases, impossible. For instance, none of the aforementioned approaches could model non-linearity, or context-dependent causal relationships, where different interactions involve the same genomic entities in different contextual conditions, such as variable gene expression, factor binding, and methylation levels, or the presence of DNA damage. In these cases, **network** [125-134] and **structural models** [135-138] allow an efficient integration between conceptual, logical, and analysis layers, performing way

better than traditional conceptual data models, that are designed specifically for data acquisition, storage, and distribution, where a static data model is sufficient.

3.3. Biochemical signatures and the Genomic Interaction Model

In biology, *a **signature** is a collection of features significantly associated to a specific observable trait (i.e. phenotype)* [35-38], allowing detection, classification, or prediction of a genomic feature on the base of experimental data. The process by which we quantify and collect molecular signatures is called **profiling**. If a specific **data profile** X is significantly associated to a **biological trait** Y, then X can be considered a signature of Y. Testing the X-Y association is a key step in building an interaction model. In molecular genomics, a signature identifies a specific **chromatin state** (see Paragraph 1.3) that is supposed to underlie a specific molecular function, biological process, phenotype, or disease [35-38]. Consequently, a chromatin state defines a **genomic functional element**. The ENCODE [35, 36] definition of genomic functional element is a **discrete genome segment** that encodes for a product (e.g. a protein or non-coding RNA), or displays a **reproducible biochemical signature**. However, this definition is still influenced by the idea that a specific genomic feature must be defined through a fixed **locus**. As a matter of fact, in every model discussed in paragraphs 3.1 and 3.2, including the GDM, a feature is identified by its locus, reflecting the gene-centric point of view characterizing traditional genetics and molecular biology approaches, and considering the genome as an invariant reference. For many discrete genomic features including genes, transcripts, and enhancers, a static definition is usually sufficient. However, static models encounter

severe limitations in at least two conditions. Firstly, when defining personal variability, allelic, single nucleotide, and structural variation make the adoption of a unique reference impractical. Secondly, the biochemical alterations occurring in specific cell cycle phases or in response to an environmental stimulus may affect the activity and the interactions of genomic elements, causing topological rearrangements [23, 24, 124], and translocations [16, 41]. All these features cannot be represented by a static genomic model. As an example, when we tried to model an EPI with a locus-based model (i.e. the GDM), we ended up in a massive and inefficient (i.e. partial) definition. In the first two chapters of this section, we clarified how **the genome is not a fixed entity**, but deeply changing even between cells of the same individual, and the same cell through its lifetime. Many genomic features could be not faithfully represented by a region-centric definition, especially when they involve **multiple interacting loci**, that are **copy-variant**, **mobile**, **time-variant** in their activity, and **non-linear** in their interactions. An example that we will extensively discuss in the last chapter is provided by the DNA-binding protein distribution modifications due to huge chromosomal rearrangements in cancer. DBPs **occupancy** is routinely measured using ChIP-seq assays, in which per-base continuous genomic signals are converted into discrete Protein-DNA binding regions, called **peaks**. Peak calling is performed through genome-wide testing against the background distribution, that is usually derived either from a Poisson or a Negative Binomial model (see Paragraphs 3.5.1-2). Every peak in a ChIP-seq data sample comes with a **signal value** (i.e. an aggregate measure of its original per-base value), and a **p-value**. If the ChIP-seq experiment is explorative (i.e. we do not have *a priori* information about the nature and genomic distribution of the sequenced factor), we do not have the possibility to functionally validate peak calling. Typically, signal discretization leaves

out many subtle interactions, that turn to be true signals. Moreover, in many cases, such as the RNA Polymerase II associated transcription factors, there is no genomic site-specificity. However, even neglecting the problem of assigning a discrete DNA segment to a protein-binding signal, other sources of perturbation may cause huge modifications in protein occupancy. **Genetic fusions** arise from chromosome breakage, and characterize many cancer types [41]. Broken chromosome ends are reactive and may melt with other chromosomes, generating chimeric DNA segments. If the breakpoint involves two genes, the resulting transcriptional unit will be a fusion of the 5’-end of one gene and the 3’-end of the other, giving rise to a fusion protein. Fusion-protein binding properties are completely different from the original ones, with devastating regulatory effects, including the modification of other proteins binding distribution. These fusions can be intra- or inter-chromosome and can be assimilated to a form of non-canonical chromosome and protein-protein interactions (PPI). In general, feature definition may involve multiple combinations of variables, including **distance**, a (continuous) **dynamic range** of **binding**, **transcriptional**, and **epigenetic signals**, with **interaction terms** connecting all or some of these variables. Here we come to one fundamental definition of this work:

In general, a ***genomic feature*** *is represented by a specific* ***chromatin state****, defined through* ***context-specific variables*** *that can be* ***profiled using HTS technologies****, and which relationships can be* ***modeled and tested through an interaction model***. In this way, we can represent every probable chromatin state (as a **functional element**) and possible alteration (i.e. **physiological and pathological functional change**).

There are at least five main conceptual improvements in this definition:

(i) **Genomic features are not fixed entities**. Chromatin state definition is **data-driven** and therefore it can be **dynamically updated** respect to **time-variant biological conditions** (e.g. the **environmental influence**), and the specific **genomic context** (e.g. **cell cycle phase**).

(ii) **Genomic features are knowledge-independent**. Knowledge can be used to **improve** chromatin state classification/prediction, or to **annotate** novel genomic features and their interactions (i.e. used as feature attribute).

(iii) **Unknown genomic features are included in the modeling framework**. Signature profiling allow the identification of novel, previously unknown, features.

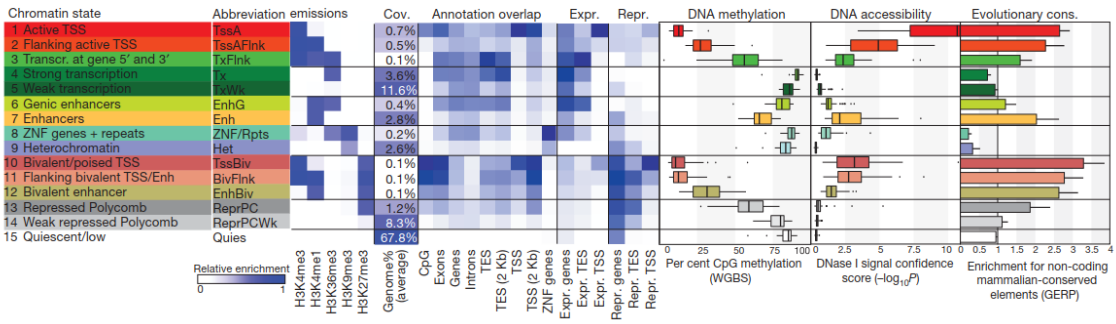
(iv) **Time-variant interactions are part of the feature definition**. The same variables may interact depending on **different biological conditions and genomic contexts**, resulting in **different chromatin states**. Therefore, an acceptable genomic model must be a **genomic interaction model** (GIM). As a logical consequence of this paradigm, **every chromatin state is associated to a biological function**, derived from its interaction set, and thus defining a genomic **functional element**.

(v) **Genomic coordinates identify an instance of a chromatin state**. A functional element exists as defined by a **state-specific time-variant signature**, that can be either a **ubiquitous** (e.g. gene or enhancer) or a **unique** (e.g. centromere for each chromosome) genomic feature, either

constitutive (e.g. inherited driver mutation) or **transient** (e.g. sporadic mutation, not inherited from parents). The chromatin state is instantiated in one or more loci.

Property (v) of the model present a substantial paradigm inversion respect to traditional genomic conceptual models. In other words, the genomic feature is no more defined as a discrete DNA segment, but rather as a genomic abstraction (i.e. a **class**, in the object-oriented programming paradigm) that can be instantiated as a **genome segment** (i.e. a locus), a **motif** (i.e. a DNA/RNA/protein sequence pattern), or an heterogeneous collection of the two. Figure 12 shows chromatin state definitions according to Roadmap Epigenomics [38]. Clearly, chromatin state definition is not dependent on its genomic location.

Figure 12. Chromatin state following Roadmap Epigenomics classification. *Chromatin states are identified by their signal profiles, including histone modifications, accessibility DNAme, expression, evolutionary conservation, and other features. Every chromatin state sefines a class of genomic feature. See main text and reference [38] for details.*



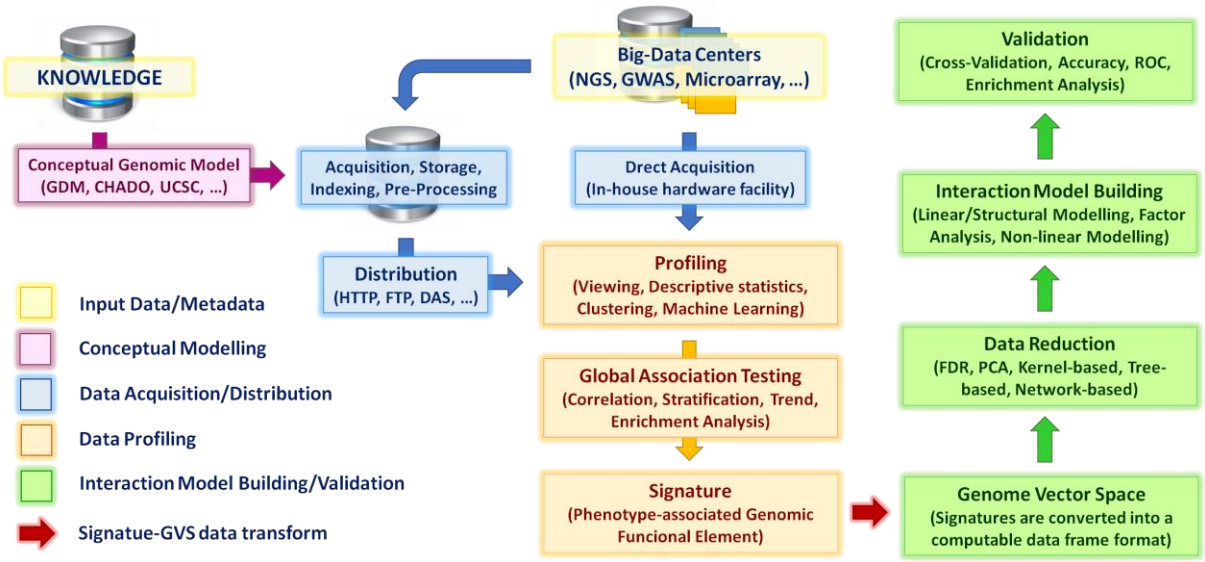
Building a comprehensive GIM is clearly not a trivial task, given the complexity of biological interactions. First, we need a method to select the **informative variables** that could be used as **predictors** of a chromatin state. Machine learning methods, including HMM and

Support Vector Machines (SVM), are fundamental for accurate classification, although they cannot provide information about causal relationships among variables. However, machine learning approaches have three advantages that can be used in model building: (i) they can be exploited as **model-free explorative methods** to collect a first set of evidences to be used as a **guideline** (this task has already been successfully accomplished by the ENCODE, Roadmap Epigenomics, and related projects, and therefore it could be not necessary), (ii) they can be combined to generate more efficient and accurate **ensemble methods**, and (iii) they can be used in the model **validation** phase, for **prediction**, rather than classification.

Figure 12 shows the general framework in which a GIM should be generated, and Figure 13 illustrates the general steps for its creation and for connecting it to conceptual modelling and data production. Knowledge and sequencing data are normally acquired using different protocols. Given the multiplicity of data sources, **distributed systems** are often a practical solution. For instance, the **Distributed Annotation System** (DAS) [143] is an open source project to exchange annotation data in bioinformatics, currently adopted by the UCSC Genome Browser [67, 119], the Ensembl genome browser [144, 145], FlyBase [146], InterPro [112], CATH [113], and other large bioinformatics databases. Other user-friendly tools, such as the Integrative Genomics Viewer (IGV) [85] or the Integrated Genome Browser (IGB) [86], allow client-side custom (i.e. local) and distributed data integrated viewing through the DAS, HTTP and FTP protocols. Thanks to web-based and open-source integrative platforms, such as BioMart [68, 69] and Galaxy [58], NGS data can be stored, analyzed, and distributed directly from remote servers. We have two types of data sources: Knowledge (from Ontologies, Knowledgebases, and Genome Browsers), and large experimental data sets (NGS, GWAS, Microarrays, and others), both

public (e.g. from SRA and GEO) and private. Typically, before integration, biologists visualize these data, for manual quality check, explorative inspection, and integrated viewing of different data types. The last task consists in the superposition of: (i) annotations, including known genomic functional elements (e.g. genes), and (ii) experimental data (e.g. TFs binding, histone modifications, chromatin accessibility, and annotated genetic variants) from international consortia (see Table 4) and custom **data tracks**. In the genome browser jargon, a data track is a homogeneous data representation, in one of the supported formats, including BED, BigWig, BedGraph, and GTF/GFF (see Paragraph 1.5), aligned onto the reference genome assembly. In a genome browser, more data tracks can be aligned and visualized at a time. However, genome browsers are not analysis tools and do not provide any information about biologically significant portions of a track, or possible data interactions. Currently, there are browsers that allow visualization of interaction data from ChIA-PET and HiC experiments, including the ChIA-PET Browser [147] and 3D Genome Browser [148], and several R packages to analyze it [149-151]. However, a comprehensive framework for modeling and analyzing biological interactions is missing. In the following discussion, we will go through a step-by-step modeling design that will consider several aspects of profile testing, data reduction, model selection, and validation.

Figure 13. Conceptual data flow schema for the creation a genomic interaction model. *NGS and other high-throughput biological data, together with annotations (knowledge), are stored in large bio-medical databases (gold boxes). Big bio-data are shared through distributed systems, using web protocols including HTTP, FTP, and DAS, or generated de novo by in-house wet-lab protocols (blue boxes). Raw data is then profiled, pre-processed (cleaning and quality check), and prepared for analysis and validation (orange boxes). Prior to the analysis, data is transformed into an efficient data structure (red arrow), that in our case is the Genomic Vector Space (GVS). Finally, the (reduced) data model is generated from the GVS. Usually, the main goals of the data model are classification and prediction, followed by both computational (e.g. cross-validation, prediction accuracy, ROC curve) and biological (e.g. enrichment analysis) validation.*



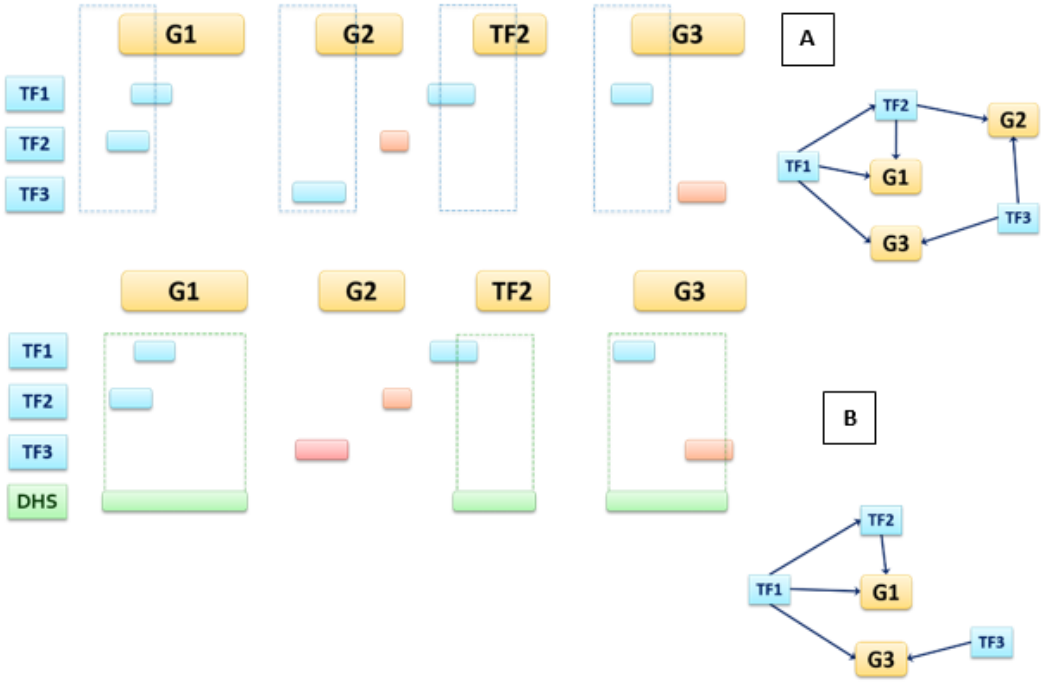
The general purpose of such an interaction model is to integrate concepts especially from the data analysis layer, that are currently not covered by conceptual models, and only developed into stand-alone methods. The general working schema to build an efficient GIM is sown in Figure 13. As we can see, traditional conceptual modeling is directly derived from knowledge, as it proposes high-level abstraction for genomic features. Although there are efforts to account for modifications respect to standard features, the outcome is either the

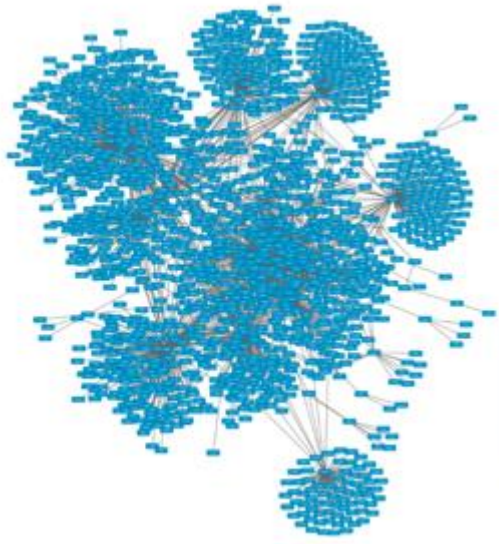
explosion of required parameters and data manipulations steps to model one to few feature interactions, or the impossibility of modeling some important feature attribute. This is what we could define as a lack of **model efficiency**. We saw an example of model inefficiency at Page 57. This type of conceptual modeling is useful to describe knowledge and the semantic relationships characterizing it. While this is not efficient for data-drive applications, it can be used to integrate data, characterize it in preliminary explorative analyses, or during the final evaluation of results (e.g. by enrichment analysis). Therefore, conceptual modelling (purple box in Figure 13) is tightly related to data characterization, collection, and distribution. Static, knowledge driven, representations of data could hold for traditional technologies, including microarrays, in which features are hard-encoded onto the genome (probes, genes, common variants). However, the exploratory nature of sequencing data do not allow such hard encoding. Even the genomic distribution of the signal is modelled and tested using analytical methods (i.e. read mapping). For this reason, sequencing data can be represented generically as **genome**-**wide profiles**. A profile is a density distribution in which *the genome represents the **support** of the **signal density function***. Modeling a genomic feature with a region-based paradigm is equivalent to model a function using only independent segments of its support, leading to an incomplete description. From paragraph 3.5, we will see the statistical framework necessary to model genomic profiles. In general, genome-wide testing for enriched signal density **peaks** and **summarization** (i.e. quantification of profiles on the base fixed genomic regions), allow us to move from whole-genome profiles to a vector representation, that we define the **Genome Vector Space** (**GVS**). GVS are multidimensional data structures deriving from **signal profile sampling**. In their simplest implementation, GVSs are analogous to an R data frame, i.e. a list of

vectors (see Figure 19 for a working example). In general, a GVS is an interaction model having form $M = \{S, \eta(X)\}_t$, where *S* is the **sample** (or **feature**) **space**, *X* is the **variable space**, and *t* is the **model evolution vector**, that is usually time or some other type of monotonic variable, including ordered categorical variables (i.e. strata). Function η, defines the set of interactions among X, identifying model **architecture** (e.g. a linear predictor, or the network structure among genomic features).

If we sample from a **population**, *S* (sampling space) are the subjects (e.g. cases and controls), and *X* are the observed variables (e.g. SNPs). If we sample from a single **genome**, *S* is the set of genomic features (e.g. promoters), and *X* is the set of molecular profiles summarized over that features (transcription factors, histone modifications, gene expression, damage events, etc.); in this case, *S* is the feature space. If we sample from an **interactome**, *M* becomes an **adjacency matrix**, *S* is the set of source nodes, and *X* the set of sinks (i.e. target nodes).

Figure 14. Creation of a signaling network from a GVS. *A GVS is a multidimensional space in which both genomic features and variables (i.e. profiles) are represented. Figures A and B, below, represent the transpose of a GVS (for visualization purposes). In a sequencing experiment, TFs profiles were sequenced and mapped onto the reference assembly (hg19). Enriched density peaks are represented as colored boxes inside the GVS. Note that sequenced TFs are gene products, and their genes can be targeted by TFs as well. Two transformation rules are used to convert the GVS into a network: (A) TFs binding a promoter region (dashed cyan line in A) establish TF→Gene (weighted) interaction, and (B) an active interaction can be only present in a region of open chromatin (DHS; green dashed line in B). The application of these rules to a K562 GVS (Chronic Myeloid Leukemia cell line), led to the graph architecture (η) showed in (C).*





C

Cell line: **K562** (CML)
TFs: **95**
Nodes: **1717**
Edges: **2367**

In all cases, *t* is a multidimensional vector. In a time-series, *t* could be simply the set of *time steps*. In a temporal signaling network *t* may have 2 dimensions, e.g. *pathway* and *time* (i.e. every interaction can be placed in a specific pathway at a given time). Every dimension in *t* define a characteristic of the context, such that every *i*-th event $[S_i, \eta_i(X_i)]_t$ has a specific value for each point in *t*. As an example, 122 TF binding sites (TFBS) have been selected and downloaded in BED format for the K562 cell line (chronic myeloid leukemia, CML) from the ENCODE repository [36, 152], and converted into a GVS, as exposed in Figure 14.

3.4. The nature of biological interactions

What came out from the previous discussions is an incredibly large number of possible genomic features which interactions give rise to even larger **phenotype variability**, including the vast range of **human disorders**. Technology advances allowed researchers to uncover the sources of this variability, although they are still way far from revealing the complete landscape of biological interactions. At the beginning of sequencing sciences, the earliest projects focused on the production of reference genomes, that could embody species-specific variability. Now it is known that individual genetic variability can be even larger than between-species one. In a recent study [153], researchers sequenced the genome of a female bonobo individual (sp. *Pan paniscus*), named Ulindi, showing that more than 3% of the human genome is more closely related to chimps and bonobos than these species to each other. The autosomal Ulindi genome shares a 99.6% sequence identity with chimpanzee (sp. *Pan troglodytes*) and a 98.7%

identity with human genome (sp. *Homo sapiens*). Therefore, the evident phenotypical and behavioral differences among these three species are due to minor sequence differences having a deep impact on the entire architecture of genomic functional interactions.

A complete understanding of genomic interactions cannot be gained only through technological advances, neither through statistical methods alone, given the overwhelming computational burden of exhaustively modeling every single feature interaction. As a rough measure of a system complexity, we could employ the Bell’s number *B(n)*, equal to the number of possible partitions assumed by a set of *n* components [90]. With *n* equal to 3 elements (e.g. 3 interacting proteins), the number of possible partitions is $B(3) = 5$: none of them interact, 3 possible pairing interactions, and all of them interact. The problem is that this number grows faster than exponential (i.e. it has a factorial growth rate). For instance, a network of 30 proteins would have $B(30) \sim 8.467 \cdot 10^{23}$ possible partitions, that is the order of magnitude of Avogadro’s constant ($\sim 6.022 \cdot 10^{23}$), and almost the estimated number of stars in the universe ($\sim 10^{23}$-10^{24}). To have a perception of how the current technology would handle these numbers, fully analyzing the ~1000 different proteins interacting in a single neuronal synaptic membrane would take roughly 2000 years [90]. Even if we consider that most of the possible protein or neuronal interactions do not actually occur, we are still in the range of at least 10^5 interactions. Therefore, the only way to model such a complex system is to observe, describe, and use their **emergent properties**, i.e. those properties that cannot be predicted by observing single system subsets or components. Some advantageous emergent properties of biological interaction networks are: (i) the presence of **driver** elements, capable of influencing and possibly **controlling** the whole

system; (ii) **modularity**, that is the capability of a group of interacting system components to operate as a **single functional unit**; and (iii) **hierarchy**, i.e. the organization of system components in an observable order. These characteristics are included in the **hierarchical network models**, a class of **scale-free models** that well represent real-life interacting systems, such as protein interaction networks, metabolic networks, social networks, and the world-wide web. Nevertheless, stochastic and time-dependent variations acting on system interactions could make their modeling exceptionally hard. Having an effective structural modeling strategy is only part of the framework. Experimental and biological variability force data analysts to test and validate the significance of biological signal profiles and their possible **causal interactions**. The concept of causal interaction involves the presence of a significant association among interacting variables, and a specific hierarchy, in which a parental set of variables influence the behavior of a subordinated one (i.e. the outcome variable). We know that true biological interactions are only a subset of the possible ones. Given the exceptional dimension of the possible interaction space, an exhaustive search is unfeasible. Therefore, we need **efficient heuristic solutions** to **reduce** the interaction space and **search** for **significant interactions**. Data reduction and testing, presented as separate concepts, should be integrated in a single process: *reducing the initial interaction space (i.e. the interactome) to its **causal components***. This is particularly evident in the so-called **disease module** detection [125, 126, 130, 131]. If we define the **interactome** as the architecture η of possible molecular interactions of an organism (often represented by an ensemble of **known** molecular interactions), a disease module is an interactome subset which elements and/or interactions are **perturbed** respect to the **physiological conditions**. System perturbation can be

measured and statistically tested against a control data set. Statistical testing induces a series of other modeling issues summarized in Table 7 and discussed in Paragraph 3.5. Therefore, to have a computable representation of a complex biological (genomic) system, the **interaction architecture** must be coupled to a **statistical model**. Here, we will refer to this **integrated architectural-statistical system** as the **Genomic Interaction Model** (**GIM**). Ideally, a GIM should be able to include every possible feature in the biological system under consideration, but typically a limited set of sequencing data types are available for a study. This is because of “rate limiters”, represented mainly by sequencing and maintenance costs, that still heavily influence genome research in many centers [17-19]. When publicly available samples are compatible with the biological model in use (e.g. same protocol, organism, strain, or cell line), integration with external data is allowed, although knowledge integration may not catch individual-specific variability. In general, it is possible to define a set of common data types that are produced in almost every sequencing experiment. They include profiling of (commonly used NGS technologies are in square brackets): (i) **protein binding** [ChIP-seq]; (ii) **gene expression** [RNA-seq, GRO-seq]; (iii) **histone modification** (HM) [ChIP-seq]; (iv) **DNase-I hypersensitive sites** (DHS) [DNase-seq]; (v) **DNA methylation** (DNAme) [Methyl-seq, Bisulfite-seq]; and (vi) **chromosome-interacting regions** [ChIA-PET, HiC, 5C]. Assays (i) and (ii) are the most common ones, since they provide a direct measurement of gene expression and regulation, that are easily convertible into graphical interaction models [19, 26-34, 154-156]. Assays (iii), (iv), and (v) are used for epigenetic profiling (HM and DNAme), and to study regulatory mechanisms and chromatin accessibility (HM and DHS) [26-32, 34, 40]. Finally, group (vi) assays are used to map chromatin **long-range interactions** (LRI) [21-25, 124], including EPIs and huge

chromatin looping interactions, leading to the configuration of broad **topologically-associated domains** (TAD) [23]. Chromatin looping interactions can be profiled also using ChIP-seq, with specific antibodies against proteins involved in chromatin reshaping, such as CTCF, Cohesin, and components of the Mediator complex [124]. Special kinds of assay are employed to model non-canonical genomic features, such as those appearing in pathological conditions, including DNA damage and translocations [14-16]. Damaged chromatin regions can be mapped through γH2AX, with low resolution (~500Mbp). A much more sensitive technology, capable of mapping individual **double-strand break** (DSB) events is **BLISS** [160] (see Chapter 4 for details).

Table 6. Common genomic features described by sequencing profiles. *Different types of whole-genome profiling assays can be used to characterize precise genomic features. The following table lists the principal ones and the interactions they define. Assay code is: RNA-seq [R], GRO-seq [G], ChIP-seq [C], DNA methylation profiling [M], DNase-seq [D], HiC [H], ChIA-PET [P], and BLISS [B].*

Profile	Assay	Feature	Interaction type
Gene expression	R	Transcript, DEG, Gene fusion	Co-expression, NC
RNA synthesis	G	Transcript, DEG, Gene fusion	Co-expression, NC
DNA binding	C	PBS, TFBS, Promoter	Signaling, Binding
Chromatin regulation	C	Promoter, Enhancer, SE, Polycomb	Co-regulation
DNA methylation	M	Promoter, CpG islands, DMR	Co-regulation
Chromatin accessibility	C, D	Promoter, Enhancer, DHS	Binding
Chromatin topology	H, P	Chromatin domain, TAD	EPI, LRI
DNA damage	C, B	gammaH2AX, DSB	NC

Gene fusion events can be detected by observing anomalies in the gene expression profile, through RNA-seq and GRO-seq [154-159]. Table 6 reports a schematic overview of the common genomic features defined trough specific sequencing assays. Many error sources may influence the discovery of true biological signals, due to technical limitations, and the presence of an intrinsic background noise. Given the deep sensitivity reached by sequencing technologies and the

compositional differences and biases among diverse genomic regions, true biological signal is usually **heteroschedastic**, **overdispersed**, masked by **complex interactions** and **non-linear effects**, and could be close to **background noise**. For these reasons, the first step after signal profiling is cleaning and testing the significance of the profiled signal. Table 7 shows common issues raised by feature and feature interaction testing. All these aspects, and the possible modeling solutions, will be discussed in the next paragraph. The general take-home message is that **biological signal profiles** are modelled not just as fixed entities (like, for example, in the relational model), but rather as **random variables**, whose behavior can be described and tested using underlying **statistical models**.

Table 7. Typical issues in causal modeling and possible solutions.

Issue	Description	Statistical method
Background variability	Background variability may partially overlap with true signal variability, even in good experiments.	• **Local background model** • **Reproducibility**
Heteroschedasticity	Different sub-populations of (genomic) signals come with different variance. In other terms, groups of genomic loci can be considered as sampled from different populations.	• **Variance correction** • **Mixture modelling** • **Stratified modelling**
Over-dispersion and **zero-inflation**	Over-dispersion occurs when biological signal variance is higher than expected by a background distribution (e.g. Poisson). Zero-inflation, instead, occurs when the dataset is sparse, with more zero-values than expected by the background model.	• **Model variance correction** • **Negative binomial distribution** • **Zero-inflated models**
Confounding	Confounding happens when an external confounder variable masks the relationship between an independent and a dependent variable. The confounder can be an environmental factor (e.g. air pollution), a phenotypical condition (e.g. age or weight), or a contextual variable (e.g. gene expression).	• **Mixture modelling** • **Stratified modelling**
Type I error **(false positive)**	Incorrect rejection of the null hypothesis (i.e. non-significant difference between signal and background), when it is true. In other words, the biological signal is considered significant when it is actually noise.	• **p-value threshold** • **FDR correction** • **Variance moderation**
Multi-collinearity	Two or more variables in the specified model are strongly correlated, in such a way they could be assimilated as a single variable.	• **Variance inflation factor** • **Model reduction** • **PCA**
Non-linearity	Linear modelling is often the best and less complex solution to model a biological phenomenon. However, many sources of non-linear interactions (e.g. long-range interactions and gene fusions) may deeply affect causal relationships among variables.	• **Linearity testing** • **Non-linear modelling** • **Model-free analysis**
Sampling bias	Sampling from a population of (genomic) signals may lead to biased estimates due to compositional inconsistencies (e.g. biased gene expression) or unmodelled interactions (e.g. genes from the same pathway), causing artificially reduced or inflated variability, respect to the whole population.	• **Randomization**
Overfitting	Overfitting occurs when a classifier (or a predictor) performs very well on a given sample (or training) set, but underperform with a blind validation set. This happens because the classifier (or predictor) is built (trained) onto a restricted dataset, lacking of generality and/or negative examples.	• **Model variance-bias and complexity optimization** • **Cross-validation** • **Area under ROC**

3.5. Biochemical signatures and the Genomic Interaction Model

In the previous paragraph, we discussed the possible sources of complexity in biological systems and the common statistical issues related to their modelling. In this paragraph, we will enter the details of statistical feature and feature interaction testing. The first required step is to assess the statistical significance of the profiled biological signal. In the following paragraphs, a comprehensive statistical framework for sequencing data will be presented, to provide a backbone for further genomic interaction modeling.

3.5.1. Models for tag density peak detection

Genomic functional features can be represented in a variety of different ways; the most common is through **discrete genome segments**. When deriving from an enriched signal density profile(s), a genomic segment is called a **peak**. As discussed in paragraph 1.5, a good sequencing experiment include libraries composed by tens of millions of DNA fragments, called **reads**, to be aligned onto the **reference assembly**, or assembled in a reference-free manner. Once aligned, a series of quality checks must be performed to avoid sequencing biases and technical artefacts that may influence read alignment distribution. Once aligned, reads are often referred as **tags**. **Tag density distribution** along the genome is **not uniform** and **sparse**. *A **tag density peak**, or simply a peak, is a genomic region (i.e. a locus) in which there is a **significant accumulation** of tags over **measured or estimated background*** [26].

Every tag density peak has a **source**, defined as the **point of chromatin cross-linking**, that is the point in which a chromatin-binding protein of interest is covalently bound to DNA, to obtain a complex that can be precipitated using a protein-specific antibody, and purified from the sample solution. This is the central step in **Chromatin Immunoprecipitation** (**ChIP**) [26-34], used to investigate **protein-chromatin interactions**, including **histone modification profiles**. The source of a peak corresponds to one of the protein **inferred binding site**. Common examples of chromatin-binding experiments include **transcription factor binding site** (**TFBS**), **repair factor**, and **histone mark** (**HM**) **binding/occupancy** profiling. In general, seven steps in peak profiling can be delineated: (i) **signal profile definition**, (ii) **background model definition**, (iii) **peak call criteria**, (iv) **artefact peaks filtering**, (v) **significance ranking** and **signal scoring**, (vi) **multi-sample normalization**, and (vii) **multi-sample data integration**. Steps (i) to (v) involve single samples, while steps (vi) and (vii) may also involve data from other protocols that ChIP-seq, including RNA-/GRO-seq, variant calling, and DNA damage profiling.

Cross-linked genomic DNA is sheared using physical (e.g. sonic waves) or enzymatic (e.g. MNase) agents. In any case, DNA fragments are generated randomly to have overlapping fragment ends. Overlapping fragments are fundamental for two reasons: (i) they do not map in the same position, and (ii) they can be concatenated by end superposition. The former property allows duplicate (i.e. artefact) elimination, since duplicates map exactly in the same genomic position, avoiding signal over-estimation. However, in many sequencing experiment duplicates are not artefact. For instance, we retain duplicate reads when we sequence the same, coincident, event (e.g. sequence-specific enzymatic cut) happening in multiple cells. Sequencing fragments have

a typical length of ~150-400bp, but for sequencing efficiency reasons, usually only their ends are PCR-amplified and sequenced (**paired-end library**). This means that every fragment is represented by two aligned tags, called **paired-ends**: one for the forward and one for the reverse strand. Every fragment is 25-150 bp long, depending on the sequencing protocol. Tags with missing mates and unmapped reads must be excluded from **signal profiling**. Signal profiling, consisting in **tag count smoothing**, include genome **binning**, **sequence biases correction** and **signal quantification**. Binning is generally performed by different algorithms using non-overlapping windows, ranging from few tens of bases (typically ~30-50bp) up to 1Mbp. Reads count is always scaled by library size and compared to a given **background model**, to generate a local score (*s*). The typical model is based on the **Poisson distribution**, in which mean and variance can be modelled as a single parameter (λ), having mass probability function:

E1

$$P(n) = e^{-\lambda}\frac{\lambda^n}{n!} \quad \forall n \in \mathbb{N}$$

with the probability of finding at least *m* reads in each genome bin equal to:

E2

$$P(n \geq m) = 1 - \sum_{n=0}^{m-1} e^{-\lambda}\frac{\lambda^n}{n!}$$

where *n* is the variable modelling the number of reads per bin. While the Poisson model offers the advantage of estimating a single parameter (λ), it inherently assumes mean-variance equality. This assumption has been questioned in different studies [33, 34], as not matching empirical data variability. This issue arises from two main causes: (i) intrinsic **sequence bias**, and (ii) biological data **over-dispersion**. Sequence biases can be due to either local or global

genome properties. **Local chromatin structure**, **copy number variants**, together with **DNA amplification** and **nucleotide composition**, may have a deep impact over biological signal dispersion. While explicitly modelling every property of the DNA sequence is not feasible, many commonly used algorithms, including MACS [33] and SICER [34], do account for local λ corrections. In MACS, the local λ is defined independently for each peak:

E3

$$\lambda_{local} = max(\lambda_{BG}, [\lambda_1,] \lambda_5, \lambda_{10})$$

where λ_{BG} is the plain Poisson λ, and λ_1, λ_5, and λ_{10}, are local λ estimations in a window of 1, 5, and 10kbp, respectively, centered at the peak location in the control sample, or in the ChIP-seq sample, in absence of a control (in this case, λ_1 is omitted). ChIP-seq signal has a typical bimodal signal, which modes are separated by a distance (*d*), that is estimated by cross-correlation profiling between strands. In MACS, the forward and reverse modes are shifted by *d*/2 towards the 3' end. After mode shifting, 2*d* sliding windows are tested along the genome to find significant tag enrichments. Overlapping windows are merged and each tag position is extended by *d* bases from its center. The point in the defined peak region with the maximum tag density is called **summit**, i.e. the estimated binding site. These types of enriched tag density regions are called **point-source peaks**. Although most protein binding factors and HMs can be called this way, some HMs are more efficiently defined using a **broad-source peak** detection method. These peaks can be represented either by broad signal profiles without a clear local maximum (e.g. H3K27me3), or as a series of multiple local summits close to each other (e.g. H3K27ac). As an alternative approach, the genome can be segmented in fixed-length (*W*) non-overlapping bins and then tested for tag enrichment over the background. This is the strategy pursued by SICER method [34].

Considering an experiment with library size *N*, the score of a genome bin containing *m* reads is defined as $s(m) = -\log P(m, \lambda)$, where $P(m, \lambda)$ is a Poisson distribution with parameter $\lambda = N \cdot W/L$. *L* is the **effective genome fraction** [31, 32], defined as the amount of *uniquely mappable genome sequence*, used for tag count normalization. $P(m, \lambda)$ defines the probability of finding *m* reads in a genome bin, with a uniform background probability for a read to be mapped across the genome (random mapping). Therefore, the higher the score, the less the probability that the tag density profile is observed by chance. Clusters of enriched regions, called *islands*, receive an additive score, equal to the negative logarithm of the joint probability of the observed parameter configuration, given the random background model. The probability $M(s)$ to find an island with score *s*, at a given genomic position is:

E4

$$M(s) = h^{g+1}\tilde{M}(s)h^{g+1}$$

where *h* is the probability for a bin to be ineligible for being part of an island, *g* is the gap size allowed for bin merging, and $\tilde{M}(s)$ is a recursive kernel function defined as:

E5*a*

$$\tilde{M}(s) = G(\lambda, m_0, g)\int_{s_0}^{s}\tilde{M}(s - s')\rho(s')ds'$$

with boundary condition $H(\lambda, m_0, g)^{-1}$, and where m_0 is the read count (height) threshold, $s_0 = -\ln P(m_0, \lambda)$, and $\rho(s)$ is the score probability distribution for a single bin:

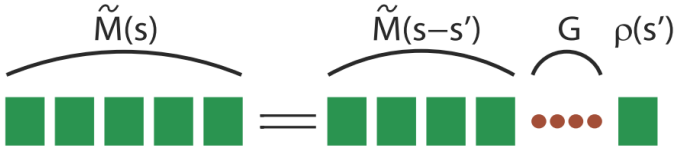
E5*b*

$$\rho(s) = \sum_{m \geq m_0}\delta(s - s(m))P(m, \lambda)$$

with $\delta(\cdot)$ being a Dirac delta function (i.e. it is equal to 1 when $s - s(m) = 0$). $G(\cdot)$, defined as *gap factor*, equals the portion of ineligible genome for an island incorporation:

E6 $$G = 1 + h + h^2 + \cdots + h^g$$

Figure 15. Graphical representation of SICER recursion.



Here, the effective (unique) genome fraction L is modelled as a constant factor, specific for a given organism. A more general, sequence-specific, concept of **mappability** was introduced for the first time by Rozowsky *et al*. (2009) [31], where they showed how PolII ChIP-seq tag density peaks at TSS are enriched in uniquely mappable bases. In this case, mappability was computed as a table in which *k*-mers from a specific genome locus are related to their occurrence in the entire genome. This definition allows L to be computed on any individual genome. In the original procedure, 12-mers were used to build a binary hash table recording their genomic positions, for each chromosome. The hash table is then used to optimize 12-mer occurrence count in the genome. A *k*-mer with occurrence 1 is a uniquely mappable segment of genome.

CG content can also affect sequencing and mapping quality. The stronger triple G≡C hydrogen bond influence RNA Polymerase efficiency and cause PCR amplification biases. Similarly, to uniquely mappable *k*-mers, CG dinucleotides are particularly enriched at TSS, where ~70% of human promoters show the presence of CpG islands (i.e. regions with higher CpG content than the rest of the genome) [161].

3.5.2. Over-dispersion, zero-inflation, and the Negative Binomial Model

Given a random variable *Y*, called **outcome** (e.g. the number of mapped reads at a genomic locus), we may define the **mean-variance relationship**, according to the Poisson model, as:

E7 $$\mathrm{var}(Y) = \mathrm{E}(Y) = \mu$$

where E(*Y*) = μ is the expected value of *Y*, and var(*Y*) its variance. In the previous paragraph the expected Poisson outcome value has been indicated with λ, to highlight its identity with variance. While the Poisson model identifies the variance of a random variable with its expected value, a more general definition considers the variance of *Y* as **proportional** to its expected value:

E8 $$\mathrm{var}(Y) = \phi\mathrm{E}(Y) = \phi\mu$$

where ϕ can be assumed as a **dispersion parameter**. In other words, if we consider the Poisson model as a reference (i.e. ϕ = 1 and μ = λ), a model for which ϕ > 1 is defined as **over-dispersed**, or **under-dispersed** if ϕ < 1. Biological count data often falls in the former definition [162]. This means that fitting biological count data to a Poisson model (ϕ = 1) would lead to consistent parameter estimates, but conservative standard errors respect to the over-dispersion condition (ϕ > 1), prejudicing significance assessment. One solution is to correct standard errors by estimating ϕ using the generalized Pearson’s χ^2 statistic, with *n* observations and *b* parameters:

E9 $$\chi^2 = \sum_{i=1}^{n} \frac{(y_i - \mu_i)^2}{\mathrm{var}(y_i)} = \frac{(y_i - \mu_i)^2}{\phi\mu_i}$$

Equating E9 to its expected value $n - b$ and solving by ϕ, the estimator for ϕ will be:

E10

$$\hat{\phi} = \frac{\chi_b^2}{n-b}$$

An alternative and general strategy is to define the **conditional distribution** of the outcome by modelling θ as an **unobserved multiplicative random effect**, so that $Y|\theta \sim P(\mu\theta)$. This random effect captures the unobserved variability that either increases ($\theta > 1$) or decreases ($\theta < 1$) the expected outcome μ for a given set of covariates X. If we took $E(\theta) = 1$ with θ being observable, so that μ represents the expected outcome for the average individual, we would fall into the Poisson model. Although θ is **not observable**, we could add assumptions to its distribution to compute the **unconditional distribution** of the outcome. Since θ can assume any non-negative real value, a convenient way of modelling it is through a **Gamma distribution** with parameters α (**scale parameter**) and β (**rate parameter**, $1/\theta$). Gamma distribution has mean α/β and variance α/β^2, so if we take $\alpha = \beta = 1/\sigma^2$, distribution mean becomes 1 and its variance σ^2. This is the distribution of the **number of failures** before the k-th **success** in a series of **independent Bernoulli trials**, with common probability of success equal to p and failure equal to $1 - p$. This is called the **Negative Binomial distribution**, having the general form (with $Y = y \in \mathbb{R}^+$):

E11

$$P(Y=y) = \frac{\Gamma(\alpha+y)}{y!\,\Gamma(\alpha)} \frac{\beta^\alpha \mu^y}{(\mu+\beta)^{\alpha+y}}$$

where $\alpha = k$ and $p = \beta/(\mu + \beta)$.

The probability mass function $NB(k; h, p)$ in the case $Y = k \in \mathbb{N}$, with h being the number of failures and k the number of successes is:

E12

$$P(Y = k) = \binom{k + h - 1}{k} p^k (1 - p)^h \quad \forall k, h \in \mathbb{N}$$

with mean $hp/(1 - p)$ and variance $hp/(1 - p)^2$. In the general form (Equation E11), with $\alpha = \beta = 1/\sigma^2$, expected value and variance (i.e. the first raw moment and second central moment, respectively) can be derived without the assumption of a Gamma distribution for θ, given $E(\theta) = 1$:

E13*a*

$$\mathrm{E}(Y) = \mathrm{E}[\ \mathrm{E}(Y|\theta)\ ] = \mathrm{E}(\theta\mu) = \mu\mathrm{E}(\theta) = \mu$$

E13*b*

$$\mathrm{var}(Y) = \mathrm{E}[\ \mathrm{var}(Y|\theta)\ ] + \mathrm{var}[\ \mathrm{E}(\theta\mu)\ ] = \mathrm{E}(\theta\mu) + \mathrm{var}(\theta\mu) =$$

$$= \mu\mathrm{E}(\theta) + \mu^2\mathrm{var}(\theta) = \mu + \mu^2\sigma^2 = \mu(1 + \mu\sigma^2)$$

Therefore, the Negative Binomial distribution has **mean** μ (i.e. consistent with the Poisson λ), and **variance** $\mu(1 + \mu\sigma^2)$. If σ^2 is assumed as dispersion parameter ϕ (see Equation E8), we conclude that, for $\sigma = 0$, there is no dispersion respect to the expected value μ, and therefore $NB(\mu, \phi)$ collapses into a Poisson $P(\mu)$ distribution. However, *if* $\sigma > 0$, *then the variance* $\mu(1 + \mu\sigma^2) > \mu$, *and therefore there is* ***over-dispersion*** *respect to the Poisson model*. Although the Negative Binomial can model possible data over-dispersion, it does not account for **zero-inflation**, that is the *presence of more zero outcomes than expected by the model*. In this case, we can use a model with **two latent classes** of statistical units (e.g. individuals of a population, or genes of an individual genome): (i) **zeroes** (i.e. $Y = 0$), and (ii) **non-zeroes** (i.e. $Y > 0$). This model induces a combination of a **logit model** (see Paragraph 3.5.4), predicting which of the two latent classes a

statistical unit belongs, and a **Poisson** or **Negative Binomial** model, predicting the outcome for the non-zeroes latent class.

3.5.3. Modelling count data using Generalized Linear Models

Till now we discussed the general statistical framework to model genome-wide tag density enrichments, considering genomic properties that may cause deviations from the background statistical model. We also discussed how the vast majority of the genome-wide signal is not significantly higher than the background noise, generating sparse profiles and zero-inflated signal distributions. In a typical ChIP-seq experiment, discrete genomic intervals inside the aligned raw signal profile are isolated and called as tag density peaks. These peaks provide an empirical and statistical evidence for the definition of genomic features including protein-binding sites, and regions marked by histone modifications. However, in many cases, we want to quantify signal density over already existing features, including transcriptional units, genes, exons, TSS, and enhancers, that are usually provided to analysis algorithms as GTF or GFF files. This procedure is called **summarization**. Summarization is a typical procedure in RNA-seq, but it can be extended to any other type of sequencing count data, including ChIP-seq. In an RNA-seq assay, purified RNA is converted into complementary DNA (cDNA) by a reverse transcriptase. cDNAs are naturally produced by retroviruses to integrate their RNA-based genetic information into DNA-based genomes. Since cDNAs from mature mRNAs do not contain introns, they can also be used for cloning eukaryotic genes in prokaryotes. cDNA is sheared and sequenced like in other NGS technologies. Usually, RNA-seq assays are performed to monitor gene activity in one or more experimental

conditions, cell cycle stages, or to compare the expression profile across different phenotypes. In every case, a data matrix is produced. As in other NGS methods, there should be at least 2 technical replicates per sample [27, 28]. Typically, one data matrix is produced per experiment, containing tens of thousands of features (usually genes or transcripts) as rows, and as many columns as samples (e.g. individuals, including replicates). Samples can be further grouped by experimental condition (e.g. diseased vs. Healthy, or treated vs. untreated). Likewise, ChIP-seq and other sequencing technologies, RNA-seq (and GRO-seq) produce count data (i.e. natural numbers) that tends to be over-dispersed and zero-inflated. The total number of uniquely mapped tags for each *i*-th sample is representative of its library size (N_i). The genome is represented by a set of homogeneous features (e.g. genes) $g = 1, 2, ..., G$; where G is the total number of features, corresponding to the row count in the data matrix. Therefore, considering n samples, count data is organized in a $G \times n$ matrix. This being said, also for RNA-seq a $NB(\mu, \phi)$ model is used. According to equation E13*a*, the expected feature expression y_{gi} (e.g. gene expression) for the feature g in the *i*-th sample is:

E14

$$\mathrm{E}(y_{gi}) = \mu_{gi} = \nu_{gi} N_i$$

where $y_{gi} \sim NB(\mu_{gi}, \phi_g)$, and ν_{gi} is the proportion of mapped reads for feature g in the *i*-th sample. It is also assumed that tag counts from repeated runs of the same RNA sample follow a Poisson distribution, and therefore $\mathrm{var}(y_{gi}|\nu_{gi}) = \mathrm{E}(y_{gi}|\nu_{gi})$, that corresponds to technical variability. The proportion ν_{gi} is not actually observed. Rather, we measure the fraction of cDNA fragments p_{gi} produced for a feature g in the *i*-th sample, underlying the unobserved feature expression. In other terms, we assume that $\mathrm{E}(y_{gi}|p_{gi}) = p_{gi}N_i$. According to equation

E13*b*, var(y_{gi}) = $\mu_{gi}(1 + \mu_{gi}\phi_g)$. The dispersion parameter ϕ_g is equal to the squared *coefficient of variation* (CV), and μ_{gi} is defined by equation E14. Variable $\phi_g^{1/2}$ represents the **biological coefficient of variation** (BCV). This model is currently implemented in the edgeR package [157-159] to test differential expression in (RNA-seq) count data. In differential expression analysis, we have multiple experimental conditions j = 1, 2, ..., m. Comparing them implies doing pairwise comparisons, in which the null hypothesis is always no difference between expected gene expression between two conditions *J* and *K*, i.e. H_0: $\nu_{giJ} = \nu_{giK}$, or in other terms, the contrast H_0: $\nu_{giK} - \nu_{giJ} = 0$. To model gene expression levels in every condition, we need a function *f* that relates the outcome *Y* (e.g. gene expression) to the effect of every condition in each sample, considering a standard normally-distributed prediction error ε:

E15

$$Y = f(X_1, X_2, \ldots, X_m) + \varepsilon$$

where X_j is called a **predictor**. Typically, *f* is not known *a priori*, and therefore it is assumed to be simple and adaptable function, that is usually a linear one. Linear models can be generally expressed by the equation:

E16

$$y_{gi} = \beta_{g0} + \beta_{g1} x_{gi1} + \beta_{g2} x_{gi2} + \cdots + \beta_{gm} x_{gnm} + \varepsilon_{gm}$$

where β_{g0} is the **intercept**, x_{gij} are the predictors for every *j*-th condition, and β_{gj} are the **effects** of the condition over the outcome. This model assumes a Normal distribution of count data (e.g. gene expressions), and a standard Normal distribution for the error term, implying E(ε) = 0. Model general form can be written as:

E17*a*

$$y_{gi} = \sum_{j=0}^{m} \beta_{gj} x_{gij} + \varepsilon_{gj}$$

or in matrix notation:

E17*b*

$$Y = \begin{bmatrix} y_{g1} \\ y_{g2} \\ \vdots \\ y_{gm} \end{bmatrix} = \begin{bmatrix} \beta_{g0} \\ \beta_{g1} \\ \vdots \\ \beta_{gm} \end{bmatrix} \times \begin{bmatrix} 1 & x_{g11} & x_{g12} & \cdots & x_{g1m} \\ 1 & x_{g21} & x_{g22} & \cdots & x_{g2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{gn1} & x_{gn2} & \cdots & x_{gnm} \end{bmatrix} + \begin{bmatrix} \varepsilon_{g0} \\ \varepsilon_{g1} \\ \vdots \\ \varepsilon_{gm} \end{bmatrix} = \beta_g X + \varepsilon$$

where X is called **design matrix**, and β_g is the **parameter vector**, i.e. the vector of effects to be estimated from every condition. However, we defined our count data model using an underlying $NB(\mu, \phi)$ distribution. Hence, we rather use a **generalized linear model** (**GLM**), that is characterized by a generalized linear equation, defined as follows:

E18

$$E(Y) = u^{-1}(\beta X) = \mu \qquad \text{or equivalently} \qquad u(\mu) = \beta X$$

where βX is the **linear predictor** (i.e. a linear combination of unknown parameters β, to be estimated), and $u(\cdot)$ is called **link function**, specific for every GLM distribution family, providing the relationship between linear predictor and mean. The link function can be expressed in terms of mean-variance relationship, having general form:

E19

$$\mathrm{var}(Y) = \mathrm{v}[\, u^{-1}(\beta X)\,] = \mathrm{v}(\mu)$$

where $\mathrm{v}(\mu)$ is a function over the mean. In summary, GLMs can be decomposed into: (i) a **random component** $Y|X \sim f(\theta, \phi)$ that is assumed to follow an **exponential** behaviour, (ii) a **systematic component** βX (i.e. the **linear predictor**), and (iii) the **link function** $u(\cdot)$ such that $u(\mu) = \beta X$. The exponential (random) component has the general form:

E20*a*

$$f(y;\theta,\phi) = \exp\left\{ \frac{\tau y - b(\tau)}{a(\phi)} + c(y,\phi) \right\}$$

where functions $a(\cdot)$, $b(\cdot)$, and $c(\cdot)$ are known functions identifying a specific GLM family, with θ being the **canonical** (or **location**) parameter, and ϕ the **dispersion** (or **scale**) parameter, corresponding to the first and second moment of the function, respectively:

E20*b*

$$E(y) = b'(\tau) = \mu_i \qquad \text{and} \qquad \text{var}(y) = b''(\tau)a(\phi) = \sigma_i^2$$

where $b'(\cdot)$ and $b''(\cdot)$ are the first and second derivative of $b(\cdot)$, respectively. Function $a(\cdot)$ is in the form $a(\phi) = \phi/\omega$, where ω is the prior weight, usually fixed to 1. As reported at the beginning of the discussion, this framework can be used for any summarized count data besides RNA-seq, including GRO-seq and ChIP-seq (accounting for possible technology- and platform-specific biases). Here the problem with GLMs is to find the correct estimators for θ and ϕ, that best fits our observations (e.g. the biological signal profiled through sequencing data). This problem can be generalized to any set of parameters θ. For simplicity, let us consider a set $y = y_1, y_2, \ldots, y_n$ of *n* **independent and identically distributed** (**iid**) **observations**, sampled from a distribution with unknown probability density function $\tilde{f}(y)$, that is assumed to describe reality. If $\tilde{f}(y)$ comes from a **distribution family** $f(\cdot|\theta)$, characterized by the parameter set θ, we define $\tilde{f}(y) = f(y|\tilde{\theta})$. The distribution family is called **parametric model** (e.g. one of the GLM family models). The aim of **Maximum Likelihood Estimation** (**MLE**) [162] is to find an estimator $\hat{\theta}$ that can assume values as close as possible to the true parameters $\tilde{\theta}$. The **likelihood function** *L* is mathematically defined as the **joint probability distribution** of **observed data**:

E21*a*

$$L(\theta|y) = \prod_{i=1}^{n} f(y_i|\theta)$$

transformed to log-likelihood ($\mathcal{L}$), for computational convenience:

E21*b*

$$\mathcal{L}(\theta|y) = \sum_{i=1}^{n} \ln f(y_i|\theta)$$

Differently from probability, that describes the chance of observing a data set y given the model with parameters θ (i.e. the underlying probability density function), likelihood describe the situation in which y (i.e. the set of observations) is given and we want to know the chance that a specific model, with parameters θ, generated it. Therefore, MLE has indeed two purposes: (i) **estimate** the **best parameter values** from data, and (ii) **compare different models** (i.e. different parameter sets) to assess which one **best fits** the data.

The first step is accomplished by finding the maximum of E21*b* (that is equivalent to maximizing E21*a*), so that $\partial^2 \mathcal{L}(\theta|y)/\partial\theta_j^2 < 0$ for *j* = 1, 2, ..., *m* (with *m* being the number of parameters). In other terms, the MLE for vector θ is defined as $\{\hat{\theta}_j\} \subseteq \{\arg\max_\theta \mathcal{L}(\theta|y)\}$, if a maximum for $\mathcal{L}$ exists. In cases in which the MLE cannot be solved analytically (e.g. when *m* is too large and the probability density function is non-linear), it is possible to use (heuristic) numerical optimization methods, that however are not guaranteed to reach the global optimum. These definitions can be extended also to non-iid conditions, if it is possible to define a joint probability density function, and *m* is finite and not dependent on *n*. In general, every GLM family can be fitted to data using the same **iterative weighted least squares** algorithm. At each iteration, a working dependent variable *z*, equal to the derivative of the link function (for each data point):

E22

$$z_i = \hat{\eta}_i + (y_i - \hat{\mu}_i)\frac{d\eta_i}{d\mu_i}$$

where $\hat{\eta}_i = \hat{\beta}X_i^T$ and $\hat{\mu}_i = u^{-1}(\hat{\eta}_i)$.

Then, the iterative weight is calculated as:

E23

$$w_i = \frac{\omega_i}{b''(\theta_i)\left[\frac{d\eta_i}{d\mu_i}\right]^2}$$

Parameter estimates are done by regressing *z* over the predictors x_i:

E23

$$\hat{\beta} = (X^T W X)^{-1} X^T W Z$$

that is the least square estimate of β, where *X* is the model matrix, *W* is the diagonal matrix of weights w_i, and *Z* is the response vector of elements z_i. The estimation is repeated until the estimate change by less than an arbitrarily small amount (i.e. until convergence).

Let us consider a model of interest M, with parameter estimates $\hat{\theta}_i$ and fitted values $\hat{\mu}_i$. We define the saturated model M^* as the model for which there are as many parameters to be estimated as data points. This model leads to perfect fit, but it would have no predictive value. Now let θ_i^* and $\mu_i^* = y_i^*$ be respectively the fitted values and parameter estimates for M^*. If we consider M^* and M as two competing **nested models** (i.e. the linear predictor of the smaller model is a subset of the largest one), the likelihood ratio criterion to compare them is defined as:

E24

$$-2\ln\Lambda = 2\sum_{i=1}^{n} \frac{y_i(\theta_i^* - \hat{\theta}_i) - b(\theta_i^*) + b(\hat{\theta}_i)}{a_i(\phi)} = \frac{D(y,\hat{\mu})}{\phi}$$

where Λ is the likelihood ratio between M^* and M, $a_i(\phi) = \phi/\omega_i$ and $D(y,\hat{\mu})$ is called **deviance**:

E25

$$D(y,\hat{\mu}) = 2\sum_{i=1}^{n} \omega_i\left[y_i(\theta_i^* - \hat{\theta}_i) - b(\theta_i^*) + b(\hat{\theta}_i)\right]$$

The negative sign is needed since Λ assumes negative values. In Paragraph 3.5.3, we saw how genomic count data is affected by **over**-

dispersion, and therefore, in the GLM framework, it is convenient to express the mean-variance relationship as var(*Y*) = $\phi\mu$. In case $\phi \neq 1$, the traditional MLE tends to underestimate dispersion [162]. The over-dispersion case fits the description of **Quasi-Likelihood** (QL) given by Wedderburn (1974) [163].

Maximum QL estimation (**MQLE**) assumption differ from MLE in some points. First, the true underlying density for the outcome variable is no more restricted to the exponential family. Secondly, the only requirements come from the definition of the first and second moments of the density function, and the relationship connecting them. Instead of considering the derivative of $\mathcal{L}$ respect to θ, we derive respect to the mean value μ. We refer to the mean-variance relationship expressed by equations E19 and E20*b*. For constant ϕ, var(*Y*) = ϕv(μ), with mean-variance relationship conditions:

E26

$$E\left[\frac{\partial \mathcal{L}(\theta)}{\partial \mu}\right] = 0 \qquad \text{var}\left[\frac{\partial \mathcal{L}(\theta)}{\partial \mu}\right] = \frac{1}{\phi \text{v}(\mu)}$$

According to McCullagh and Neelder [162], the function satisfying these conditions is:

E27

$$q_i = \frac{y_i - \mu_i}{\phi \text{v}(\mu_i)}$$

for every *i*-th data point. This function defines the quasi-likelihood function for each point as:

E28

$$Q_i = \int_{y_i}^{\mu_i} \frac{y_i - t}{\phi \text{v}(\mu_i)}$$

The ML estimator $\hat{\theta}$ for this function is solved by the equation:

E29

$$\frac{\partial}{\partial \theta} \sum_{i=1}^{n} Q_i = \sum_{i=1}^{n} \frac{y_i - \mu_i}{\phi \text{v}(\mu_i)} \frac{\partial \mu_i}{\partial \theta} = \sum_{i=1}^{n} \frac{y_i - \mu_i}{\phi \text{v}(\mu_i)} \frac{X_i}{u(\mu_i)} = 0$$

It has been demonstrated that MQLE is asymptotically consistent with the true estimand (i.e. for *n* tending to infinite, $\hat{\theta}$ is equal to $\tilde{\theta}$ with increasing probability), although it is often less efficient (and never more efficient) than a MLE. Different extended QL definitions have been provided. Based on extended QL model, a Pseudo-Likelihood (PL) function [162] has been developed, by substituting deviance, used in MLE and QLE, with Pearson’s χ^2. Other likelihood function modifications have been proposed for different applications. For example, the R package edgeR uses a weighted conditional likelihood [157-159] function to test differential expression using and underlying Negative Binomial model.

3.5.4. GLMs for (latent) categorical features

The **logistic** (or **logit**) **model** [162] is a regression method used when the outcome variable is **categorical**. When the outcome is in the mode {0, 1} (e.g. {healthy, *diseased*}), the outcome is **binary** (this is the modality discussed in Chapter 4, then it will be taken as reference for this section). Differently from a linear model, the conditional distribution $Y|X$ follows a **Bernoulli distribution** (rather than a Gaussian distribution), i.e. $Y|X \sim B(k; p)$, where $B(k; p) = p^k(1 - p)^{1-k} \forall k \in \{0, 1\}$, with mean $\mu = p$ and variance $p(1 - p)$. In our modeling framework, we assume $k = 1$ to be the *DNA damage* event. We experimentally genome-wide validated this event through BLISS assays [160], using the statistical framework exposed in Paragraphs 3.5.2 and 3.5.3. However, in principle, we wanted to assess which are the **causative factors** and/or the genomic conditions favoring DNA damage occurrence, to be able to **predict** such events. To this end, machine learning methods could be useful to classify such events on the base of

the available signal profiles, however they do not provide any information about possible causal relationships in data. For this reason, we modeled the possible *DNA damage* event as a **latent variable** η, defined as:

E30

$$\eta_i = \beta_0 + \beta X_i + \varepsilon_i$$

where $\beta = \beta_1, ..., \beta_m$ is the parameter vector. In other terms, we model the latent variable η as the linear combination of m+1 explanatory variables X_j (with j = 1, ..., m), plus the error term ε, for each i-th data entry. The dummy variable x_0 = 1 is equal for every data entry, and β_0 represents the intercept, i.e. the value assumed by η when all the predictors are equal to 0 (however, for us β_0 has no relevant biological interpretation). Variable η is used to assign data to one of the two outcomes $Y = \{0, 1\}$:

E31

$$Y = \begin{cases} 1 & \text{if } \eta > 0 \\ 0 & \text{otherwise} \end{cases}$$

In this model, the response variables Y_i are not identically distributed, since $P(Y_i = 1 \mid X)$ differs for every i-th data entry, although they are independent, given X (X is the design matrix; see Paragraph 3.5.3). To note, condition $\eta > 0$ is equivalent to $\varepsilon < \beta X$. In the logistic model, the error term ε follows a standard logistic distribution. Following definition E31, we have (β_0 will be included inside the βX term, for simplicity):

E32

$$P(Y_i = 1 \mid X_i) = P(\eta_i > 0 \mid X_i) = P(\beta X_i + \varepsilon > 0) =$$
$$= P(\varepsilon < \beta X_i) = \text{logit}^{-1}(\beta X_i) = p_i$$

The **probit model**, is defined the same way, but modeling ε as following a standard Normal distribution. Logistic has a similar shape as a Normal distribution, with heavier tails and therefore it is less

sensitive to outliers. This property is desired considering the almost constant presence of outliers in genomic count data. The logistic function is defined as:

E32

$$F(X) = \frac{1}{1 + e^{-(\beta_0 + \beta X)}}$$

that is the ***probability of the outcome to be a case*** (i.e. "1": *DNA damage*), ***rather than a control*** (i.e. "0": *non-damaged DNA*). Given his definition, the logistic model is a case of GLM, with logit link function:

E33

$$u(p) = \ln\left(\frac{p}{1-p}\right)$$

The function $u(F(x))$, called **log-odds** (or **logit**), (equal to the inverse of the logistic function; E32) is equivalent to the GLM linear predictor:

E34

$$u(F(X)) = \ln\left(\frac{F(X)}{1 - F(X)}\right) = \beta_0 + \beta X$$

where ***odds*** = $e^{\beta_0 + \beta X}$. For a continuous independent variable x_j (i.e. a predictor), the **odds ratio** (**OR**) is defined as:

E35

$$OR_j = \frac{\text{odds}(x_j + 1)}{\text{odds}(x_j)} = \frac{\frac{F(x_j + 1)}{1 - F(x_j + 1)}}{\frac{F(x_j)}{1 - F(x_j)}} = \frac{e^{\beta_0 + \beta_j (x_j + 1)}}{e^{\beta_0 + \beta_j x_j}} = e^{\beta_j}$$

Therefore, the OR is a measure of **odds unit increase**, i.e. the odds multiply by e^{β_j} for every unit increase in x_j (or equivalently, the log-odds increase by β_j for every unit increase in x_j). In the specific case of a 2×2 contingency table, the OR can be interpreted in terms of ratio between the occurrence of cases and controls in different conditions. Let us assume, for instance, the number of damaged genes, respect to a fixed threshold θ_P of DNA Polymerase 2 (*Pol2*). We have two

variables: *DNA damage* (binary), and *Pol2* (continuous). The odds-ratio will be defined as:

E36

$$OR = \frac{\frac{q}{c}}{\frac{Q-q}{C-c}}$$

where q is the number of damaged genes when *Pol2* > θ_P, c is the number of damaged genes when *Pol2* < θ_P, Q is the total number of genes with *Pol2* > θ_P , and C is the total number of genes with *Pol2* < θ_P. In this case, the association between *DNA damage* and *Pol2* could be tested using a χ^2 test or a **Fisher exact test** (see next Paragraph for details). This, however, implies to assess the **validity** of the threshold θ_P before counting, and in any case, do not account for the **simultaneous effect** of other variables (e.g. gene expression, transcript length, or the presence of repair factors). In the next paragraph, we will discuss all these aspects, gaining momentum in case of **confounding** and/or variable **interaction**.

Another problem with this model, as in every GLM, is over-dispersion. For n trials and k successes (e.g. the occurrence of k damaged genes in a set of n genes), the outcome distribution will follow a **Binomial distribution** with probability function:

E37

$$B(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad \forall k \in \mathbb{N}$$

having mean np and variance $np(1 - p)$. This distribution implies **independent trials**, with **common probability of success** p. *If these conditions hold, binomial data cannot be over-dispersed* [162]. However, if there are sources of **variability** or **correlation** among binary responses, not accounted by this model, over-dispersion may occur. In the DNA damage case, gene-gene interaction and gene co-expression may influence p, making it uneven for different groups of

genes (i.e. **group effect**). A similar effect is expected when the same variable is measured over time (i.e. **time series**, or **longitudinal study**). Therefore, the dispersion parameter ϕ accounts for the portion of **unexplained variability**, due to the departure from binomial model assumptions. Considering equations E13*a* and E13*b*, we can define the unconditional expected value and variance as:

E38*a*

$$\mathrm{E}(Y) = \mathrm{E}[\ \mathrm{E}(Y|\theta)\ ] = \mathrm{E}(n\theta) = n\mathrm{E}(\theta) = np$$

E38*b*

$$\mathrm{var}(Y) = \mathrm{E}[\ \mathrm{var}(Y|\theta)\ ] + \mathrm{var}[\ \mathrm{E}(n\theta)\ ] = \mathrm{E}[n\theta(1-\theta)] + \mathrm{var}(n\theta) =$$

$$= n[p - \phi p(1-p) - p^2] + n^2\phi p(1-p) = np(1-p)\sigma^2$$

where $\sigma^2 = 1 + \phi(n-1)$ is assumed as **heterogeneity** term, accounting for the deviations from the binomial model. Given this generalization, for $\phi > 0$ there is over-dispersion, for $\phi = 0$ we have the basic Binomial model, while for $n = 1$ the Binomial collapses into a Bernoulli distribution. Again, given Equation E10, ϕ can be estimated using Pearson's χ^2 with *b* parameters:

E39

$$\hat{\phi} = \frac{\chi_b^2}{n-b} = \frac{\sigma^2 - 1}{n-1}$$

with

$$\chi^2 = \sum_{i=1}^{n} \frac{(y_i - n\hat{p}_i)^2}{n\hat{p}_i(1-\hat{p}_i)}$$

3.5.5. Model complexity, bias-variance tradeoff, and model selection

Accurate **parameter** and **error estimation** is a key point to understand the significance of the **causal relationship** between predictor and outcome, as well as the capability of the entire model to describe them. A general problem, typical of every model, is that increasing the

number of descriptors model fitting increases almost always. However, this may lead to **overfitting**. This problem occurs when a model that is built on a sample data set, underperforms with a blind (i.e. external, completely new) test data set. Intuitively, this problem can be related to the number of predictors we use to explain the outcome or, in general, to the *difference between the predicted outcome value and its real value*. If this difference is almost 0 for the sample set, it will not ensure the same result on the test set. As a simple preliminary example, let us consider a linear model fitting with a polynomial function, having general form:

$$y(k) = \sum_{j=0}^{k} a_j x^j$$

where the *j*-th coefficient a_j multiplies the term with *j*-th degree x^j. The scalar $k \in \mathbb{N}$ defines the degree (i.e. the highest exponent) of the polynomial having *k*+1 terms. The degree is set by the user, and the difference between the predicted and actual outcome value will be lower and lower. However, the gain in fit will rapidly decrease with increasing *k* values, and the overall model will gain redundant descriptors. This is the case in which the model is too **complex**: too many canonical parameters are estimated with no reliable estimate for variance. In this case, the model will miss the general phenomenon underlying data. This can be summarized using two concepts: **bias** and **variance**. If the model can explain only a very specific data set, then it is **biased** towards a specific type of variability, meaning that it lacks in **generality**. The idea to find the best tradeoff between model complexity and bias is to reduce the number of predictors to a minimal set that minimizes the proportion of **residual variance**, that is the proportion of variability not explained by the model. Let us consider a training set composed by *n* observations x_i, with *i* = 1, 2, …, *n*. Suppose

the existence of a function f such that $y = f(x) + \varepsilon$, where every $y_i \in y$ is the true value for the corresponding observed $x_i \in x$. Each value predicted by $f(x)$ differs from the **actual value** y by an **error** ε, that is normally distributed with mean 0 and variance σ_ε^2. As exposed in Paragraph 3.5.4, f is usually not known and therefore the analyst chooses an *ad hoc* model to replace it. Specifically, we want to find a model $\hat{f}(x)$, approximating $f(x)$, such that the **expected squared prediction error** $\mathcal{E}(x)$, respect to y, will be minimized. The expected squared prediction error is defined as:

E40

$$\mathcal{E}(x) = E\left[\left(y - \hat{f}(x)\right)^2\right] = E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right] + \left(E[\hat{f}(x)] - f(x)\right)^2 + \sigma_\varepsilon^2 =$$

$$= \mathrm{var}(x) + \mathrm{bias}(x)^2 + \sigma_\varepsilon^2$$

where σ_ε^2 is called **irreducible error**, that is a noise term characterizing the $y \sim f(x)$ relationship, and that cannot be accounted by any model. In other words, $\hat{f}(x)$ could be at most as accurate as $f(x)$ in predicting the true value y, if only one could get rid of σ_ε^2. In general, a model with **low variance** will produce **precise** predictions (i.e. close to the mean prediction $E[\hat{f}(x)]$), while a model with **low bias** will produce **accurate** predictions (i.e. close to the true value y). The most commonly used index to assess **goodness of fit of a model** is the **coefficient of determination** (R^2), equivalent to the ***proportion of variance of the outcome explained by the predictors***:

E41

$$R^2 = 1 - \frac{S_R}{S_T} = 1 - U$$

where U is the proportion of **unexplained variance**, S_R is the **residual sum of squares**, S_T is the **total sum of squares**, and S_E is the **explained sum of squares** (for $i = 1, 2, \ldots, N$):

E42

$$S_T = \sum_i (y_i - \bar{y})^2 \qquad S_R = \sum_i (y_i - \hat{y}_i)^2 \qquad S_E = \sum_i (\hat{y}_i - \bar{y})^2$$

where y_i is the actual *i*-th value, $\hat{y}_i$ is the predicted value for y_i, and $\bar{y}$ is the average over the actual values (i.e. over y). In case of a **simple linear regression model** (i.e. the outcome is explained by only one predictor), the R-squared coefficient is equal to the square of the **Pearson correlation coefficient** (in this case, the notation r^2 is used). In general, the R^2 holds as a goodness of fit measure for linear regression. In logistic regression (see Paragraph 3.5.5), **deviance** (see Equation E25) is used instead of sum of squares, providing a measure of lack of fitting of the model given data. The reason is that, while linear regression assumes **homoschedastic error variance** (σ_ε^2), logistic regression has intrinsically **heteroschedastic errors**.

As anticipated in Paragraph 3.5.4, the **residual deviance** (*D*) is a function of Λ, that is the **likelihood ratio** between the **subject model** M and the **saturated model** M^*:

E43

$$D = -2\ln\Lambda = -2\ln\frac{\mathcal{L}_M}{\mathcal{L}^*}$$

Deviance has an asymptotic χ^2 distribution [162] and therefore can be used to test lack of fitting through a Pearson's χ^2 test. Hence, *a **non-significant p-value** is a sign that an arbitrarily small portion of variance is not explained by* M (i.e. the model has a **good fit**). In the R-software implementation of GLMs, the residual deviance is displayed with the related **residual degrees of freedom** (*b*), and then the significance can be assessed using the function: *p-value* = 1 – `pchisq`(*D*, *b*); where the `pchisq` function yields the probability density for the χ_b^2 distribution. This procedure verifies the proportion of unexplained variance of a model, providing a relative measurement respect to the saturated

model. To test the absolute efficacy of the subject model M, the difference respect to a **null model** (M_0) deviance is defined as:

E44

$$D_0 - D = -2\left[\ln\frac{\mathcal{L}_0}{\mathcal{L}^*} - \ln\frac{\mathcal{L}_M}{\mathcal{L}^*}\right] = -2\ln\frac{\mathcal{L}_0}{\mathcal{L}_M}$$

where *the null model has no other predictors than the* ***intercept*** (i.e. the grand mean). If this difference is significant (i.e. the residual deviance is significantly smaller than the null deviance), then *the linear predictor significantly improved model fitting*. The correspondent significance is given by: *p-value* = 1 – `pchisq`($D_0 - D$, $b_0 - b$). These two tests (residual deviance and null deviance difference) fall into the definition of **likelihood ratio test** (LRT), since they are both based on Λ. Deviance analysis and the LRT can be used to assess a single model.

However, in molecular biology we are typically in the situation of investigating the possible (i.e. unknown) causal effects of known factors, respect to an observed phenotypical or molecular trait. Every factor that is profiled to (potentially) describe the observed trait can be assumed as a predictor. Different *m* linear combinations of predictors generate *m* different linear predictors η_1, η_2, …, η_m, and therefore *m* different models M_1, M_2, …, M_m.

In this scenario, every model represents a possible combination of biological factors that could cause the observed trait. The capability of comparing and excluding alternative models allow us to understand which of the considered factors have a true causal effect on the trait. To assess this causal relationship, the observed trait must be encoded as an outcome. Usually a categorical (more often binary) outcome is used, e.g. {*diseased*, *healthy*}, {*treated*, *untreated*}, {*mutant*, *wild-type*}, {*driver*, *transient*}, and so on. Sometimes, outcomes could be conveniently combined, e.g. {*diseased treated*, *diseased placebo, diseased untreated, healthy*}. Another strategy is **stratification**, i.e. the

same outcome is modeled in different strata (groups) of the whole population. In Chapter 4, we will see this last example, in which the outcome *Y* = {*damaged promoter*, *damage-free promoter*} is modeled in four classes of genes, stratified by expression level.

In Paragraph 3.5.4, we discussed the role of MLE and its possible corrections for count data. Parameter estimation is critical to both assess the significance and estimate the effect of a given predictor. We saw how the first goal of MLE is parameter estimation, given the observed data, under the assumption that a specific model generated it. Since many different models can explain data with different efficiency, the second goal of MLE is to compare them to determine which is the best one (i.e. the one that best fits data), thus answering to the aforementioned issue. A general framework in probability and information theory to compare the distance between two alternative distributions, is provided by the **Kullback-Leibler divergence** (KLD), also called **discriminant information** [162]. As introduced in paragraph 3.5.3, our models can be represented by functions of data $\tilde{f}(y)$ that can be used to approximate the **unknown reality function** $f(y)$. We assume that there exists a family $\tilde{f}(y)$ of distribution functions, with parameters $\tilde{\theta}$, such that reality can be described according to it, i.e. $\tilde{f}(y) = f(y|\tilde{\theta})$. If we consider $f(y)$ and its approximation $\tilde{f}(y) = f(y|\tilde{\theta})$, the KLD theory can measure the amount of **information loss** when we represent reality with the given approximation:

E45

$$I(f, \tilde{f}) = \int f(y) \ln \frac{f(y)}{f(y|\tilde{\theta})} dy$$

This is sometimes called also **maximum entropy principle**, since $I(f, \tilde{f})$ correspond to the expected value of the negative Boltzmann’s

entropy. The aim of **model selection** is to minimize this information. In practice, the reality function cannot be known, and thus reported as a constant term. Therefore, paradoxically, reality is not considered. KLD theory is instead the key point to make relative comparisons between alternative approximations (i.e. alternative models) of reality, to understand which is *the most efficient representation of the **real-world process underlying data***. However, also in its relative version, the KLD is often too complex to be calculated. The **Akaike Information Criterion** (**AIC**) aims to *estimate the KL information loss by means of **maximum log-likelihood***:

E46

$$AIC = 2k - 2\mathcal{L}(\hat{\theta}|y)$$

where *k* is the number of parameters to be estimated, and $\hat{\theta}$ are the parameter values that maximize the log-likelihood function $\mathcal{L}$ (see Paragraph 3.5.4). The AIC decreases as the goodness of fit (maximum likelihood) increases, but is penalized by the increasing number of parameters *k*. Therefore, the AIC controls at the same time **model complexity** (variance) and **model fitting** (bias), avoiding over-fitting. In certain cases, such as over-dispersion and reduced sample size, the basic AIC definition cannot be applied due to problems with likelihood estimation (see Paragraph 3.5.4). Given the definition of quasi-likelihood, the **Quasi-AIC** (**QAIC**) for **over-dispersed count data** is defined as:

E47

$$QAIC = 2k - 2\frac{\mathcal{L}(\hat{\theta}|y)}{\hat{c}}$$

where $\hat{c}$ (often called *c-hat*) is a **variance inflation factor** (**VIF**), unique for the whole model.

Later on in the discussion we will see how to estimate VIFs for every predictor, in order to control **multi-collinearity**. Cox and Snell [164] proposed an approximation for estimating $\hat{c}$:

E48

$$\hat{c} = \frac{\chi^2}{b}$$

where *b* are the model's degrees of freedom. In practice, the $\hat{c}$ value can be approximated using the ratio between residual deviance (that is asymptotically distributed as a χ^2) and residual degrees of freedom. The empirical estimates for sampling variance $\mathbf{var}(\hat{\theta}_i)$ and covariance $\mathbf{cov}(\hat{\theta}_i, \hat{\theta}_j)$ can be simply obtained by multiplying the theoretical model-based variance and covariance times $\hat{c}$. If $\hat{c}$ is used, the number of parameters *k* in the QAIC increases by 1.

Another complication derives from small **sample size**. Sample size *n*, corresponding to the number of observations (i.e. **statistical units**), should be adequately higher than the number of variables *k*. When *N* is too small (i.e. *n* approaches *k*), the plain (Q)AIC under-performs due to inefficient parameter estimation. for this reason, a **second-order AIC** (called **AICc**) has been defined by Sugiura (1978) [165]. A general formulation for the (Q)AICc is:

E49

$$[Q]AICc = K - 2\frac{\mathcal{L}(\hat{\theta}|y)}{\hat{c}}$$

where

$$K = \frac{2k(k+1)}{n-k-1}$$

i.e. the parameter number term 2*k* is multiplied by a correction factor. This correction is generally applied for low **effective sample sizes** (namely when $n/k < 40$) [162]. If $\hat{c} = 1$ then the QAIC collapses into AIC. From now on, within the general model selection procedure description, the term "AIC" will implicitly include the possibility to

modify it to QAIC under the aforementioned conditions (the second-order (Q)AICc will never be used in this work). A systematic analysis of a set $\{M_j\}$ of alternative models can be done considering the **AIC difference**

E50*a*

$$\Delta_j = AIC_j - AIC_{min}$$

and the **relative likelihood**

E50*b*

$$\rho_j = e^{-\frac{\Delta_j}{2}}$$

for the *j*-th model. Given a set of competing models, the best model is the one having the lowest AIC. This model will be taken as reference and all the other models can be compared to it, yielding an AIC difference Δ_j. This measure can be used to calculate the relative likelihood ρ_j, that is proportional to the probability for the *j*-th model to minimize the estimated information loss, given the best model $\widehat{M}(\theta,k) = \arg\min_{\theta,k} AIC$. For instance, if ρ_j = *c*, then model $\widehat{M}(\theta,k)$ is *c* times more probable to minimize information loss than model M_j.

As an alternative to AIC, the **Bayesian information criterion** (**BIC**) [162] can be used, in restricted situations. It is defined as:

E51

$$BIC = k \cdot \log(n) - 2\mathcal{L}(\hat{\theta}|y)$$

As opposite to AIC, BIC *is not an estimate of relative KLD* (see Equation E45). Although BIC has a similar formulation respect to AIC, it has been developed independently. In practice, BIC assumes the existence of a true model and that it is present in the set of candidate models. In this framework, the aim of model selection is to find the true model, with a probability that approaches 1 as the sample size increases (such models are often called *dimension-consistent*). However, given these assumptions, BIC is more suited for relatively simple models (i.e.

where the true model could be reasonably reached) with large amounts of data, and in general for prediction purposes, rather than finding the best causal model underlying a complex (biological) processes. Furthermore, AIC tends to have better inferential properties, especially under the assumption that the true model is not in the candidate model set.

In the context of model selection, *where* to start is a key step. In absence of prior information, a good starting point could be the null model M_0 (i.e. the model with no other predictors than the intercept). Model building is done by adding predictors and diagnosing intermediate model performances. This approach could be misleading, since the result might depend on the specific path we adopted to build the model (to note: although the AIC does not depend on the specific predictors order, modeler’s choices can be influenced by a specific intermediate model fitting result). If the number of predictors is relatively small, one could try every possible linear combination. However, if we also consider interaction, moderation, and mediation effects, the space of possible models explodes with the number of explanatory variables included. As an alternative, we could try different reference models, built in their first formulation by experts in the field. In Chapter 4, we will use a strategy based on two alternative **reference models** that will provide (from different perspectives) the ***full representation*** *of our* ***biological system***. Here, “full representation” means the *most comprehensive model that can be provided as a direct expression of the biologist’s* ***working hypothesis****, given the available* ***experimental setting***. Hence, the reference will include the whole set of *potentially* causal predictors and their interactions. Here, to distinguish the generic term “interaction” with the concept of interaction in statistics, the latter will be called

“statistical interaction”, from now on. However, *given the exploratory nature of sequencing experiments, the reference model* (R_0) *is not necessarily the best one*. In the present work, three strategies were pursued for **model selection**, starting from R_0: (i) **partitioning**, (ii) **synthesis**, and (iii) **stratification**. The reference model (and its derived models), is diagnosed to calculate a set of values using: **residual** and **null deviance** (with correspondent degrees of freedom and p-values), **AIC**, **estimated effects** on the outcome (i.e. β, that in the logistic model are interpreted as **Odds-Ratios**, equivalent to e^{β}) and their **significance**. Reference assessment highlights the role of every predictor within the working hypothesis, to confirm or invalidate the initial assumptions. However, it is often possible to group predictors to describe single processes as part of process underlying the outcome. For instance, DNA damage can be either **favored** by the presence of specific conditions (e.g. transcript length, expression, and open chromatin), or **determined** by the presence of damage-specific profiles (e.g. the presence of γH2AX or repair factors). The former category of predictors is called **prognostic variables**, while the latter are called **indicator variables** (or simply indicators). We may decide to partition the reference into prognostic and indicator models to assess the relative contribution of the two components inside the working hypothesis, to assess variable effectiveness in predicting and determine the outcome (e.g. DNA damage). During partitioning one may observe the presence of predictors that never contribute to model fitting and decide to remove them. On the other hand, other predictors may vary in **effect size** (i.e. β, the magnitude of the effect of a predictor over the outcome) and **significance**. This could also highlight the presence of predictors with significant effect, but **high variability** (as revealed, for instance, by a high **confidence interval**).

Sometimes this **inflated variability** could be a sign of **multi-collinearity** among predictors, due to **confounding** effects (discussed below), or to a general **correlation structure** underlying the observed data, such as the correlation among gene expression levels due to possible gene-gene interactions (see Paragraph 3.5.8). However, sequencing data pre-processing, library size scaling, profile normalization, and interaction modelling should largely attenuate these effects, revealing only true biological variation (assuming a good experimental quality). Multi-collinearity occurs when two or more predictors are so highly correlated that they turn to be redundant (i.e. one predictor can be accurately predicted by another one through a linear transformation). This effect can be monitored using **variance inflation factors** (VIFs), for each predictor. VIFs can be estimated directly from the model, given the data. Let us consider the linear predictor function $\eta(j) = x_j^T \beta$. The smallest possible variance, $\text{var}_0(\beta)$, is the one related to the linear predictor having $j = 1$ (i.e. 1 estimated coefficient β):

E51

$$\text{var}_0(\beta_j) = \frac{\sigma^2}{\sum_{j=1}^{n}(x_{ji} - \bar{x}_i)^2}$$

However, for $j > 1$, the estimate variance is:

E52

$$\text{var}(\beta_j) = \frac{\sigma^2}{\sum_{i=1}^{n}(x_{ji} - \bar{x}_i)^2} \cdot \frac{1}{1 - R_j^2}$$

where R_j^2 is the determination coefficient (Definition E41) obtained by regressing the j-th predictor over the remaining ones.

The second factor of Equation E52 is the VIF for the *j*-th predictor, simply equal to the ratio between the estimate variance and its minimum value:

E53

$$VIF_j = \frac{\mathrm{var}(\beta_j)}{\mathrm{var}_0(\beta_j)} = \frac{1}{1 - R_j^2}$$

Since a certain level of correlation among predictors is generally not a problem, an accepted rule of thumb is to allow a maximum VIF of 4 (other, more relaxed, rules have been proposed, but given the high variability associated to sequencing data, it is advisable to use the most stringent cut-off). Besides being monitored, variance inflation cannot be directly corrected, since it is an intrinsic data feature, rather than a modeling issue or a noise source. Solutions include the possibility to remove extra predictors, or combining them into a single variable, for instance through PCA (see Paragraph 3.7).

Once partitioning and model diagnostics revealed the key predictors from the reference model, we may proceed to the “base” model synthesis. This include removal of the non-significant predictors (i.e. **reduction**), possible additional **normalization** and/or **correction** procedures to account for extra variability or biases, modeling **interactions**. These four steps compose what is described here as **model synthesis**. Reduction (step 1) is a quite immediate passage, since it removes extra predictors on the base of their significance and model AIC shrinking (provided an acceptable model performance). However, when the removed predictors are many (say > 2) one and belonging to different partitions of the reference, the possible lack of significance and/or model fitting under-performance could be due to missing interactions. Here, the elements of structural modeling will be presented, inside the **causal inference** framework, as a basis for the convergence of this field with **graph theory**. In Paragraph 4.2, a more

comprehensive **Structural Equation Modeling** (**SEM**) framework will be exposed. The first type of statistical relationship we consider is **statistical interaction** (Figure 15*a*). Statistical interaction is a **non-additive** influence of two variables (*x, z*) on a third one (*y*):

E53

$$y = c + \beta_1 x + \beta_2 z + \beta_3 xz + \varepsilon$$

where *c* is the intercept, β_j are the direct effects, and ε is the error term. Interaction effects may be masked if the model do not specify them explicitly, i.e. the main observed effect of a variable over the outcome could be due to an interaction effect. For this reason, explicitly modeling an interaction between predictors could improve both single predictors and the overall model fitting. The **mediation model** is characterized by a mediator variable *t* generating an indirect relationship between the independent variable *x* and the outcome *y* (Figure 15*b*):

E54

(a) $$y = c_1 + \beta_1 x + \varepsilon_1$$

(b) $$y = c_2 + \beta_2 x + \gamma t + \varepsilon_2$$

(c) $$t = c_3 + \beta_3 x + \varepsilon_3$$

where β_j are the direct effects of the independent variable on the outcome and the mediator variable, and γ is the effect of the mediator on the outcome. The **moderation model**, instead, takes into account a variable *z*, called the **moderator**, that affects the relationship between the independent variable and the outcome, by enhancing, silencing, changing the influence or the direction of the causal relationship (Figures 15*c,d*). Moderation can be evaluated by estimating the effect from the interaction (i.e. the β_3 parameter in Equation E53). Moderation can be also viewed as the estimation of the $x \rightarrow y$ effect change among different levels of a third variable *z*. Moderation and

mediation can be also combined in different ways (Figure 15*e*). A general framework for **moderated-mediation** can be modelled as follows [167]:

E55

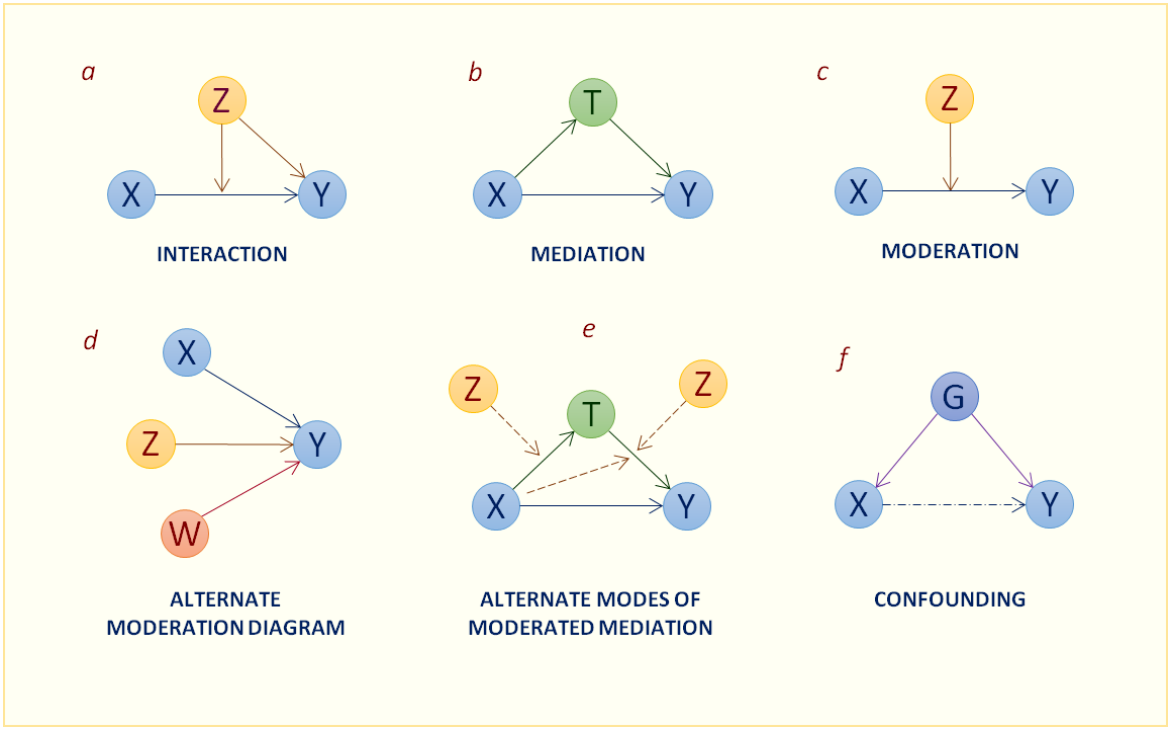
(a) $$y = c_1 + \beta_1 x + \beta_2 z + \beta_3 xz + \varepsilon_1$$

(b) $$t = c_2 + \beta_4 x + \beta_5 z + \beta_6 xz + \varepsilon_2$$

(c) $$y = c_3 + \beta_7 x + \beta_8 z + \beta_9 xz + \gamma_1 t + \gamma_2 tz + \gamma_3 xt + \gamma_4 xtz + \varepsilon_3$$

Figure 16 present the various **path diagrams** for each of the abovementioned relationships. One last effect, that is however not desired, is **confounding** (Figure 15*f*). It occurs when an external factor, not considered inside the model, influence both predictors and the outcome. This interaction may alter, or completely mask, true effects. While there is no general prior rule to detect unobserved external factors causing confounding, this effect can be cancelled out using a **stratified model**. Confounding can be due to various known factors, including **environment** (e.g. exposure to physical-chemical agents), **behavior** (e.g. dietary habits), **phenotypical traits** (e.g. sex or age), **diseases** (e.g. interacting diseased phenotypes), and **unobserved biological processes** (e.g. cell cycle phase). While some of these factors can be categorize in a relatively simple way (e.g. sex, age, behavior), other, specifically related to bio-chemical disorders, are more difficult to split into separate levels. In these cases, unsupervised methods, including **PCA**, **clustering**, and **mixture models**, may provide at least a series of guidelines.

Figure 16. Different types of basic statistical relationships. *Path diagrams of five basic types of statistical relationships are shown in the picture below: (a) interaction, (b) mediation, (c, d) moderation, (e) mediated moderation, and (f) confounding. X is an independent variable, and Y is the outcome. Z is the moderator variable, T is the mediator, and G the confounder. In the alternate moderation diagram, the interaction with the moderator is explicitly represented by W = XZ. Mediation and moderation could be combined in different alternative ways, as indicated by the brown dashed arrows (e). The X→Y relationship could be masked by a cofounding effect due to a variable G, representing an influence that is external to the model (e.g. an environmental factor). Model stratification can be adopted to cancel out the confounding effect.*



3.5.6. Bootstrap-based sampling and model validation

The whole conceptual and statistical system we discussed till now is oriented to the creation of a faithful model of biological variation, that could arise directly from data, with only minimum impact of acquired (and possibly misleading) knowledge. Having the best possible model is therefore the main, although only the first, step. This best model will be given as granted in every further analysis, thus having a scale for comparing suitable alternatives is critical. Given a set of *m* suitable models M_j (for *j* = 1, 2, ..., *m*), the **Akaike weights** for each *j*-th model are defined as:

E56

$$w_j = \frac{\exp\left\{-\frac{1}{2}\Delta_j\right\}}{\sum_{h=1}^{m}\exp\left\{-\frac{1}{2}\Delta_h\right\}}$$

such that $\sum_j w_j$ = 1. Akaike weights are therefore a scaled version of Δ_j. In this way, w_j is the weight of evidence for M_j to be the **KL-best model** (see the previous paragraph), among the *m* considered ones. Naturally, adding, removing, or changing the model set M will imply re-computing w_j values. The question is now:

Is there any other independent method to rank models than w_j?

The solution comes from **model-based sampling theory**. The key basic assumption from which we start is the same we adopted in the previous paragraph: *there exists some* ***conceptual probability distribution*** *representing the* ***unknown reality function*** $f(y)$, *that generated the observed data set* y. On the other hand, if we consider our data set as actually deriving from f, assuming a sufficiently large sample size, we can consider our dataset as a good approximation of the space of all possible observations. In other terms, we assume y as

being our **ground truth**. The general idea is to **random sampling** B new data sets y_b from y (for b = 1, 2, ..., B), with replacement (i.e. observations can be duplicated). This method, called **bootstrap**, is a kind of **Monte Carlo method** frequently used in statistics for robust (i.e. not dependent upon a distribution) estimation of sampling variance, standard errors, or confidence intervals. Given the above definition of ground truth, we may consider the bootstrap samples y_b as a proxy for B real-world samples drawn from f.

Since bootstrapping is distribution-independent, it can be faithfully used either when the theoretical statistic distribution is complex or unknown, or as an alternative indirect method to assess the properties of the distribution underlying data. Therefore, bootstrap can be also used to have an independent way of assessing model performances. This task can be achieved through the estimation of **model selection frequencies** (s_j). Suppose having *m* models M_j (for *j* = 1, 2, ..., *m*), each one representing a possible KL-best approximation of the real biological process. For each data set y_b, an $AIC_j(b)$ value for every model is calculated and a KL-best model $\hat{M}_j(b)$ is found. Reasonably, the *j*-th best model at each iteration is not expected to be always the same. After B iterations, every model will have a specific selection frequency $s_j = B_j/B$. Selection frequency s_j can be used to rank model performance, in the same way w_j is used.

Given the causal modelling framework depicted in the previous paragraphs, s_j is used as a collateral way to test model performances, together with to w_j. Besides implementation details and validation procedures, the aim of **model validation** is to assess the **predictive accuracy** of the best model(s), independently from the causal inference procedure (typically performed in the GLM/SEM framework)

adopted to build it. It is possible to model an alternative modelling strategy, based on bootstrap empirical distribution, comprised in the same information theory provided by KLD. **Random Forests** (**RFs**) are an **ensemble machine leaning** based on **decision trees**. In a typical decision tree algorithm, every *i*-th observation is represented by a multidimensional vector (v_i). Precisely every data entry can be defined by two components: the **features vector** $v_i = x_{i1} + x_{i2} + ... + x_{ih}$ and a **target variable** y that must be reached. When y is categorical, the RF algorithm is called a **classifier** (RFC), while for continuous y it is called **regressor** (RFR). Feature must be decided *a priori*, on the base of the modeler's preference, to give the best possible description of data. In decision trees, a **top-down** divisive strategy is adopted: at each tree split, the algorithm finds the data separation that maximizes an **information gain criterion** (or conversely, minimizing an information loss criterion). In most cases, this criterion coincides with the KLD (see Equation E45). In general, the **expected information gain**, due to the difference from the initial data configuration to the current one, is measured in terms of change in information entropy:

E56

$$I(T,k) = H(T) - \sum_{a \in v(k)} \frac{|x \in T | x_k = a|}{|T|} \cdot H(|x \in T | x_k = a|)$$

where $H(\cdot)$ is the entropy function, T is the training set, composed by data entries in the form (v, y) where v is the features vector and $v(k)$ is the value of the *k*-th feature x_k, and y is the target variable. As an instance of $H(\cdot)$, in the case of a binary outcome (i.e. {0, 1}), the entropy function is given by:

E57*a*

$$H(T) = -P(y = 1) \cdot \log[P(y = 1)] - P(y = 0) \cdot \log[P(y = 0)]$$

maximized for $P(y = 1) = P(y = 0) = 0.5$, and minimized when either of the two is 1 (and the other will be 0). In general, the **entropy function** has the form:

E57*b*

$$H(T) = -\sum_{k=1}^{m} p(x_k)\log(x_k)$$

In the context of decision trees, Equation E56 can be interpreted as the difference between the entropy before and after the tree split, yielding the information gain at that split. In RFCs (the type of RFs we will use in Chapter 4), maximizing information gain at each split will lead to an update of the probability for a certain data point to belong to a specific class, until the final split (tree leaves). Therefore, every leave will be associated to a probability of belonging to a specific class. These characteristics of decision trees allow a natural integration with the forward analysis stage. In particular, 5 decision trees properties are required to accomplish this integration: (i) data structure, input model architecture, and outcome type can be explicitly and easily represented as input for the decision tree algorithm, (ii) decisions are clearly traceable through the binary structure of the tree, given an entropy function, (iii) robustness (i.e. the method performs well in different modelling contexts, almost independently from the particular assumptions under which data has been generated), (iv) the decision tree model can be explicitly assessed through statistical testing, and (v) the output of the decision tree model is easily interpretable in probabilistic terms.

Beside the NP-completeness of an optimal decision tree solution, that can be compensated by heuristics [167], one non-negligible problem affects decision trees as they grow deeply: huge decision trees tend to be affected by high variance. In the extreme case, the splitting procedure can lead to a perfect classification of every element of the

training set, up to the existence of a single example (i.e. a single data vector) for each category, in each leaf. This would make a tree perfectly unbiased, since every node can be correctly classified. However, the variance of such model would be unreasonably large (i.e. the model is completely over-fitted to the training set). To avoid variance explosion and over-fitting problems, RFs are based on a random bootstrap-based generation (called bootstrap aggregating or ***bagging***) of multiple decision trees for the same training set. At each split, random sampling form the feature space (i.e. **random subspace sampling**) allow the algorithm not to over-train on specific features. Therefore, at each iteration, only a subset of features is used and, from them, a new training set is generated by bootstrap.

CHAPTER 4

Applications in Genome Research

Odio la vile prudenza che ci agghiaccia e lega
e rende incapaci di ogni grande azione,
riducendoci come animali che attendono
tranquillamente alla conservazione
di questa infelice vita senz'altro pensiero

Giacomo Leopardi
Letter to his father Moraldo, 1819

4.1. Genomic instability in Breast Cancer

Transcription in eukaryotes is a complex process involving intensive enzymatic activities in enhancers, and the promoter region of active genes. These activities comprise RNA Polymerase recruitment and activation. Recent studies [12-16], revealed a strong connection between transcriptional activity and **DNA repair**, suggesting a high frequency of programmed **DNA damage** correlated with **transcriptional activation**. It has been reported how nuclear receptors in MCF7 breast cancer cells requires topoisomerase II-beta (TopoIIb) binding to the target promoters to generate a transient **double-strand DNA break** (**DSB**). The ability of a cell to repair this physiological damage is then critical to maintain genomic stability, and avoiding

chromosome aberrations including **genetic fusion**. Mechanical torsional stress over the DNA double helix induced by the physiological activity of the RNA polymerase II Ser5p (POLR2A, abbreviated “Pol2” as a variable name), could be one of the key inducers of TopoIIb activity. Therefore, Pol2 localization, abundance, and activity could influence DSB susceptibility besides the absolute transcriptional activity of a gene. To test these potentially causative factors we studied them in the modeling framework described in the previous chapter, using in MCF10A human mammary gland cells as reference biological system [180].

Diploid mammary-epithelium cells MCF10A-AsiSIER [180] were used to produce HTS samples for this study. In these cells, the restriction enzyme AsiSI induces DSBs, under Tamoxifen induction. To identify the digested AsiSI sites, we performed ChIPseq experiments using antibodies against DSB sensing and repair factors, including NBS1, γH2AX, XRCC4 and RAD51. AsiSI-digested sites were mapped within open chromatin domains (defined by H3K4me3, H3K4me1, and H3K27ac profiles), in promoters and active enhancers. In addition to Tamoxifen-induced AsiSI-associated damage, we also detected endogenous DSBs (i.e. non-AsiSI in absence of Tamoxifen induction), using the pipeline described in Paragraph 4.1.3. Endogenous damage shows the simultaneous presence of a significant BLISS enrichment over the background noise and sensing/repair factor (i.e. NBS1, XRCC4, and RAD51) enrichment. Notably, we found no enrichment for DSBs within gene bodies (Fisher’s exact test $p=0.172$, $OR=0.96$), while a strong DSB enrichment was found in promoters (Fisher’s exact test $p<2.2e-16$, $OR=2.62$), with a total occurrence of 636 DSBs in 627 promoters (i.e. the fragile promoters). Remarkably, fragile promoters and their gene bodies were not enriched in common fragile sites (CFS).

The remaining Tier1 DSBs map inside intergenic regions. Moreover, 833 intra- and intergenic (non-promoter) DSBs map within 815 active enhancers (Fisher’s exact test p<2.2e-16, OR=32.2), calculated as H3K27ac/H3K4me1-enriched regions. Overall, in epithelial mammary cells, promoters and enhancers tend to be significantly enriched in DSBs.

4.1.1. Modelling the association between risk factors and DNA damage

To model causal association among risk factors and DNA damage at promoters, we assumed that DSBs are point-source events that involve the promoter region (TSS ± 2.5kbp). We will see in the Paragraph 4.1.3 how this (biologically correct) definition is indeed induced from a more complex statistical analysis of **BLISS** [160] genomic signal distribution. To have a comprehensive picture of DNA damage risk, we measured signal density for 8 factors: **gene expression** (measured using both **RNA-seq** and **GRO-seq**), **Pol2**, **H3K4me1** (Me1), **H3K4me3** (Me3), **H3K27ac** (Ac), **Nbs1**, **Xrcc4**, and **Rad51**. We also considered **gene length** (L), where only the canonical (i.e. longest) transcript is used per gene, since measured transcription levels do not consider isoforms. Gene expression is normalized to **TPM** (transcript per million uniquely mapped reads), to have the same normalized unit between RNA- and GRO-seq counts. TPM for the *i*-th gene is obtained by dividing its tags count (n_i) by the sum of counts over all genes, normalized by gene length, per million uniquely mapped reads, i.e. $TPM_i = 10^6 \cdot (n_i/L_i)/\sum_i(n_i/L_i)$. However, while RNA-seq tag count summarization has been done considering exons only, GRO-seq measures the effective amount of transcription over all the transcript length (including both exons and introns). We denoted with T_R and T_G

the RNA- and GRO-seq based expression levels, respectively. Overall, we have both RNA- and GRO-seq expression levels for N_G = 20396 genes. Every measured variable has been converted to logarithm in base 2 of tag counts (plus 1 as pseudo-count to avoid log of 0). Every ChIP-seq profile have been down-sampled (by randomly remove reads from the input alignment BAM files) to the lowest library sized sample, to account for signal magnitude differences.

Variables can be grouped according to their characteristics. Nbs1, Xrcc4, and Rad51 are repair factors, that accumulates in case of damage, and therefore are used as **indicator variables**. Indicator variables should always have a significantly positive effect in damaged promoters, and therefore can be assumed as **control parameters**. Since Xrcc4 and Rad51 are specific for two different repair mechanisms, i.e. **non-homologous end joining** (**NHEJ**) and **homologous recombination** (**HR**) respectively, they may reveal differential repair mechanisms for different interaction patterns. Histone markers are instead **contextual variables**, in the sense that they predict the **chromatin state** (i.e. active or repressed) at each genomic site (i.e. for each promoter). Finally, L, T_R, T_G, and Pol2 are **transcription-related variables**, that provide information on measured gene expression levels (note that given the definition of TPM, gene length and expression levels have been decorrelated). Together, contextual and transcription-related variables represent the set of our **prognostic variables**, that can be used to blindly predict promoter susceptibility, with no prior assumptions on their role in DNA damage formation.

Based on the BLISS signal significance (see Paragraph 4.1.3), we divided DSB data into 3 tiers, with decreasing confidence: **Tier1** (high confidence DSBs, N_{Tier1} = 8132), **Tier2** (low confidence DSBs, N_{Tier2} =

42304), and **Tier3** (very low confidence, N_{Tier3} = 38409). For causal inference purposes, we used Tier 1 DSBs only as **true damage events** (i.e. our ground truth). Among the N_G reference genes, N_P = 627 promoters (corresponding to 636 DSBs) were damaged at Tier 1 (i.e. cases), while N_0 = 2064 promoters resulted completely DSB-free (i.e. free from DSBs of any tier; i.e. controls). BLISS signal could be used also as a continuous variable, although several problems prevented us from adopting this choice, including strong signal heteroscedasticity, high local background levels, and possible interference from restriction enzyme activity that may cause sudden signal drops. Influenced also by these biases, BLISS signal resulted not correlated with gene expression (r^2 = 0.012). However, considering DSBs as discrete point-source events, we may assess if there is skewness respect to their expected occurrence and/or trends in their proportion, considering intervals of expression levels. We repeated every analysis considering either T_R or T_G.

We defined four expression strata: (i) **Class1**, including genes having TPM < 1 (N_1 = 82); (ii) **Class2**, including genes having expression levels under the lower quartile (i.e. poorly-expressed genes; N_2 = 99); **Class3**, including genes expressed above the lower quartile but below the top 10% expression (N_3 = 366); and (iii) **Class4**, including genes with very high expression level (i.e. top 10%; N_4 = 80). The general idea is to distinguish poorly-expressed from active genes, with a special attention to the expression extrema behavior. Since GRO-seq showed a stronger association with DNA damage, only T_G-related descriptive statistics will be reported. In general, RNA-seq showed similar but less significant results, mainly due to the larger variability affecting RNA-seq samples and therefore a poorer RNA-seq discriminant accuracy between cases and controls.

Observed vs. expected frequencies showed an increasing trend for classes 1 to 4 (p-value < 0.01): 0.51, 0.85, 1.21, and 1.71. The expected frequency of damaged promoters for the *j*-th class is calculated as $c_j N_D / N_G$, with c_j being the total number of genes over N_G for class *j*. This result showed us which was the transcriptional context that either favors or impede the DSB occurrence at promoters. However, to test the association between DSB occurrence and expression it is necessary to compare per-class cases proportion against controls.

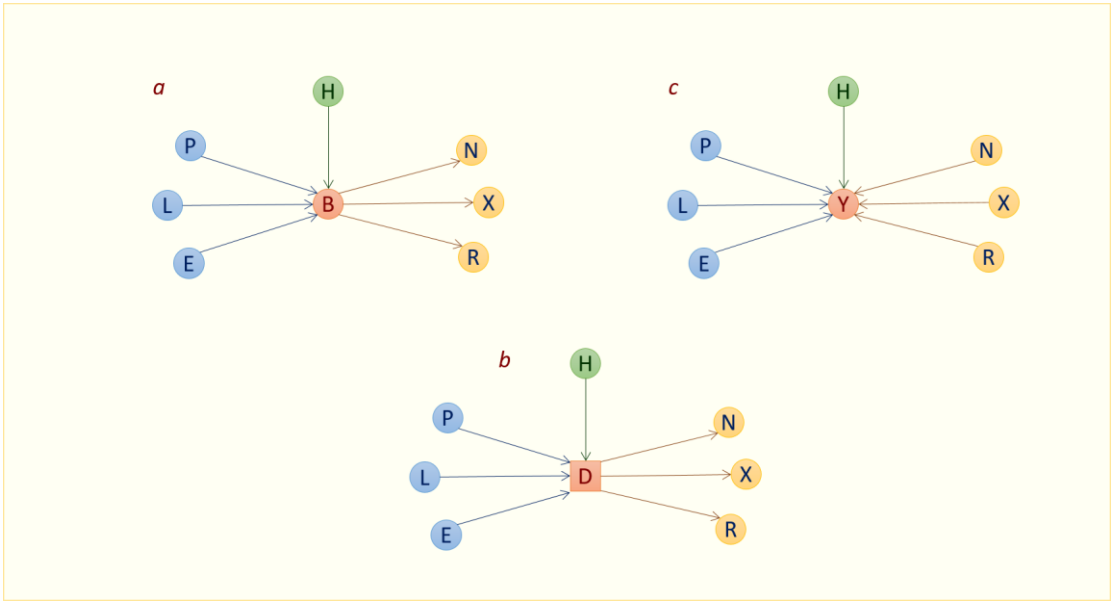
Table 8. DSB data grouped by expression class. *DSB data is grouped in 4 transcription classes (1 to 4) corresponding to increasing expression levels. Cases (Y1) are defined as Tier1 DSB-positive promoters, while promoters free from DSBs of any tier define controls (Y0).*

		Class				
		1	2	3	4	
Variable	Y1	82	99	366	80	Cases
	Y0	429	515	940	180	Controls
	T	511	614	1306	260	Total

Using a Pearson's χ^2 test, we detected a significant association between expression and DNA damage at promoters (χ^2_{ASSO} = 57.1641, 3df, p = 2.37E-12). We also found a significant trend in DSB proportions, increasing with classes 1 to 4 (Armitage's trend test χ^2_{TREND} = 47.8488, 1df, p = 4.6E-12). This means that the higher the expression, the higher the proportion of damaged promoters. Anyway, this demonstrates only that a higher expression is associated to a higher production of DSBs, without explaining how and how efficiently DSBs are repaired. To assess the capability of a linear model to explain this variability, we calculated the residual chi-squared (χ^2_{RES}) as the difference between the association and trend chi-squared values. Assuming a restrictive 99% confidence interval, with df = df_asso – df_trend = 2, we obtained a tabulated χ^2_{REF} = 9.2103, that is only

slightly less than the residual one ($\chi^2_{RES} = \chi^2_{ASSO} - \chi^2_{TREND} = 9.3153$). Although $\chi^2_{RES} > \chi^2_{REF}$, there is only a minimal difference at a significance level of 0.01. The small residual component indicates that a linear model can be a good approximation of DSB-transcription association, although there could be some small unexplained variability due to non-linear interactions. However, we can easily monitor the proportion of unexplained variance through deviance analysis.

Figure 16. Three possible model for DNA damage at promoters. *Path diagrams of three possible strategies to model DNA damage. Conceptually, DNA damage (red) can be modelled a point-source event that is caused by transcription-related factors (blue) and contextual variables (green), and causing the accumulation of repair factors (gold). Diagram “a” models this explicitly, using BLISS signal as a proxy for repair factor accumulation. Diagram “b” tries to overcome BLISS limitations by modelling damages as a latent class C. Since DSBs have been genome-wide called, obtaining three confidence tiers, we can model them as the observed damage outcome Y. In this case, repair factors are modelled as indicators of damage, measuring the risk that a true damage is found in a specific promoter. Variable keys are: P = Pol2, L = gene length, E = gene expression (either RNA-seq or GRO-seq), H = histone marks Me1, Me3, Ac (in the diagram they are reported as a unique variable only for visualization purposes, but they are separated in the applied model), N = Nbs1, X = Xrcc4, R = Rad51, B = BLISS continuous signal, D = latent damage class, Y = damage outcome.*



There are at least three ways to model DNA damage through GLMs (Figure 16). In the first two (Figure 16*a*,*b*), prognostic factors cause a damage, that results in the accumulation of repair factors (i.e. the outcomes). In diagram *a* damage is represented as a continuous variable (i.e. BLISS signal). However, due to the abovementioned biases this model could suffer severe problem with parameter estimation and significance assessment. Diagram *b* overcomes these problems by modelling DNA damage as a latent class *D* (i.e. a categorical binary variable). This could help in improving the empirical threshold use to rank DSB events for ground truth definition (see Paragraph 4.1.3). However, since true damage events have been **called** (i.e. **genome-wide tested**) and grouped in **confidence tiers**, adding a latent class could be an unnecessary complication. Hence, the most reliable way of modelling DNA damage, is to assume that DSBs are the observed **damage outcome**, caused by the accumulation of **risk factors** and **contextual factors** (i.e. markers of open chromatin). Furthermore, **repair factors** in this model are assumed as **indicator variables**, and their causality is inverted respect to diagrams *a* and *b*. This interpretation is different from the previous model, in the sense that indicators are assumed as a **proxy measures of damage intensity**, substituting the highly uncertain BLISS signal. This new interpretation has other two advantages: (i) indicators provide a further evidence of damage, allowing a more **reliable classification**, and (ii) different indicators are involved in different repair mechanisms, allowing us to detect possible **patterns of damage repair**. A further predictor, that has not been reported in Figure 16, is the **class variable** G. This variable id somehow different from the others, not only because it’s a factor of 4 levels (i.e. Classes 1 to 4), but because of its properties. We showed how G is associated with DNA damage although expression is

not correlated with it. In fact, gene expression levels define trends in data respect to DNA damage, but we still observe a consistent fraction of highly-expressed genes having DSB-free promoters, as opposite to poorly-to-not-expressed genes with damaged promoters. This fuzzy separation is the consequence of a **confounding effect** of expression over the **underlying true damage mechanism**. In other terms, there is a variable in the model (i.e. Pol2) that both influences gene expression levels and DSB occurrence at promoters, but not a direct causal relationship between gene expression and DNA damage. Still, expression is reported as an independent variable, to test the hypothesis that transcript levels could cause DNA damage. Instead, we introduced G as a stratifying variable to remove the confounding effect. Since G is related to T_G, modelling G inside the linear predictor as a categorical variable is not advisable, since it may cause **multicollinearity** (anyway monitored using **VIFs**). To avoid this effect, our aim was to produce four different damage models, underlying **four candidate DNA damage-associated mechanisms**. This model was implemented as a **logistic model** with binary damage outcome:

Y(*k* = 1): *damaged promoter*; *Y*(*k* = 0): *damage-free promoter*

Before starting with model testing, we planned a **hierarchical model selection** strategy (see Paragraph 3.5.6). We first defined two competing **reference models**: R_0 and R_1. The linear predictors for the two models are respectively defined as:

E58

(R_0)

$$Y_k \sim L + T_h + Pol2 + Me1 + Me3 + Ac + Nbs1 + Rad51 + Xrcc4 + G$$

(R_1)

$$Y_k \sim L + T_h + Pol2 + Me1 + Me3 + Ac + Nbs1 + Rad51 + Xrcc4$$

where T_h can be either T_G or T_R. The ~ symbol, instead of =, indicate proportionality since the parameters (i.e. β coefficients) have been omitted from the formulas for visualization purposes. These coefficients will be estimated during model fitting and their exponential e^{β} will be interpreted as odds-ratio (OR). Models R_0 and R_1 only differ because the latter lacks the class term G. Reference models have only three purposes: (i) competing each other, to test the effectiveness of G in improving modelling, (ii) providing a lower bound for the AIC ranking, and (iii) provide a basis for model partition to independently assess the role of expression-related, prognostic, and indicator variables. From the first comparison, R_0 resulted the best model (AIC_{R0} = 1817.3 vs. AIC_{R1} = 1957.5). We used GRO-seq, since T_G improved the model respect to T_R. We then started fitting partitions of model R_0:

E59

$$(M_0) \qquad Y_k \sim L + T_G + G$$

$$(M_1) \qquad Y_k \sim L + T_R + G$$

$$(M_2) \qquad Y_k \sim L + T_G + Pol2 + Me1 + Me3 + Ac + G$$

$$(M_3) \qquad Y_k \sim L + T_G + Nbs1 + Rad51 + Xrcc4 + G$$

being, respectively: (i) **expression model**, (ii) **alternative expression model**, (iii) **prognostic model**, and (iv) **indicator model**. None of these partitions performed better than R_0, although M_2 and M_3 scored closer to it (AIC_{L2} = 1994 and AIC_{L3} = 1838.1). The comparison between M_0 and M_1 respect to R_0 confirmed that GRO-seq is more informative than RNA-seq, at least in our system (AIC_{L0} = 2349.2 and AIC_{L1} = 2414.8). Both in R_0 and M_2, histone marks Me3 and Ac did not have significant effects. As resulted from deviance analysis, every model

significantly improved respect to null model, and therefore significant predictors significantly contributed in explaining the outcome. We generated the **DNA damage model**:

E60

(M_4)

$$Y_k \sim L + T_h + Pol2 + Me1 + Nbs1 + Rad51 + Xrcc4 + G$$

That provided a first gain of fit respect to R_0 (AIC_{L4} = 1814). This improvement confirmed the validity of the selection procedure. This model represents the synthesis step discussed in Paragraph 3.5.6. As expected, var(G) resulted slightly inflated, and therefore we stratified M_4 in other 4 independent models: M_5 (Class1), M_6 (Class2), M_7 (Class3), and M_8 (Class4). Initially, every class-specific model has the same formula as M_4, by removing predictor G:

E61

(M_{init}) $\quad Y_k \sim L + T_h + Pol2 + Me1 + Nbs1 + Rad51 + Xrcc4$

The only initial difference between them is the portion of data onto the model is fitted (i.e. class-specific). Being fitted on different subsets, these models cannot be compared neither each other nor against M_4, according to AIC. On the other hand, we used AIC and deviance analysis to assess the improvement of the initial model after a series of **reduction** steps.

For Each *j* in {Class1, Class2, Class3, Class4}:

(i) The initial common model M_{init} is fitted on the *j*-th dataset

(ii) The fitted model is tested against the null model

(iii) If (ii) is significant: the AIC value is stored

Else: raise an error: the model failed to explain the true biological process.

(iv) If exists a variable with p-value ≤ 0.05:

a. Variables are ranked according to p-value: the last one is removed

b. The new model is fitted:

If the AIC of the new model is lower,

then restart from (ii)

Else: restore the non-significant predictor and exit.

(v) The final reduced model M_j is tested for multicollinearity and residual variance.

(vi) Exit.

Therefore, AIC ranking is performed internally to each class. The final estimates are reported in Table 9. The final class-specific models were:

E62

(M_5) $Y_k \sim L + T_h + Pol2 + Nbs1 + Rad51 + Xrcc4$

(M_6) $Y_k \sim L + T_h + Xrcc4$

(M_7) $Y_k \sim L + Me1 + Pol2 + Xrcc4$

(M_8) $Y_k \sim Pol2 + Xrcc4$

None of the models resulted variance-inflated, and no significant residual (i.e. unexplained) variance was left out. We also tried to model Pol2-length, Pol2-expression, and Pol2-repair factor interactions, with no improvements in model fitting.

Table 9. Estimates from the final stratified logit models for DNA damage occurrence in MCF10A cells*. Model rank and filtering was based on the AIC optimization and deviance analysis. The final AIC values for M4-M8 are reported, although they are not directly comparable. Values for fields L to Xrcc4 are the OR from logistic model fitting. The color code reflects the significance of the estimated OR: very strong ($p < 0.0001$; cyan), strong ($p < 0.001$; green), good ($p < 0.01$; gold), weak ($p < 0.05$; red). White boxes with a --- symbol indicate non-significant effects. The Class field specifies which data portion has been fitted to the initial common model; ALL indicates the initial model performances, for the whole dataset.*

Model	AIC	L *	TPM *	Me1	Pol2	Nbs1	Rad51	Xrcc4	Class
M4	1814	1.82	1.46	1.11	1.32	1.18	1.21	2.92	ALL
M5	194.17	1.45	6.03	---	1.99	1.83	2.48	---	1
M6	370.54	2.21	1.53	---	---	---	---	1.71	2
M7	913.83	2.04	---	1.16	1.29	---	---	7.17	3
M8	234.05	---	---	---	2.05	---	---	4.54	4

* L = gene length in log_2bp; TPM = GRO-seq expression in log_2TPM. All the other predictors units are log_2TagCount.

Results shown in Table 9 demonstrate what has been initially hypothesized. Firstly, **gene length** and **Pol2** are the variables having the largest effect show that the mechanisms causing DNA damage are not directly consequence of expression levels, but rather collateral to it. A simple explanation is that the torsional stress induced by Pol2, especially in longer genes, may enhance both transcription and transcription-associated DSB formation (e.g. due to TopoIIb activity). This explains the confounding effect of gene expression over the other model relationships. Secondly, model estimates may deeply change among classes. Repair factors show alternative influence of either Nbs1/Rad51 (Class1) or Xrcc4 (Classes 2-4), possibly due to predominance of specific repair mechanisms. On the other hand, Pol2 effect is significant in Classes 1, 3 and 4, but not in Class2. Interestingly, also Xrcc4 levels are lower in Class 2 than Classes 3 and 4. This also suggests also that gene length and Pol2 have synergic effects on DSB formation. In other terms, DNA damage occurs both for longer genes in presence of poor-to-moderate Pol2 activity, and shorter genes when Pol2 activity is very high. Notably, Class 3, where high transcriptional

activity is combined with longer gene lengths, shows also a higher Xrcc4 association with the outcome, suggesting that this class of genes could be the most susceptible to genomic instability.

It is possible to validate the damage model (M_4), independently from the logit model and MLE estimation, but remaining within the information theory framework provided by KLD (see Paragraph 3.5.7), thus producing comparable results. Therefore, to test the prediction accuracy of the damage model we performed a **4-fold cross-validation** using a **RFC** [169], trained on 10000 bootstrap samples from the original dataset, and randomly choosing 3 variables per split (i.e. the squared-root of the number of predictors, rounded to integer). A randomization step preceded the cross-validation to produce 4 homogeneous independent sub-sets. Firstly, data rows were shuffled, to obtain a random distribution of chromosome identifiers. Then, from the shuffled data, 4 different data frames were generated. Data frame generation follows a pseudo-random sampling procedure, with two fixed parameters: **class proportion** γ, and **damage fraction** δ. These parameters are calculated from the initial population and correspond to the proportion of genes per class (γ_1 = 19% Class1 genes, γ_2 = 22.8% Class2 genes, γ_3 = 48.5% Class3 genes, γ_4 = 9.7% Class1 genes), and the percentage of damaged promoters in the whole population (δ = 23.3%). The pseudo-random sampling was repeated until the error from the two parameters dropped below 2%. On average, the 4 sub-sets have 2018 training examples to be validated against 673 test promoters.

After the 4-fold cross-validation RFC cycles we calculated the **mean variable impact** (MVI) and the **prediction accuracy** (A). At each of the 4 iterations, the portion of dataset left out by the bootstrap sampling is used to calculate the **Gini impurity** [168] decrease, averaged over all

trees, for each variable. The **mean decrease** in **node impurity** (or mean decrease Gini impurity, MDG) is a measure of how well every single predictor performs (on average) during each iteration. We averaged MDG values (i.e. the MVI) over the 4 cross-validation iterations to rank single predictor classification performance. As expected, confirming the logistic model results, the best performing variables were: gene length (MVI_{Length} = 185.83), Xrcc4 (MVI_{Xrcc4} = 150.46), and Pol2 (MVI_{Pol2} = 93.5), followed by Nbs1 (MVI_{Nbs1} = 72.46), GRO-seq (MVI_{GRO} = 68.75), Me1 (MVI_{Me1} = 64.77), and Rad51 (MVI_{Rad51} = 62.22). Predictive accuracy is instead calculated as the average of sums of the TP and TN over all the predicted values, for each of the *k* iterations (i.e. the correct predictions over the data set size): $A = \sum_i (TP_i + TN_i) / kN$. Using all the predictors from M_4, we reached a predictive accuracy of 87.33%, decreasing to 83.36% using M_4 prognostic variables only (i.e. excluding Nbs1, Rad51, and Xrcc4).

After validation, we assessed the individual accuracy for every M_4 predictor. We calculated both class-specific and overall thresholds maximizing case/control separation for each predictor (namely the **optimal cut-point**) [170]. The AUC (area under the receiving operating characteristic curve) generated by the threshold was used as a measure of **discriminant accuracy** (i.e. the accuracy of the threshold maximizing the discrimination between cases and controls). The classical criterion used to calculate the optimal cut-point is the **Youden index** [170]. However, since its formulation, other criteria have been proposed, with different performances on the base of the experimental data and the study design. These include methods based on sensitivity and specificity (including the Youden index), maximum χ^2, minimum p-value, and prevalence-based criteria. Since the general nature of our application, we compared different criteria, using as

baseline the Youden index. We selected the criterion that produced the optimal cut-points with the **best improvement** in terms of AUC, respect to the Youden's one, and showing the **lowest between-classes variability** (i.e. minimize the sum of squared differences between cut-points from different classes). We finally selected the **sensitivity-specificity equality** criterion [170].

Table 10. Optimal thresholds for DNA damage-associated predictors. *The table shows the optimal thresholds for DSB-associated predictors, based on sensitivity-specificity equality criterion, that demonstrated to be much more accurate than Youden index (reported as a bad genomic indicator reference). Each threshold is calculated both for groups and for the overall dataset. AUC confidence intervals (95%) determine the quality of the discriminant variable. In general, discriminant variables having AUC > 70% are considered as acceptable, and optimal if AUC > 80% (both in gold). H3K4me1 has not been reported, since its role is only slightly determinant in the model, and not discriminant.*

Variable*	Youden	Threshold	GroupAUC	Low_CI	Up_CI	Quality	Global	AUC
Len_1	16211	**14892**	**71,00%**	64,20%	**77,80%**	Fair	**15493**	**73,80%**
Len_2	58496	**19759**	**80,70%**	**75,30%**	**86,10%**	Optimal		
Len_3	26015	**19186**	**79,30%**	**76,50%**	**82,10%**	Good		
Len_4	8629	4034	58,20%	50,50%	65,90%	Bad		
TPM_1	0,99	0,99	67,40%	61,50%	**73,30%**	Poor	7,45	60,80%
TPM_2	4,32	5,07	62,90%	56,70%	69,00%	Bad		
TPM_3	19,56	28,64	53,30%	49,90%	56,80%	Bad		
TPM_4	232,49	166,80	60,70%	52,70%	68,70%	Bad		
Pol2_1	6,820	**7,948**	**90,60%**	**87,90%**	**93,30%**	Optimal	**9,262**	**75,70%**
Pol2_2	8,811	8,827	66,80%	61,10%	**72,50%**	Poor		
Pol2_3	9,877	**9,781**	**76,70%**	**73,90%**	**79,40%**	Good		
Pol2_4	11,031	**10,945**	**80,30%**	**74,30%**	**86,30%**	Optimal		
Xrcc4_1	7,644	**7,768**	**90,90%**	**88,20%**	**93,70%**	Optimal	**9,047**	**79,90%**
Xrcc4_2	8,267	9,000	66,30%	60,80%	**71,80%**	Poor		
Xrcc4_3	9,267	**9,134**	**83,00%**	**80,50%**	**85,50%**	Optimal		
Xrcc4_4	9,674	**9,592**	**82,40%**	**76,80%**	**88,10%**	Good		
Nbs1_1	8,140	**7,977**	**91,80%**	**89,20%**	**94,50%**	Optimal	**8,765**	**70,70%**
Nbs1_2	7,889	8,817	57,60%	52,30%	62,90%	Bad		
Nbs1_3	8,693	8,804	66,40%	63,20%	69,60%	Bad		
Nbs1_4	9,386	**9,170**	**73,80%**	67,10%	**80,50%**	Fair		
Rad51_1	7,707	**7,972**	**90,60%**	**87,90%**	**93,30%**	Optimal	7,977	64,70%
Rad51_2	6,409	8,150	59,10%	53,10%	65,00%	Bad		
Rad51_3	7,304	7,977	58,30%	55,00%	61,60%	Bad		
Rad51_4	8,313	8,209	51,10%	43,60%	58,70%	Bad		

* Length is reported in bp, TPM is referred to log_2TPM for GRO-seq counts, and the other (ChIP-seq) variable are in log_2TagCount.

AUC calculation allowed us to detect good **discriminant variables**. Table 10 reports both variable-specific and class-specific thresholds and AUCs. The Youden index perform poorly in almost every condition (often with a high rate of FN), and therefore it has been considered as a lower bound for an acceptable discriminant variable. Furthermore, Youden index oscillates, with huge variability among classes. Superiority to Youden index, lowest inter-class variability, and sensitivity-specificity equality are summarized inside the AUC measurements. A discriminant variable with AUC > 70% is considered as good if its confidence interval (CI) boundaries are both above 70%, or fair if only the upper boundary is above 70%. An optimal discriminant variable, instead, has an AUC > 80% and its boundaries both above 70%. Poor discriminant variables have only the upper boundary > 70%, while bad ones have all AUCs < 70%. We discarded both poor and bad variables. Again, discriminant accuracy calculations allowed us to confirm with a third independent estimate the DNA damage model (M_4) results. Repair factor show singularly slightly different behavior than as a combination inside the model. However, besides repair factors (that are known to be recruited in presence of a DNA damage), we are more interested in prognostic factors (i.e. unknown respect to damage), and specifically transcription-related ones. As predicted by the DNA damage model, Length is discriminant in Classes 1, 2, and 3, while Pol2 is discriminant in Classes 1, 3, and 4. Strikingly, gene expression is never discriminant, confirming that gene expression levels are only a collateral evidence, caused by the same molecular mechanisms that provokes DNA damage. Expression can be used as a confounder, or in absence of other data as a poorly-performant proxy variable. Overall thresholds confirm variable importance as a discriminant factor, providing a useful first-sight risk

assessment indicator. In Paragraph 3.5.5, we introduced the utility of Fisher exact test in evaluating the discriminant significance of a threshold.

Table 11. Fisher's exact test applied to the whole dataset of case-control promoters*. The two thresholds for gene length (A) and Pol2 (B), obtained through the sensitivity-specificity equality criterion, have been applied to the case-control Tier1 promoter dataset for assessing the odds-ratio and significance at the optimal cut-point. In both cases, damaged promoters had about 4.4-times higher odds than controls (p < 2.2E-56).*

A	DSB+	DSB-
$L \geq \theta_L$	425	664
$L < \theta_L$	202	1400

OR = 4.43, *p* = 2.17E-56

B	DSB+	DSB-
$P \geq \theta_P$	420	655
$P < \theta_P$	207	1409

OR = 4.36, *p* = 2.59E-55

In our case, we worked out two global prognostic thresholds: (i) the global gene length threshold θ_L = 15.5 kbp, and (ii) the global Pol2 threshold θ_P = 9.262 $\log_2$TagCount. These are the thresholds that maximize the difference between cases and controls. Fisher test, can be interpreted in terms of OR for a promoter to be damaged rather than damage free, allowing also a straightforward interpretation of the thresholds. Tables 11*A*,*B* report counts according to θ_L and θ_P, respectively. At optimal cut-point, damaged promoters had 4.4-times higher odds than controls.

4.1.2. Modelling non-linear interactions: association between DNA damage and translocations in Breast Cancer

We analyzed the association between **DSB** events and the occurrence of **gene fusions**. The association was investigated using a dataset of F = 5147 high confidence gene fusions, involving **intron-exon junctions** [179], among which 497 recur in more than one subject. We firstly modelled the prior probability of a damage P_{D+} and fusion P_{F+} events, considering them as independent point-source events that can occur in susceptible regions of the genome, identified by 238703 hg18 intron regions.

We first identified which of these regions are fusion-positive (F+) or DSB-positive (D+). We found a total of N_{F+} = 6796 F+ introns and N_{D+} = 5757 D+ introns, corresponding to a genomic coverage of P_{F+} = 154Mbp/3Gbp = 0.051 and P_{D+} = 210Mbp/3Gbp = 0.07. Then we calculated the expected number $E_{FD+} = F \cdot P_{F+} \cdot P_{D+}$ = 18, and its control value $E_{FD-} = F \cdot P_{F+} \cdot (1 - P_{D+})$ = 244. Value E_{FD+} (or E_{FD-}) corresponds to the expected number of independent fusion events that would be found associated (or not associated) to a DSB event by randomly picking a base within the candidate intron regions, in F trials.

To test DSB-Fusion association, we then set a series of Fisher’s exact tests, on the base of a threshold distance, as shown in Table 12. The observed number of DSB-associated fusion events (n_{FD+}) has been calculated by counting all the candidate F+ introns that were also D+. By contrast, all the F+ introns that were free of DSBs accounted for controls (n_{FD-}). We then calculated the distance from the closest DSB for both DSB-associated fusions (FD+) and for controls (FD-).

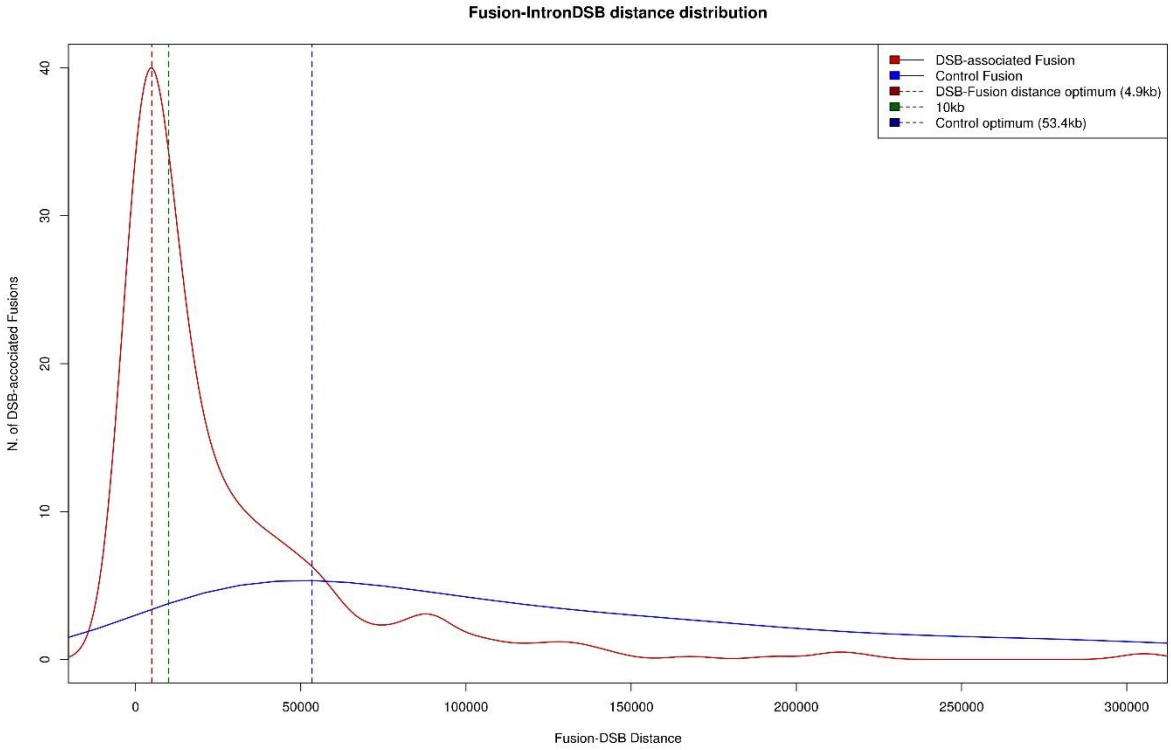
Table 12. Fisher's exact tests for assessing Fusion-DSB association. *A series of Fisher exact tests has been performed to assess the association at various Fusion-DSB threshold distances. These thresholds have been sampled from the distance distribution extracted from empirical distance data, through kernel density estimation. Fields are: threshold distance in kbp (θ_L), number of observed DSB-fusion associations (n_{FD+}), number of the observed DSB-negative fusions (n_{FD-}), odds-ratio for a fusion to be associated with a DSB (OR), and p-value of the association (p). Parameter ρ indicate the percentage of data explained by the model. The red threshold coincides with the optimum distance for DSB-associated fusions (i.e. the maximum value of the distance density distribution). The green threshold corresponds to the equality point, in which the absolute number of observed DSB-fusion associations is equal to the absolute number of control (i.e. DSB-independent) fusions. Finally, the blue threshold corresponds to the control distance density maximum. The last test includes all the observed DSB-associated fusions.*

θ_L	n_{FD+}	n_{FD-}	OR	p	ρ
4.9	127	57	29.87	9.50E-46	30%
10.0	197	131	20.27	2.36E-45	46.5%
20.0	261	261	13.52	1.11E-37	62%
53.4	356	696	6.93	1.28E-21	83.5%
80.0	383	1033	5.02	6.57E-15	90%
+∞	422	2817	2.03	1.46E-3	100%

We then used a Gaussian kernel density estimation (see Figure 17) to find distance maxima for both FD+ and control fusions. As opposite to the previous models, here we use a Gaussian kernel and a Fisher's exact test to model non-linear interactions. Density maximum for FD+ fusions was 4.5kbp, that is within the 5kbp error boundaries for the DSB event (i.e., from the previous paragraph, a DSB is a region DSB $\pm$ ε_D, where max(ε_D) = 5kbp). The optimum for controls is instead 53.4kbp, indicating that the association between fusions and DSBs can be clearly determined by distance. Besides optimum, thresholds can be easily set in terms of percentage of data explained by the model (ρ).

In general, considering a critical **genomic feature** (in our case, the **intron-exon junction**), for which we have **sufficiently large empirical data** (i.e. **fusions**), we can find one or more functions (in this case **distance density**), that can be used as an **independent kernel function** to **model interactions** within data (i.e. the **causal relationship between fusions and DSBs**). In biological terms, there is a (distance density) function that allow us to detect the **critical distance** from an intron-exon junction within which a DSB event **may cause** a genetic fusion.

Figure 17. Kernel density estimation for distance in DSB-associated and control fusions. *Gaussian kernel density estimations have been calculated for DSB-fusion distances in both DSB-associated fusions (red density) and control fusions (blue density). Both distributions have a clear global maximum (red and blue dashed vertical lines). The green dashed line is a reference 10kbp distance.*



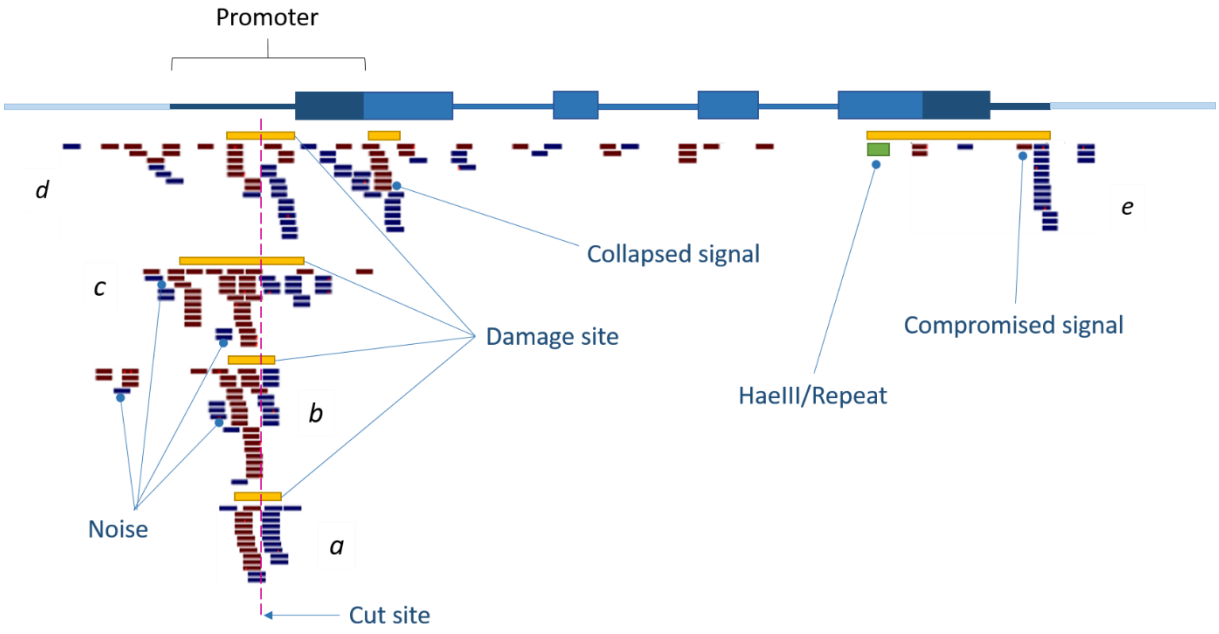
Modelling this non-canonical interaction between DSBs and intron-exon junctions as a possible causal mechanism for fusion occurrence, did not require a fixed conceptual model, in which the features (DSBs and junctions) are necessarily the static enumeration of their genomic location. Here, genomic features are represented through their probability and distance density distributions (as discussed in paragraph 3.3).

4.1.3. Modelling Double Strand Breaks

In Paragraph 3.5.2 we described the general framework for modelling genomic features from signal profiles. The procedure of genome-wide testing for detecting such data-driven features is defined **calling**. We saw models for point-source features, broad-source features, and summarized count data (e.g. gene expression, or enhancer activity). In general, a DSB is a point-source event in which both DNA strands are broken. BLISS technology allows the genome-wide detection of these events, with high confidence. However, many factors could negatively influence signal profile, including marked heteroscedasticity, high background signal, restriction enzyme interference, repeats and low complexity regions. In absence of any of these interferences, the BLISS signal appears as in Figure 18*a*: a pair of pileups adjacent to the cut site, with upstream reverse reads (dark red), and downstream forward reads (dark blue). This specific orientation is called a correct polarity. As shown in Figures 18*b*,*c*, reads with incorrect polarity are considered as noise (different from background, that is non-significant white-noise signal). Often, signal from different neighboring cut sites can merge or spread away from the original cut site (possibly due to small shifts in

the cut site from different cells), generating an overall broad-source signal that cannot be further resolved.

Figure 18. Different morphologies of the BLISS profile.



Another common problem is signal collapse, when the two pileups partially overlap, although they maintain the correct polarity. Since there is no clear explanation for them, they were simply flagged and excluded from further analysis (anyway, they usually represent around 1% of the total called BLISS peaks). We modelled peak calling using the Homer peak caller algorithm [171], with some minor modification, to cope with BLISS-related biases:

(i) Calling procedure is independent for each strand. This can correct for partial signal clearance (i.e. lack of one of the two pileups surrounding the cut site), and signal spreading (broad signal tails are often asymmetric, generating imbalanced strand counts). To account for strand specific signal skewness, we introduced the **strandness** parameter:

E63

$$v_j = log_2\left[\frac{\hat{n}_{j+} + c}{\hat{n}_{j-} + c}\right] P_j$$

where $\hat{n}_{j+}$ is the number of tags within significant pileups inside a damage site (see below). The $\hat{n}_{j+}$ and $\hat{n}_{j-}$ values are the contributions from forward

and reverse strand, respectively. The last term $\boldsymbol{P_j}$ = max(-log(p-value(*j*))), is the maximum of negative log10 p-values of the significant calls in the damage site. Scalar *c* is the pseudo-count. A significant pileup is defined as a **call**. The maximum p-value used in our analyses for a call is 1E-4.

(ii) The **damage site** is defined as the genomic segment covered from the beginning of the reverse pileup to the end of the associated forward pileup. This distance has a maximum limit equal to the parameter $\boldsymbol{\omega}$ = 5kbp. The limit of 5kbp corresponds to half the call local window (i.e. 10kbp). The local window is used to account for local variation (this is an important feature, given the strong BLISS signal heteroscedasticity).

(iii) If a call (from either + or – strand) do not have any companion within $\boldsymbol{\omega}$, it is flagged as an orphan signal (Figure 18*e*), and associated to the closest restriction site (in our case HaeIII) or repeat. If none of these elements are present within 5kbp, the call is discarded.

(iv) The total signal coming from a damage site is evaluated according to three parameters:

E63a Strength:

$$\boldsymbol{w_j = P_j \cdot log_2\left(\hat{n}_{j+} + \hat{n}_{j-}\right)}$$

E63b Balance:

$$\boldsymbol{Q_j = k \cdot max_k\left[q_{jk} \cdot log_2\left(\frac{\sum_i n_{ik} + c}{\sum_i n^o_{ik} + c}\right)\right]}$$

with

$$\boldsymbol{q_{jk} = \begin{cases} 1 & if \quad \sum_i n_{ik} - \sum_i n^o_{ik} > 0 \\ 0 & otherwise \end{cases}}$$

where $\boldsymbol{n_{ik}}$ is the sum of every *i*-th pileup for the *k*-th strand within the damage site *j*, and $\boldsymbol{n^o_{ik}}$ is the same calculation in the control sample. If no control sample is present, balance is not calculated.

The last parameter is **clearance**, equal to the proportion of bases in a damage site that are covered by an interfering feature (HaeIII, repeats, low complexity regions, low mappability regions). Genomic gaps are removed from calculations. Clearance is used to detect possible artificial signal removal, eligible for recovery (see below).

The calling procedure is executed in three steps: global call, local call, and targeted call. The global call is run to find the most significant hotspots of BLISS signal, but missing marginal tag density peaks. The local call is performed to recover more subtle variations over the background, with much less influence from huge density peaks. The targeted call is optional and performed only if a set of *a priori* genomic regions are known to be subject to damage. This call will be focused only on those regions, with stringent local parameters (although implemented in our pipeline, we used it only for parameter calibration purposes). The modelling framework is a Poisson-based GLM (see Paragraph 3.5.2).

AsiSI digestion (see Paragraph 4.1.4) allowed us to calibrate damage site parameters, referring to true cut events. We first called the damage sites combining the global and local call results (also combination procedures are automated inside our pipeline), as described above. Then we calculated TP (cut AsiSI sites detected as damage sites), FN (cut AsiSI sites not called as damage sites), TN (unrestricted AsiSI sites not called as damage sites), and FP (unrestricted AsiSI sites called as damage sites). We calculated a threshold strength level ($\boldsymbol{\theta_w}$ = 75) as the $\boldsymbol{w}$ value that maximizes TP and TN and minimizes FP and FN. The other calling parameters were adjusted in the same way (e.g. the calling bin size, global and local fold changes). Cut AsiSI sites were empirically identified as being Nbs1/γH2AX-positive. AsiSI cut sites allowed us to produce also a characteristic distribution of damage site lengths: $\boldsymbol{\lambda_{min}}$= 35bp, $\boldsymbol{\lambda_{25}}$= 90bp, $\boldsymbol{\lambda_{50}}$= 179bp, $\boldsymbol{\lambda_{75}}$= 357bp, $\boldsymbol{\lambda_{max}}$= 3968bp. Damage sites having $\boldsymbol{\lambda}$ > 200bp (~$\boldsymbol{\lambda_{50}}$) are considered as broad-source (i.e. they are likely to contain more than 1 DSB event). Value $\boldsymbol{\lambda_{max}}$< 4kbp confirmed that was a slightly permissive distance parameter.

Therefore, formally, a DSB is defined starting from a damage site as: mid-point ± $\boldsymbol{\varepsilon_D}$, where max($\boldsymbol{\varepsilon_D}$) = $\boldsymbol{\omega}$/2, in input, and max($\boldsymbol{\varepsilon_D}$) = $\boldsymbol{\lambda_{max}}$/2, in output. As shown in Figure 18, a damage site not always contains the strand separation point in its center, given the uneven tail length in signal distribution. This is likely due to slight shifts in the cut site position (i.e. the

true DSB position) in different cells. Therefore, in many contexts, we identified the DSB with the entire length of its damage site, rather than its mid-point. In this interpretation, the ***DSB** is a **latent variable**, estimated by its damage site*. During logistic model fitting (Paragraph 4.1.1), this assumption was used to model damaged promoters.

One last modification we had to consider is complete signal clearance. This happens when the summit of the tag density is destroyed by a restriction enzyme or another interfering factor. It is possible to recover many signals by measuring the residual BLISS signal as:

E64

$$\boldsymbol{\rho_j = P_j \cdot log_2(u_{j+} + u_{j-})}$$

almost identical to the strength definition, with the exception that $\boldsymbol{u_{jk}}$ is the residual signal for the strand *k* in the damage site *j*. The residual signal for each strand is calculated within 2kbp from the damage site mid-point, excluding the core 200bp surrounding it. Putative damage sites with $\boldsymbol{\rho_j \geq \theta_w}$ were marked for possible recovery.

After calling, a raw set of 190160 putative DSBs was detected. Using a p-value threshold of 0.0001, this number decreased to 88845 DSBs (Tier3). Filtering by strength ($\boldsymbol{\theta_w}$), further reduced this set to 55436 DSBs (Tier2). The final high-confidence Tier1 DSBs was obtained by detecting which of the Tier2 signals coincided with an Nbs1/Rad51/Xrcc4 enrichment site, leading to 8132 high-confidence genome-wide DSBs. Reproducibility of these DSBs, called from a second independent sample, ranged from 60% to 70% from Tier 3 to 1.

These DSB sets were used to build the DNA damage model and all the following analyses, including fusion-DSB association. Notably, many relevant associations resisted (with lower significance) in Tiers 2 and 3, suggesting that a large portion of significant DSBs is present also in these data sets, confirming that basal DNA damage is a highly frequent event, associated with normal transcriptional activities.

4.2. Discovering novel biomarkers in Frontotemporal Dementia

Frontotemporal Dementia (FTD) [173] is a complex disorder including various forms of cognitive impairments, associated to the degeneration of frontal and temporal brain lobe. It is the second most prevalent form of **neurodegenerative dementia**, after Alzheimer’s disease. Although few determinants are known to cause FTD-associated phenotype (including MAPT, GRN, C9orf72, TDP-43, and UBQLN2), many multiple loci seem to affect FTD etiology with relatively **small effect size**. Due to FTD phenotype complexity and the few evidences at molecular level, there are no comprehensive theories for FTD etiology, to date. In this paragraph, the methods used to build this theory through multivariate linear networks, will be exposed. Results indicated a general perturbation of Calcium/cAMP homeostasis and metabolism impairments, causing loss of neuroprotection and cell damage, respectively. The methodologies that will be presented in this section complement those exposed in the previous paragraph, moving from a local DNA-damage model, tested genome-wide, to a systemic molecular model, applied to the causal relationship between a complex phenotype and the genetics underlying it.

Our network-based method relies on three inputs: (i) quantitative data, (ii) set of seed genes, and (iii) a reference interactome. Quantitative data used in this study consisted of the first principal components (PC1), summarizing genotypes from our previous SNPs-to-gene approach of a case-control GWAS [176]. Briefly, raw case data considered 634 FTD samples, from 8 Italian research centres [172]. A total of 530 patients were included in the study, for which the age of

onset was 64.1 ± 20.7 years (Mean ± SD); range: 29.0–87.0) with male-to-female ratio 243/287. Raw control data (n = 926) has been collected during the HYPERGENES project (European Network for Genetic-Epidemiological Studies) [181]. Mean (±SD) age was 58.2 ± 6.1 years (range, 50.0 – 97.0). Quantitative data were generated by genotypes, using an additive encoding: 0 for the frequent homozygote genotype, 1 for the heterozygote, and 2 for the rare homozygote genotype. Polymorphisms were grouped by gene membership, where the gene region is defined by its locus coordinates ±5000 base pairs. A single gene was then represented as a continuous score by taking the first supervised PC1 on a subset of SNPs selected using outcome (case-control) information, i.e. supervised PC1 was a weighted sum of the SNPs within the gene region that maximize the variance of the gene score, which then varies with the outcome [176].

Seeds are defined as genes of interest, arbitrarily be chosen using existing knowledge (e.g. from databases such as OMIM, DisGeNET, or by text mining), and/or experimental data analysis (e.g. GWAS, gene expression, or ChIP-seq data). We used the list of FTD-associated genes as seed list, taken from our previous SNP-to gene approach [176], applying an FDR threshold < 10%, and yielding 280 putatively FTD-associated genes. Finally, we selected KEGG as source of interactome, due to several advantages provided by this database, including annotation curation and completeness, and GO biological process mappability for functional annotation. Moreover, KEGG provides a directed interactome, and therefore causality can be easily interpreted

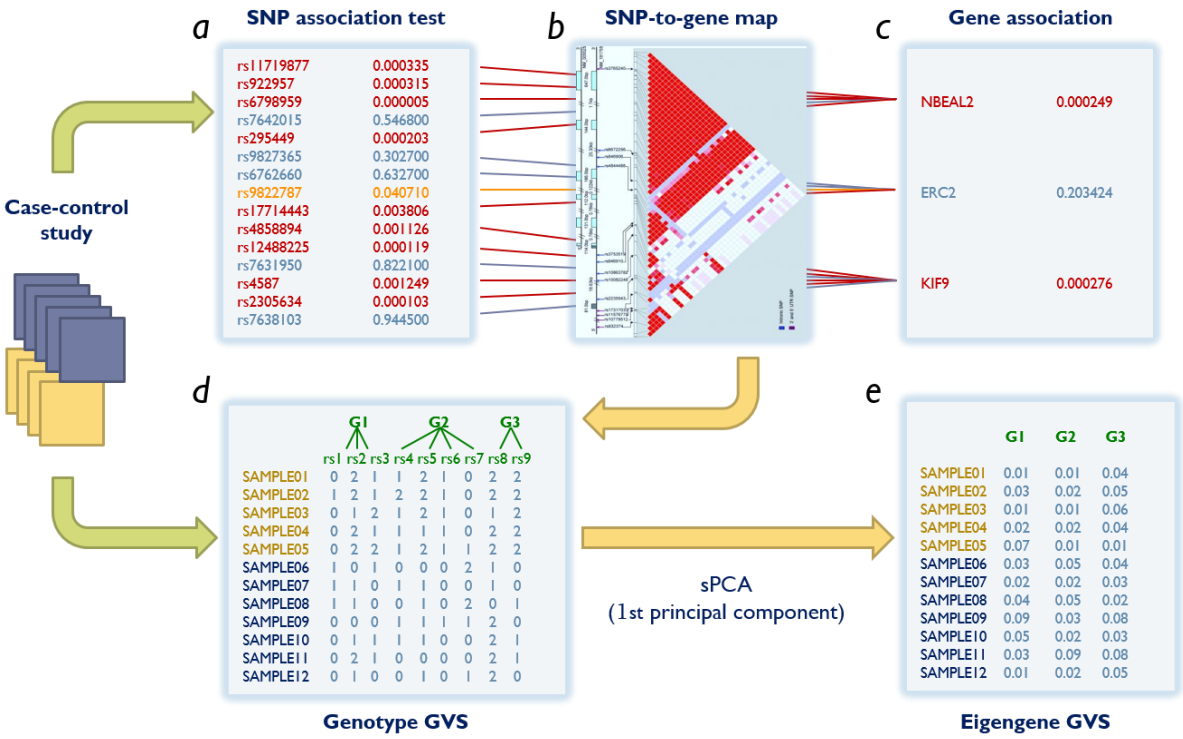
in terms of biological signalling, and used for model validation, although edge direction is not a necessary requirement in our method.

4.2.1. Modelling genetic variability through Genomic Vector Spaces

Genome-**Wide Association Studies** (GWAS) [172, 173] is a technology used to genotype hundreds of thousands of polymorphisms (namely **Single Nucleotide Polymorphisms**, or **SNPs**), to characterize genotypic variation at the base of complex phenotypes. A common study design is **case**-**control**, in which genetic variation from diseased samples is compared with healthy ones. In our FTD dataset, we compared genotypes from 530 FTD cases against 926 controls.

Genotypes for each locus are encoded as 0 for the common allele, and 1 for the alternate allele. Therefore, using an **additive encoding**, 2 represents the alternate homozygote, 1 the heterozygote, and 0 the reference homozygote. Figure 19 reports the general procedure used to summarize genomes into a computable data structure called **genome vector space** (**GVS**), in which every genomic feature is represented as a vector.

Figure 19. Genotype summarization into a genomic vector space.



Population structure were firstly analyzed by PCA and identity-by-descent analysis to assess that no relatedness between cases and controls was present. Imputation procedures lead to the generation of ~2.5 million SNPs. Then SNPs underwent to quality-check and analysis through logistic regression. The significance for every polymorphism (Figure 19*a*) was assessed using a likelihood ratio test (testing every genotype independently). Genotyped polymorphisms are 1 to few bases long, for many of which no functional information is available. On the other hand, tens of thousands of genes from the human genome are (at least partially) functionally characterized. To take advantage of this information, we grouped SNPs into genes, represented by the canonical transcript extended by 5kbp from both ends. This threshold was selected using two criteria: (i) the **linkage disequilibrium** (LD), associating SNPs inside genes to those being outside, tend to drop dramatically beyond 5kbp, and (ii) the number of

non-**causal** (i.e. noisy) SNPs increases with distance from transcripts, reducing the power of gene-based statistical tests (Figure 19*b*). Given this SNP-to-gene gathering, we may directly associate single genes to phenotypes by summarizing SNP-associated p-values (Figure 19*c*). Various methods can be used for SNP-to-gene summarization, including testing the effective number of associated SNPs using LD (i.e. **GATES test**) [173], logistic kernel-regression models (**SKAT**) [174], and **supervised PCA** (**sPCA**) [175]. The ensemble of these three models allowed us to define 280 novel FTD-associated genes [176]. Among the three methods, sPCA generated an **eigengene GVS**, from the original **genotype**-**based GVS** (Figures 19*d*,*e*). The assumption behind sPCA is that only a subset of SNPs (represented as a latent variable), in a gene, is associated to the disease phenotype. The first principal component (**PC1**) of a gene (i.e. the **eigengene**) is used as an estimate of the latent variable. SNPs are ranked on the base of their log-likelihood, and only the first quartile is retained, to rule out noisy data. These SNPs are then used for PCA. A univariate logistic model is used to evaluate effect and significance of the eigengene over the binary outcome {*cases*, *controls*}. Therefore, PC1 values summarize genotypes from significant SNPs, from the additive encoding, to a continuous scale. In this representation, a gene is a vector of 530 + 926 PC1 values. Using the 280 FTD-related genes (each associated to a PC1 vector), that will be called **seeds** from now on, we generated an interaction network using pathway information form KEGG. KEGG signaling pathways are directed and curated, allowing a straightforward interpretation in terms of causal biological interactions. In this way, we converted the GVS into a directed graph.

We convert this graph into a **Structural Equation Model** (**SEM**) [135-138], a linear multivariate network describing the relationships among observed variables:

E65*a*

$$\boldsymbol{Y_j = \sum_{k \in pa(j)} \beta_{jk} Y_k + U_j \qquad \forall\, j \in V}$$

with covariance structure

E65*b*

$$\boldsymbol{cov(U_j, U_k) = \begin{cases} \psi_{jk} & if \quad j = k \; or \; j \in sib(j) \\ 0 & otherwise \end{cases}}$$

where $\boldsymbol{V}$ = { $\boldsymbol{i}$ } is the index set of the graph vertices $\boldsymbol{Y_i}$ (i.e. the observed variables), *pa*($\boldsymbol{i}$) is the parent set of the $\boldsymbol{i}$-th observed variable (i.e. the set of variables acting on $\boldsymbol{Y_i}$), and *sib*($\boldsymbol{i}$) is the siblings set of the $\boldsymbol{i}$-th observed variable (i.e. the set of variables influenced by the same parent set of $\boldsymbol{Y_i}$). While the system of linear equations (E65*a*) defines every node as characterized by the relationships with its parent set, the covariance structure (E65*b*) defines the unobserved relationships among nodes. Like other relationships we discussed in Paragraph 3.5.6, the SEM architecture (i.e. **path diagram**) can be also seen as a mixed graph Γ = (V, E), including both directed (i.e. causal) and bidirected (i.e. covariance) interactions in the edge set E, among variables in the vertex set V. Figure 20*a* shows a schematic representation of the SEM building procedure.

SEM framework has an important feature: it can model information flow through nodes (i.e. genes), such that **direct** and **indirect effects** of one node to another can be estimated. SEM can model every effect simultaneously, providing also indices of goodness of fit and modification indices, providing hints for model architecture

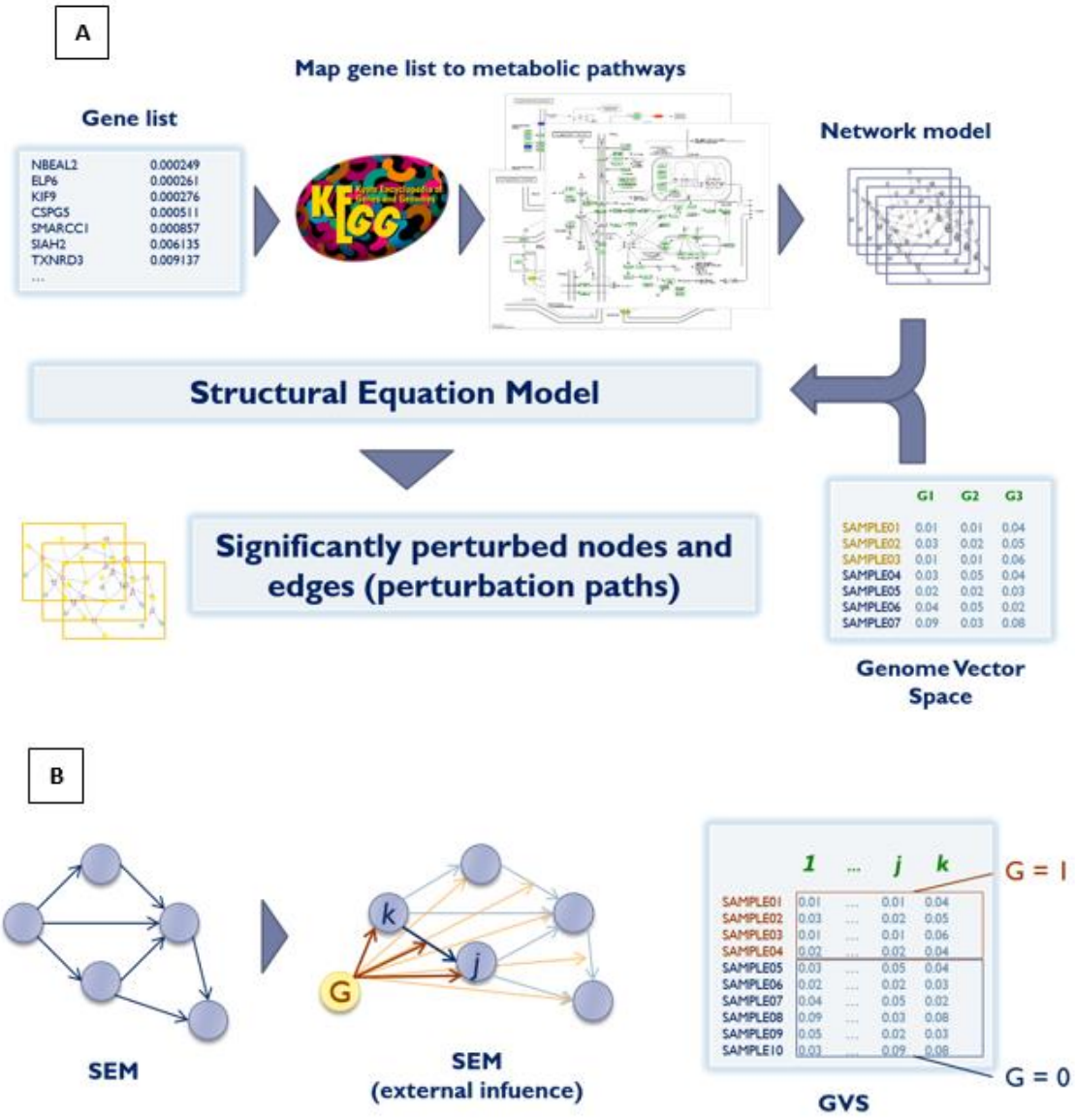
improvement. SEM models were assessed using AIC scores, and the Standardized Root Mean Squared Residual (SRMR, see Equation E66) is used as main indicator of good model fitting (i.e. SRMR < 0.05). SRMR is a measure of difference between the **observed covariance matrix** (*S*) and the **model-induced covariance matrix** ($\Sigma(\theta)$):

E66

$$\boldsymbol{SRMR} = \frac{\sum_{j=1}^{n}\sum_{k=j+1}^{n}\left(\boldsymbol{s}_{jk} - \boldsymbol{\sigma}_{jk}\right)^2 / \boldsymbol{s}_{jj}\boldsymbol{s}_{kk}}{\boldsymbol{n}(\boldsymbol{n}+\boldsymbol{1})/\boldsymbol{2}}$$

The model-induced covariance matrix is defined as $\Sigma(\theta) = (I - B)^{-1}\Psi(I - B)^{-T}$, where $Y = BY + U$ (i.e. the linear equations system), $\Psi = \text{cov}(U)$ is the unobserved covariance structure of the model, and $\theta = (\beta, \psi)$ are the model parameters. The null hypothesis under which the model is estimated is that $\Sigma(\theta) = \Sigma_0$, where $\Sigma_0 = E(S)$. A Pearson’s χ^2 test is used, with $\chi_d^2 = -2[\mathcal{L}(\Sigma(\theta)) - \mathcal{L}(\Sigma_0)]$ with $d = \boldsymbol{n}(\boldsymbol{n} + 1)/2 - b$ degrees of freedom (where *b* is the number of parameters in the model).

Figure 20. Schematic representation of SEM building procedure. *A SEM can be used (A) for feature interaction modeling, starting from a list of features (e.g. genes) and a reference interaction network (e.g. KEGG signaling pathways). Once the gene list is mapped onto a reference we obtain a network model that can be easily converted into a SEM. SEM can be fitted and used for finding significant nodes and relationships in a multigroup analysis or (B) it can be inspected for significant association with a specific phenotype G (G may represent any external influence on the model).*



Non-significant p-values (i.e. $p < 0.05$) indicate that the model well explained data variability. In other terms, we desire the model to explain as much data variability as possible, with no residual variability left. Once the SEM is built, we can proceed in two ways: (i) perform a **multigroup analysis** (the case-model vs. the control-model), or (ii) model genetic variability and group influence within the same model. For the FTD data we used the second strategy (Figure 20*b*).

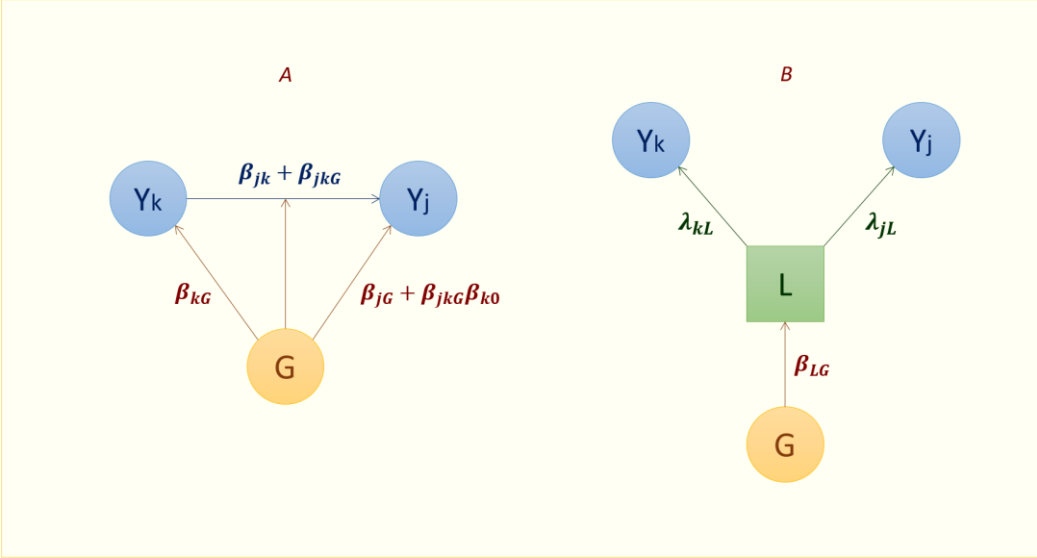
We already modelled external perturbation on the model as a confounding effect (see Paragraph 4.1.1). However, in that case, this influence was undesired and therefore cancelled out through model stratification. In other conditions, there could be some external influence that cannot be addressed inside the model, but we know somehow influencing it. If we could know which features are specifically influenced, we would explicitly add them inside the model. But in case of unknown causal relationships we can represent this external influence as generally acting over all the elements of the model, and estimating their actual impact on nodes and interactions. The premise is that the phenomenon underlying data is only part of the universe, that is represented by the model. We call this initial model the interactome, since it contains the whole set of known trustable interactions, or at least an acceptable approximation of it. What we want is to extract the portion of the interactome that best represents data, possibly recovering or predicting the portion of data variability that is not present in the interactome. In case of a diseased phenotype, data should point out which portion of the interactome is perturbed (i.e. altered) by the disease.

The concept of perturbation is quite immediate when we talk about structural features of the genome, including genes. For instance, a gene that is altered in its expression or carrying rare variants, respect to the normal (i.e. physiological) condition, is deemed as significantly

perturbed if its normal molecular function is compromised, lost, or a new function is gained. The concept of interaction perturbation (or, equivalently, edge perturbation) [129] is instead less immediate, but somehow related to structural perturbation. In the case of genetic variability, interaction perturbation may occur when a specific variant cause even a slight modification in the physical-chemical properties of a gene product. This modification may alter the way the gene product interacts with proteins or chromatin, and therefore the information carried by that interaction is also altered. Furthermore, this information is propagated inside the system, inducing consequent modifications in interacting genes functionality. In addition, just like loss and gain of molecular functions, also interactions can be lost or gained.

The quantification of additive genotypes into continuous variables allows this interpretation in terms of estimated effects, within the SEM framework. Since the concept of edge perturbance requires the comparison between at least two conditions (cases and controls), data can be used to evaluate multigroup differences. Another way respect to multigroup analysis is to model group influence as an external risk factor acting both on genes and their interactions. In other terms, the group (i.e. the phenotype) is modelled as a special case of moderated mediation (Figure 21*A* and Paragraph 3.5.6). We can reduce the problem to a simple pairwise interaction, in which a source node $\boldsymbol{Y_k}$ sends a signal to the sink node $\boldsymbol{Y_j}$. The information spreading from source to sink is represented by the oriented interaction $\boldsymbol{Y_k \rightarrow Y_j}$.

Figure 21. Pairwise gene interaction models to assess interaction perturbation. *Pairwise node interactions can be used to model interaction alteration in both interacting (A) and non-interacting (B) nodes. Pairwise interactions consider a source (Y_k) that generates a signal and a sink node receiving it (Y_j). The information travelling from a node to another is represented by their interaction. An external agent G influencing these three elements models the possible alteration.*
The same concept can be also extended to non-interacting elements. In other terms, significant covariation of bow-free nodes (de facto a leakage of the model), may denounce the presence of a latent perturbed interaction (L variable in B).



This interaction can be intended both in terms of **information spreading** and as an example of **causal interaction**. In the former interpretation, a **signal** (e.g. from a membrane receptor) spread the information through the system (e.g. the cytosol and the nuclear matrix), until it is carried to the final **effector** (e.g. a transcription factor). The latter interpretation is practically equivalent to the first one, but formally can be thought in terms of causal effect of the independent variable $\boldsymbol{Y_k}$ over the dependent one $\boldsymbol{Y_j}$ (Figure 21*A*). If the model does not explicitly report the interaction term, the couple ($\boldsymbol{Y_k}$, $\boldsymbol{Y_j}$) is defined as being **bow-free**. However, if a significant covariation exists between them, it is a symptom of a **latent**

interaction, missing in the interactome. This interaction can be modelled as a latent variable, that can be influenced by the external factor, just like a normal interaction (Figure 21*B*). Testing these interactions through a t-test over the whole SEM leads to interaction-specific p-values, that can be used to weight the amount of information spreading from one node to the other. If the p-value is sufficiently low, the influence of FTD phenotype significantly affects information spreading between source and sink nodes. Similarly, SEM fitting can evidence the presence of nodes that are significantly perturbed. When referring to a single node, perturbation means that the gene carries one or more variants that are significantly associated to the FTD phenotype.

Since SEM are susceptible to noise and overfitting, it is advisable to couple these methods with a data reduction algorithms. Since SEM are basically graphs, every graph based algorithm can be applied to them, including **shortest paths** [177], **random walkers** [178], and **Steiner’s trees** [139-142]. In our FTD study we applier a modified version of the shortest path heuristic (SPH) distance algorithm, applied to the Steiner’s tree problem, using the inverse of -log(p-value) as edge weight (used to calculate weighted shortest paths). The weight corresponding to 0.05 was used as threshold for defining edge perturbation.

The initial KEGG signaling sub-interactome, connecting our seeds (4073 nodes and 34606 edges), was reduced to an informative disease-relevant core of 167 nodes and 166 edges (a Steiner’s tree), maximizing system perturbation. Among the 280 seeds, 77 were included inside the final core, as perturbed nodes. Among the 166 interactions, 117 turned out to be perturbed. We then converted the core into a SEM, fitted it and extended using significant covariances (modeled using the framework in Figure 21).

The final FTD Steiner tree (Figure 22) revealed the presence of: (i) 43 sources (33/43 perturbed path sources); (ii) 49 sinks (38/49 perturbed path sinks); and (iii) 90 connectors. The Steiner’s tree backbone, defining the mainstream perturbation route, was defined by the genes *CAMK2A-EP300-TCF7L2*, bifurcating to *EGFR-CTNNB1* and *MAPK8-JUN*. SEM analysis of the Steiner tree was performed by fitting models with different constraints, where the lowest AIC score model (AIC = -9324), yielded good fitting (SRMR = 0.026), and detected 98 significantly perturbed nodes (74 FTD-seeds, and 24 non FTD-seeds), and 43 significant covariances (z-threshold > 2), connecting 43/49 leaf genes. After merging the latent variables underlying the leaf-genes covariances, we obtained a sub-network of 210 (167 genes + 43 LVs) nodes and 252 (166 + 2*43) edges.

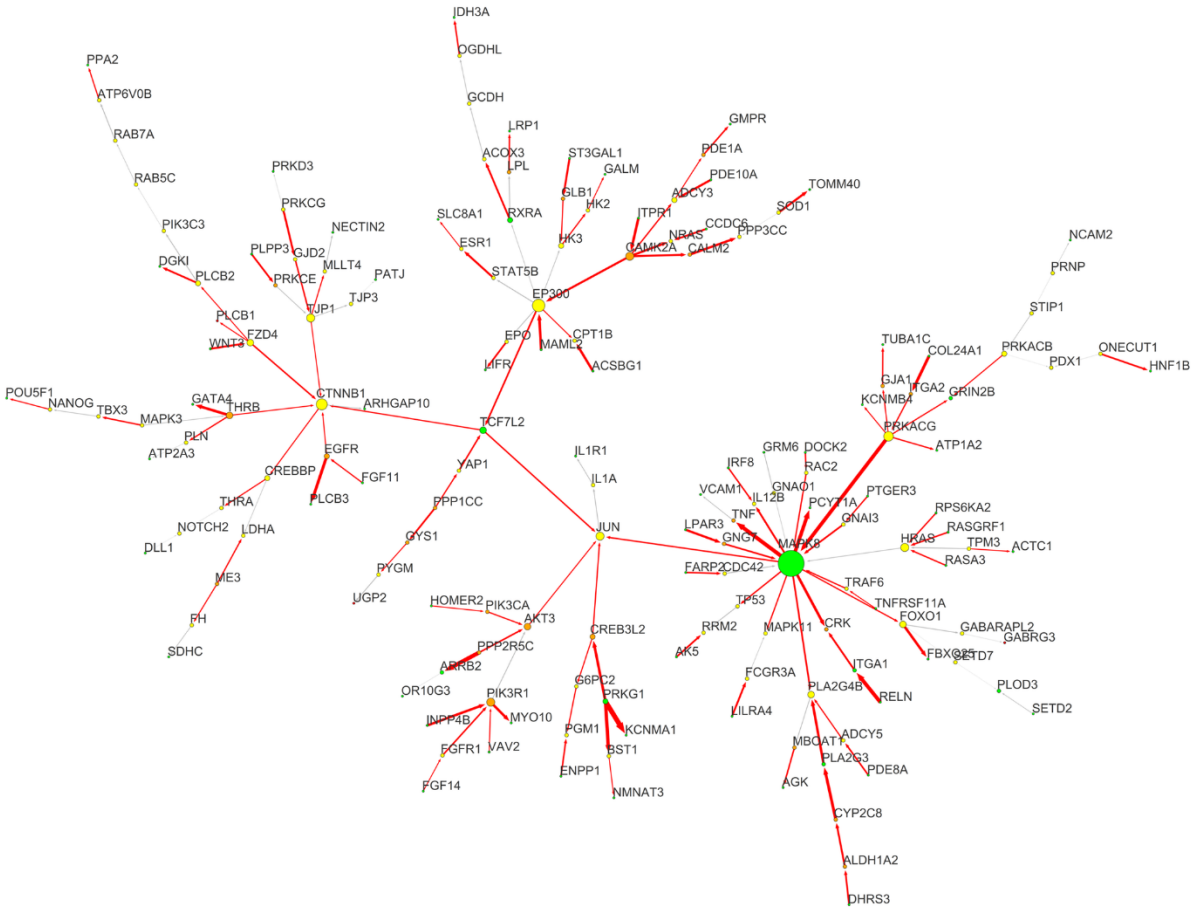
The final directed FTD-associated network revealed the key role of components of signaling pathways involved in long-term potentiation, neuroprotection and response to reactive oxygen species, including the WNT signaling pathway, Calcium homeostasis, cAMP signaling, and MAPK-JNK signaling. Specifically, the significant KEGG-enriched and Reactome-enriched functions sustained by the FTD-network (Benjamini-Hochberg adjusted Fisher’s exact test p < 1e-5), included: *MAKP signaling* (KEGG:map04010), *RAS signaling* (KEGG:map04014), *cAMP signaling* (KEGG:map04024), *cGMP-PKG signaling* (KEGG:map04022), *Gap Junction* (KEGG:map04540), *WNT signaling* (KEGG:map04310), *Insulin signaling* (KEGG:04910), *Long-Term Potentiation* (KEGG:map04720), and *FoxO signaling* (KEGG:map04068), *Signaling by EGFR* (Reactome: R-HSA-177929), *DAP12 signaling* (Reactome: R-HSA-2424491), *Transmission across chemical synapses*

(Reactome: R-HSA-112315), *Neurotransmitter receptor binding and downstream transmission in the postsynaptic cell* (Reactome: R-HSA-112314), *G protein mediated events* (Reactome: R-HSA-112040), *PLC beta mediated events* (Reactome: R-HSA-112043), *Ca2+ pathway* (Reactome: R-HSA-4086398), and *Activation of NMDA receptor upon glutamate binding and postsynaptic events* (Reactome: R-HSA-442755).

The core structure of the essential FTD-network highlighted the central role of four biological processes in FTD: (i) WNT/Ca^{2+} signaling, (ii) response to oxygen-containing compounds (ROCC), (iii) postsynaptic plasticity and Long Term Potentiation (LTP), and (iv) MAPK-JNK signaling. The most relevant evidence was the high perturbance levels of different classes of protein kinases, including PKA, PKC, MAPK, and AKT kinases, that are known to be activated in response to energetic metabolism stresses, and influencing postsynaptic LTP through modulation of Calcium and cAMP homeostasis. The FTD-network can be divided into in two main modules: (i) the non-canonical WNT/Ca2+ signaling (ncWNT)-associated module (including perturbed receptors *ITPR1-CAMK2A*, *WNT3-FZD4*, *FGF11-EGFR*, and *THRB*), and (ii) the MAPK/JNK-associated module, including *PRKACG-MAPK8*, *AKT3*, and *CREB3L2*, and their sink *JUN*. The interface between the two sub-modules is represented by the *TCF7L2-JUN-MAPK8* interaction, demonstrating an overall disruption of the DAG-IP3/Ca2+/cAMP homeostasis and energetic metabolism, leading to the induction of several oxidative stress-responsive Serine/Threonine (Ser/Thr) kinases and their related targets, including members of the CREB family,

protein phosphatases, junction proteins, K^+ cannels, and members of the FoxO signalling pathway. These critical Ser/Thr kinases include: (i) the Ca^{2+}-dependent *CAMK2A*, *PRKCG*, *PRKACG*, *PRKACB*, and *PRKG1*; (ii) the MAP-kinases *MAPK8* and *MAPK11*; and (iii) the AKT-kinase *AKT3*. Every Ser/Thr kinase has a specific influence over a group of functionally-related nodes, converging towards common “super-sinks”: *CTNNB1* and *JUN*. Given these evidences, we developed a novel metabolism-based theory for FTD-aetiology, characterized by a strong connection between Calcium and cAMP metabolism, oxidative stress-induced Ser/Thr kinases activation, and postsynaptic membrane LTP, suggesting a possible combination of neuronal damage and loss of neuroprotection leading to cell death.

Figure 22. Steiner’s tree used as core backbone for the final FTD network. *The Steiner’s tree (covariances omitted respect to the final FTD network to improve visualization) includes 167 nodes and 166 edges. Node size is proportional to its degree. Node color code: perturbed seeds (green), non-perturbed seeds (red), perturbed connectors (orange), and non-perturbed connectors (yellow). Edge thickness correspond to a weight* $w > 0.33$ *(i.e.* $p < 0.05$ *threshold over the nominal edge p-value); while edge colors correspond to Disease Ontology (DO) similarity (Wang’s measure). Similarities greater than 0.5 yield a red edge.*



CHAPTER 5

Concluding remarks

I mostri esistono, ma sono troppo pochi per essere davvero pericolosi.
Sono più pericolosi gli uomini comuni,
i funzionari pronti a credere e obbedire senza discutere …
Occorre dunque essere diffidenti con chi cerca di convincerci
con strumenti diversi dalla ragione, ossia i capi carismatici:
dobbiamo essere cauti nel delegare ad altri il nostro giudizio e la nostra volontà.

Primo Levi
Se Questo è un Uomo, 1976

At the beginning of the present discussion, we presented the genome as a **complex system**, not just as a static and monolithic entity, but as the ensemble of the **hereditary information** of an **organism** or, in evolutionary terms, of a **species**. Here the term information defines the full set of instructions to determine the **phenotype** of an organism, starting from a genomic, epigenomic, and interactomic variability, representing the hardware of such information. We also defined the conditions under which this hardware is altered, generating an unexpected departure from the original equilibrium (that we summarized in the concept of interactome). In this context, **sequencing sciences** offered through the last two decades the possibility to dive into this complexity through **high-throughput sequencing** methods, showing a versatility that was not achieved by

any other biotechnology before. This scientific revolution led to a series of paradigm changes, and the collateral need of efficient technical and methodological innovations. However, this process was not gradual, exposing genome research to an explosive **data deluge**, thus causing a series of **bottlenecks**: **technological**, due to the possibility of acquiring and distribute these data; **computational**, due to the necessity of hardware facilities to analyze such **big data**; and **methodological**, due to the existence of efficient data reduction and analysis algorithm, capable of capturing the **informative** portion of genomic complexity. First attempts to cope with these issues, led to the development of several solutions in each of these three fields, generating as much variability as the data could provide. However, in the last years, there have been huge efforts towards the generation of **unified standards and databases** (file formats, languages, and protocols) to acquire and distribute data and metadata; **ontologies** to uniform knowledge acquisition and exchange; **integrated platforms**, to improve data analysis and sharing results; and finally, **integrative** and **ensemble methodologies**, to provide a strong theoretical background, capable of capturing **true biological variation**, often masked behind **marginal synergic processes**. The goal of this thesis was to shed light on this last problem, trying to expose (at least part of) a **unifying theoretical foundation** needed to approach the study of genome complexity. We expressed this goal as a specific question:

Given a set of observations X, that we assume to contain the full description of a genomic process $f(X)$, is there any theory capable of generating a model $\hat{f}(X)$ sufficiently close to it?

If we could have an accurate description of the real biological process $f(X)$, we could address this problem from a Bayesian point of view, in which

$$P(f|X) = \frac{P(f) \cdot P(X|f)}{P(X)}$$

where the description we provide of reality, given data, depends on the prior knowledge we have about reality itself, i.e. $P(f)$. This possibility would be exciting if we only could have an acceptably large knowledge of the true biological process underlying data. Unfortunately, the current state of the art in sequencing sciences is completely the opposite: we have huge amount of HTS data (X), underlying complex biological processes (f), for which we are way far from a complete understanding.

Fortunately, the high accuracy reached by sequencing technologies, allow us to put the problem under an opposite point of view. Given the partial, but consistent, knowledge we currently have of genome complexity, we can assume that sequencing data may truly contain the information that is necessary to **build** an accurate model of reality, that we called $\hat{f}(X)$. This assumption is supported by the high-throughput gained of the current NGS technologies. The only thing we must care about, is to find a fundamental theory capable of assessing the divergence $I[f(X), \hat{f}(X)]$ between reality $f(X)$ and the model we generated from data $\hat{f}(X)$. We showed that this unifying theory actually exists, and it is represented by the **Kullback-Leibler Divergence** (KLD). KLD provides both the theoretical framework for model selection (through the definition of the AIC), and the basis for the decision rule in random forests. These two complementary aspects, are further supported by the **sampling theory**, and the

associated **likelihood function** and **Akaike Information Criterion** (AIC), that allow us to **produce** and **rank** different representations of reality, given a **parametric model** $M(X;\theta)$. We gave a generic definition of this model as the **Genomic Interaction Model**, that is deeply founded over data (i.e. it is data-driven), and expressed through the generic **Genomic Vector Space** (GVS) data structure, composed by a **sampling space** *S*, a **variable space** X (that is defined by data itself, represented by HTS **tag density profiles**), an **architecture** over data $\eta(X)$, and a monotonic variable t defining its **evolution**. In other terms, GVS = $\{S, \eta(X)\}_t$. The architecture η can be represented in different ways, from the more traditional **Generalized Linear Models** (GLM), to the more complex **interaction networks**, to the less used **Structural Equation Models** (SEM). Beside versatility, these last two representations have also the advantage of being easily coupled with data reduction algorithms (e.g. weighted shortest path, Steiner's trees, random walkers) that may improve the model by reducing the amount of noisy data onto which it is built. Together, these aspects represent a minimum theoretical core and a central data structure (i.e. the GVS), providing a base to define and model virtually any kind of **genomic interaction model** (GIM).

Besides implementation details, we also showed how we can gain the same results as the **Maximum Likelihood Estimation** (MLE), remaining in the same context of **KLD** and **sampling theory**, by coupling **bootstrap** and **cross-validation** to **machine learning**, using **Random Forests Classifiers** (RFC). This enabled the possibility to validate a model- and data-driven result, with a model-free methodology, allowing at the same time the generation of new knowledge that is independently validated within the same theoretical framework. Our aim (i.e. the output) is not to provide a description of reality (the

genome), but to gain knowledge that is used as a building block to come closer and closer to the real description of genomic complexity, such that $I\left[f(X), \hat{f}(X)\right] = \varepsilon$, with ε being smaller and smaller.

In conclusion, sequencing sciences allowed researchers all over the world to exchange not only data and methodologies, but also a unified (i.e. integrated) comprehension of genomic, and therefore, human biology complexity.

References

[1] WATSON, James D., et al. Molecular structure of nucleic acids. Nature, 1953, 171.4356: 737-738.

[2] MAXAM, Allan M.; GILBERT, Walter. A new method for sequencing DNA.Proceedings of the National Academy of Sciences, 1977, 74.2: 560-564.

[3] SANGER, Frederick; NICKLEN, Steven; COULSON, Alan R. DNA sequencing with chain-terminating inhibitors. Proceedings of the National Academy of Sciences, 1977, 74.12: 5463-5467.

[4] RITCHIE, Marylyn D., et al. Methods of integrating data to uncover genotype-phenotype interactions. *Nature Reviews Genetics*, 2015, 16.2: 85-97.

[5] WHEELER, David A., et al. The complete genome of an individual by massively parallel DNA sequencing. nature, 2008, 452.7189: 872-876.

[6] SHENDURE, Jay; AIDEN, Erez Lieberman. The expanding scope of DNA sequencing. *Nature biotechnology*, 2012, 30.11: 1084-1094.

[7] STEPHENS, Zachary D., et al. Big data: astronomical or genomical?. *PLoS Biol*, 2015, 13.7: e1002195.

[8] LI, Yixue; CHEN, Luonan. Big biological data: challenges and opportunities.*Genomics, proteomics & bioinformatics*, 2014, 12.5: 187-189.

[9] METZKER, Michael L. Sequencing technologies—the next generation. *Nature reviews genetics*, 2010, 11.1: 31-46.

[10] GLENN, Travis C. Field guide to next-generation DNA sequencers. *Molecular ecology resources*, 2011, 11.5: 759-769.

[11] SCHADT, Eric E.; TURNER, Steve; KASARSKIS, Andrew. A window into third-generation sequencing. *Human molecular genetics*, 2010, 19. R2: R227-R240.

[12] STODDART, David, et al. Single-nucleotide discrimination in immobilized DNA oligonucleotides with a biological nanopore. *Proceedings of the National Academy of Sciences*, 2009, 106.19: 7702-7707.

[13] LENZ, Georg; STAUDT, Louis M. Aggressive lymphomas. *New England Journal of Medicine*, 2010, 362.15: 1417-1429.

[14] BEATO, Miguel; WRIGHT, Roni H.; VICENT, Guillermo P. DNA damage and gene transcription: accident or necessity? *Cell research*, 2015, 25.7: 769-770.

[15] KIM, Nayun; JINKS-ROBERTSON, Sue. Transcription as a source of genome instability. *Nature Reviews Genetics*, 2012, 13.3: 204-214.

[16] SCHWER, Bjoern, et al. Transcription-associated processes cause DNA double-strand breaks and translocations in neural stem/progenitor cells. *Proceedings of the National Academy of Sciences*, 2016, 113.8: 2258-2263.

[17] VOELKERDING, Karl V.; DAMES, Shale A.; DURTSCHI, Jacob D. Next-generation sequencing: from basic research to diagnostics. *Clinical chemistry*, 2009, 55.4: 641-658.

[18] ASHLEY, Euan A. Towards precision medicine. *Nature Reviews Genetics*, 2016, 17.9: 507-522.

[19] BYRON, Sara A., et al. Translating RNA sequencing into clinical diagnostics: opportunities and challenges. *Nature Reviews Genetics*, 2016.

[20] CASTRO-VEGA, Luis Jaime, et al. Multi-omics analysis defines core genomic alterations in pheochromocytomas and paragangliomas. *Nature communications*, 2015, 6.

[21] HEINZ, Sven, et al. The selection and function of cell type-specific enhancers. *Nature Reviews Molecular Cell Biology*, 2015, 16.3: 144-154.

[22] SANYAL, Amartya, et al. The long-range interaction landscape of gene promoters. *Nature*, 2012, 489.7414: 109-113.

[23] DIXON, Jesse R., et al. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, 2012, 485.7398: 376-380.

[24] DOWEN, Jill M., et al. Control of cell identity genes occurs in insulated neighborhoods in mammalian chromosomes. *Cell*, 2014, 159.2: 374-387.

[25] HNISZ, Denes, et al. Super-enhancers in the control of cell identity and disease. *Cell*, 2013, 155.4: 934-947.

[26] PARK, Peter J. ChIP–seq: advantages and challenges of a maturing technology. *Nature Reviews Genetics*, 2009, 10.10: 669-680.

[27] LANDT, Stephen G., et al. ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia. *Genome research*, 2012, 22.9: 1813-1831.

[28] THOMAS-CHOLLIER, Morgane, et al. A complete workflow for the analysis of full-size ChIP-seq (and similar) data sets using peak-motifs. *Nature protocols*, 2012, 7.8: 1551-1568.

[29] PEPKE, Shirley; WOLD, Barbara; MORTAZAVI, Ali. Computation for ChIP-seq and RNA-seq studies. *Nature met*

[30] GHOSH, Debashis; QIN, Zhaohui S. Statistical issues in the analysis of ChIP-Seq and RNA-Seq data. *Genes*, 2010, 1.2: 317-334.

[31] ROZOWSKY, Joel, et al. PeakSeq enables systematic scoring of ChIP-seq experiments relative to controls. *Nature biotechnology*, 2009, 27.1: 66-75.

[32] KOEHLER, Ryan, et al. The uniqueome: a mappability resource for short-tag sequencing. *Bioinformatics*, 2011, 27.2: 272-274.

[33] ZHANG, Yong, et al. Model-based analysis of ChIP-Seq (MACS). *Genome biology*, 2008, 9.9: 1.

[34] ZANG, Chongzhi, et al. A clustering approach for identification of enriched domains from histone modification ChIP-Seq data. *Bioinformatics*, 2009, 25.15: 1952-1958.

[35] BIRNEY, Ewan, et al. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature*, 2007, 447.7146: 799-816.

[36] ENCODE PROJECT CONSORTIUM, et al. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 2012, 489.7414: 57-74.

[37] ERNST, Jason; KELLIS, Manolis. ChromHMM: automating chromatin-state discovery and characterization. *Nature methods*, 2012, 9.3: 215-216.

[38] KUNDAJE, Anshul, et al. Integrative analysis of 111 reference human epigenomes. *Nature*, 2015, 518.7539: 317-330.

[39] WELTER, Danielle, et al. The NHGRI GWAS Catalog, a curated resource of SNP-trait associations. *Nucleic acids research*, 2014, 42.D1: D1001-D1006.

[40] VAISSIÈRE, Thomas; SAWAN, Carla; HERCEG, Zdenko. Epigenetic interplay between histone modifications and DNA methylation in gene silencing. *Mutation Research/Reviews in Mutation Research*, 2008, 659.1: 40-48.

[41] MITELMAN, Felix; JOHANSSON, Bertil; MERTENS, Fredrik. The impact of translocations and gene fusions on cancer causation. *Nature Reviews Cancer*, 2007, 7.4: 233-245.

[42] SAWICKI, Mark P., et al. Human genome project. *The American journal of surgery*, 1993, 165.2: 258-264.

[43] EWING, Brent, et al. Base-calling of automated sequencer traces usingPhred. I. Accuracy assessment. *Genome research*, 1998, 8.3: 175-185.

[44] EWING, Brent; GREEN, Phil. Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome research*, 1998, 8.3: 186-194.

[45] LEINONEN, Rasko; SUGAWARA, Hideaki; SHUMWAY, Martin. The sequence read archive. *Nucleic acids research*, 2010, gkq1019.

[46] GORDON, A.; HANNON, G. J. Fastx-toolkit. *FASTQ/A short-reads preprocessing tools (unpublished)*: *http://hannonlab. cshl. edu/fastx_toolkit*, 2010.

[47] MEYERSON, Matthew; GABRIEL, Stacey; GETZ, Gad. Advances in understanding cancer genomes through second-generation sequencing. *Nature Reviews Genetics*, 2010, 11.10: 685-696.

[48] HODGES, Emily, et al. Genome-wide in situ exon capture for selective resequencing. *Nature genetics*, 2007, 39.12: 1522-1527.

[49] PORRECA, Gregory J., et al. Multiplex amplification of large sets of human exons. *Nature methods*, 2007, 4.11: 931-936.

[50] SMITH, Temple F.; WATERMAN, Michael S. Identification of common molecular subsequences. *Journal of molecular biology*, 1981, 147.1: 195-197.

[51] LI, Heng, et al. The sequence alignment/map format and SAMtools. *Bioinformatics*, 2009, 25.16: 2078-2079.

[52] BENTLEY, David R., et al. Accurate whole human genome sequencing using reversible terminator chemistry. *nature*, 2008, 456.7218: 53-59.

[53] JACKSON, Benjamin G.; SCHNABLE, Patrick S.; ALURU, Srinivas. Assembly of large genomes from paired short reads. In: *Bioinformatics and Computational Biology*. Springer Berlin Heidelberg, 2009. p. 30-43.

[54] MOHAMMED, Emad A.; FAR, Behrouz H.; NAUGLER, Christopher. Applications of the MapReduce programming framework to clinical big data analysis: current landscape and future trends. *BioData mining*, 2014, 7.1: 1.

[55] CHU, Cheng, et al. Map-reduce for machine learning on multicore. *Advances in neural information processing systems*, 2007, 19: 281.

[56] WHITE, Tom. *Hadoop: The definitive guide*. " O'Reilly Media, Inc.", 2012.

[57] SHVACHKO, Konstantin, et al. The hadoop distributed file system. In: *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)*. IEEE, 2010. p. 1-10.

[58] GOECKS, Jeremy; NEKRUTENKO, Anton; TAYLOR, James. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome biology*, 2010, 11.8: 1.

[59] SCHATZ, Michael C. CloudBurst: highly sensitive read mapping with MapReduce. *Bioinformatics*, 2009, 25.11: 1363-1369.

[60] GURTOWSKI, James; SCHATZ, Michael C.; LANGMEAD, Ben. Genotyping in the cloud with crossbow. *Current Protocols in Bioinformatics*, 2012, 15.3. 1-15.3. 15.

[61] LANGMEAD, Ben, et al. Searching for SNPs with cloud computing. *Genome biology*, 2009, 10.11: 1.

[62] PAPANICOLAOU, Alexie; HECKEL, David G. The GMOD Drupal bioinformatic server framework. *Bioinformatics*, 2010, 26.24: 3119-3124.

[63] SIVA, Nayanah. 1000 Genomes project. *Nature biotechnology*, 2008, 26.3: 256-256.

[64] SHERRY, Stephen T., et al. dbSNP: the NCBI database of genetic variation. *Nucleic acids research*, 2001, 29.1: 308-311.

[65] BARRETT, Tanya, et al. NCBI GEO: mining tens of millions of expression profiles—database and tools update. *Nucleic acids research*, 2007, 35.suppl 1: D760-D765.

[66] BARRETT, Tanya, et al. NCBI GEO: archive for functional genomics data sets—10 years on. *Nucleic acids research*, 2011, 39.suppl 1: D1005-D1010.

[67] KENT, W. James, et al. The human genome browser at UCSC. *Genome research*, 2002, 12.6: 996-1006.

[68] HAIDER, Syed, et al. BioMart Central Portal—unified access to biological data. *Nucleic acids research*, 2009, 37.suppl 2: W23-W27.

[69] KASPRZYK, Arek. BioMart: driving a paradigm change in biological data management. *Database*, 2011, 2011: bar049.

[70] BALL, Catherine A.; BRAZMA, Alvis. MGED standards: work in progress. *Omics: a journal of integrative biology*, 2006, 10.2: 138-144.

[71] BRAZMA, Alvis. Minimum information about a microarray experiment (MIAME)–successes, failures, challenges. *The Scientific World JOURNAL*, 2009, 9: 420-423.

[72] ASHBURNER, Michael, et al. Gene Ontology: tool for the unification of biology. *Nature genetics*, 2000, 25.1: 25-29.

[73] SCHRIML, Lynn Marie, et al. Disease Ontology: a backbone for disease semantic integration. *Nucleic acids research*, 2012, 40.D1: D940-D946.

[74] EILBECK, Karen, et al. The Sequence Ontology: a tool for the unification of genome annotations. *Genome biology*, 2005, 6.5: 1.

[75] SMITH, Barry, et al. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology*, 2007, 25.11: 1251-1255.

[76] KANEHISA, Minoru; GOTO, Susumu. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 2000, 28.1: 27-30.

[77] CROFT, David, et al. Reactome: a database of reactions, pathways and biological processes. *Nucleic acids research*, 2010, gkq1018.

[78] MAGRANE, Michele, et al. UniProt Knowledgebase: a hub of integrated protein data. *Database*, 2011, 2011: bar009.

[79] SZKLARCZYK, Damian, et al. STRING v10: protein–protein interaction networks, integrated over the tree of life. *Nucleic acids research*, 2014, gku1003.

[80] HAMOSH, Ada, et al. Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic acids research*, 2005, 33.suppl 1: D514-D517.

[81] The Cancer Genome Atlas. URL: http://cancergenome.nih.gov/

[82] ALEXANDROV, Ludmil B., et al. Signatures of mutational processes in human cancer. *Nature*, 2013, 500.7463: 415-421.

[83] GIBBS, Richard A., et al. The international HapMap project. *Nature*, 2003, 426.6968: 789-796.

[84] SHANNON, Paul, et al. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 2003, 13.11: 2498-2504.

[85] ROBINSON, James T., et al. Integrative genomics viewer. *Nature biotechnology*, 2011, 29.1: 24-26.

[86] NICOL, John W., et al. The Integrated Genome Browser: free software for distribution and exploration of genome-scale datasets. *Bioinformatics*, 2009, 25.20: 2730-2731.

[87] MCKENNA, Aaron, et al. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome research*, 2010, 20.9: 1297-1303.

[88] STEPHENS, Zachary D., et al. Big data: astronomical or genomical?. *PLoS Biol*, 2015, 13.7: e1002195.

[89] LI, Yixue; CHEN, Luonan. Big biological data: challenges and opportunities. *Genomics, proteomics & bioinformatics*, 2014, 12.5: 187-189.

[90] KOCH, Christof. Modular biological complexity. *Science*, 2012, 337.6094: 531-532.

[91] MACDONALD, Jeffrey R., et al. The Database of Genomic Variants: a curated collection of structural variation in the human genome. *Nucleic acids research*, 2014, 42.D1: D986-D992.

[92] LAPPALAINEN, Ilkka, et al. DbVar and DGVa: public archives for genomic structural variation. *Nucleic acids research*, 2013, 41.D1: D936-D941.

[93] MOCH, H., et al. Personalized cancer medicine and the future of pathology. *Virchows Archiv*, 2012, 460.1: 3-8.

[94] GEORGE, Lars. *HBase: the definitive guide*. " O'Reilly Media, Inc.", 2011.

[95] OLSTON, Christopher, et al. Pig latin: a not-so-foreign language for data processing. In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 2008. p. 1099-1110.

[96] NORDBERG, Henrik, et al. BioPig: a Hadoop-based analytic toolkit for large-scale sequence data. *Bioinformatics*, 2013, btt528.

[97] SCHUMACHER, André, et al. Scripting for large-scale sequencing based on Hadoop. In: *EMBnet. journal. The Next NGS Challenge Conference: Data Processing and Integration 14-16 May 2013, Valencia, Spain*. 2013.

[98] SHANAHAN, James G.; DAI, Laing. Large scale distributed data science using apache spark. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015. p. 2323-2324.

[99] WIEWIÓRKA, Marek S., et al. SparkSeq: fast, scalable, cloud-ready tool for the interactive genomic data analysis with nucleotide precision. *Bioinformatics*, 2014, btu343.

[100] OWEN, Sean, et al. Mahout in action. 2012.

[101] TEAM, R. Core, et al. R: A language and environment for statistical computing. 2013.

[102] SANNER, Michel F., et al. Python: a programming language for software integration and development. *J Mol Graph Model*, 1999, 17.1: 57-61.

[103] VAN ROSSUM, Guido, et al. Python Programming Language. In: *USENIX Annual Technical Conference*. 2007.

[104] COCK, Peter JA, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 2009, 25.11: 1422-1423.

[105] MUNGALL, Christopher J., et al. A Chado case study: an ontology-based modular schema for representing genome-associated biological information. *Bioinformatics*, 2007, 23.13: i337-i346.

[106] DE MACEDO, Jose Antonio Fernandes, et al. A conceptual data model language for the molecular biology domain. In: *Twentieth IEEE International Symposium on Computer-Based Medical Systems (CBMS'07)*. IEEE, 2007. p. 231-236.

[107] CHEN, Jake Yue; CARLIS, John V. Genomic data modeling. *Information Systems*, 2003, 28.4: 287-310.

[108] DEGTYARENKO, Kirill, et al. ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic acids research*, 2008, 36.suppl 1: D344-D350.

[109] PATO Ontology: https://bioportal.bioontology.org/ontologies/PATO

[110] ZANZONI, Andreas, et al. MINT: a Molecular INTeraction database. *FEBS letters*, 2002, 513.1: 135-140.

[111] HERMJAKOB, Henning, et al. IntAct: an open source molecular interaction database. *Nucleic acids research*, 2004, 32.suppl 1: D452-D455.

[112] HUNTER, Sarah, et al. InterPro: the integrative protein signature database. *Nucleic acids research*, 2009, 37.suppl 1: D211-D215.

[113] PEARL, Frances, et al. The CATH Domain Structure Database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis. *Nucleic acids research*, 2005, 33.suppl 1: D247-D251.

[114] GILLIS, Jesse; PAVLIDIS, Paul. “Guilt by association” is the exception rather than the rule in gene networks. *PLoS Comput Biol*, 2012, 8.3: e1002444.

[115] RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady. *Unified Modeling Language Reference Manual, The*. Pearson Higher Education, 2004.

[116] HUCKA, Michael, et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 2003, 19.4: 524-531.

[117] BRANDES, Ulrik, et al. *Graph markup language (GraphML)*. 2013.

[118] MASSEROLI, Marco, et al. GenoMetric Query Language: a novel approach to large-scale genomic data management. *Bioinformatics*, 2015, 31.12: 1881-1888.

[119] KAROLCHIK, Donna, et al. The UCSC genome browser database. *Nucleic acids research*, 2003, 31.1: 51-54.

[120] QUINLAN, Aaron R.; HALL, Ira M. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 2010, 26.6: 841-842.

[121] NEPH, Shane, et al. BEDOPS: high-performance genomic feature operations. *Bioinformatics*, 2012, 28.14: 1919-1920.

[122] KOZANITIS, Christos, et al. Using genome query language to uncover genetic variation. *Bioinformatics*, 2014, 30.1: 1-8.

[123] OVASKA, Kristian, et al. Genomic region operation kit for flexible processing of deep sequencing data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 2013, 10.1: 200-206.

[124] JI, Xiong, et al. 3D chromosome regulatory landscape of human pluripotent cells. *Cell Stem Cell*, 2016, 18.2: 262-275.

[125] BARABÁSI, Albert-László; GULBAHCE, Natali; LOSCALZO, Joseph. Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, 2011, 12.1: 56-68.

[126] CHO, Dong-Yeon; KIM, Yoo-Ah; PRZYTYCKA, Teresa M. Network biology approach to complex diseases. *PLoS Comput Biol*, 2012, 8.12: e1002820.

[127] LIU, Yang-Yu; SLOTINE, Jean-Jacques; BARABÁSI, Albert-László. Controllability of complex networks. *Nature*, 2011, 473.7346: 167-173.

[128] MILENKOVIĆ, Tijana, et al. Dominating biological networks. *PloS one*, 2011, 6.8: e23016.

[129] ZHONG, Quan, et al. Edgetic perturbation models of human inherited disorders. *Molecular systems biology*, 2009, 5.1: 321.

[130] GHIASSIAN, Susan Dina; MENCHE, Jörg; BARABÁSI, Albert-László. A DIseAse MOdule Detection (DIAMOnD) algorithm derived from a systematic analysis of connectivity patterns of disease proteins in the human interactome. *PLoS Comput Biol*, 2015, 11.4: e1004120.

[131] KÖHLER, Sebastian, et al. Walking the interactome for prioritization of candidate disease genes. *The American Journal of Human Genetics*, 2008, 82.4: 949-958.

[132] GHASEMI, Mahdieh, et al. Centrality measures in biological networks. *Current Bioinformatics*, 2014, 9.4: 426-441.

[133] HALLOCK, Peter; THOMAS, Michael A. Integrating the Alzheimer's disease proteome and transcriptome: a comprehensive network model of a complex disease. *Omics: a journal of integrative biology*, 2012, 16.1-2: 37-49.

[134] HIMMELSTEIN, Daniel S.; BARANZINI, Sergio E. Heterogeneous network edge prediction: a data integration approach to prioritize disease-associated genes. *PLoS Comput Biol*, 2015, 11.7: e1004259.

[135] KLINE, Rex B. *Principles and practice of structural equation modeling*. Guilford publications, 2015.

[136] PEPE, Daniele; GRASSI, Mario. Investigating perturbed pathway modules from gene expression data via structural equation models. *BMC bioinformatics*, 2014, 15.1: 1.

[137] ROSA, Guilherme JM, et al. Inferring causal phenotype networks using structural equation models. *Genetics Selection Evolution*, 2011, 43.1: 1.

[138] FAIRCHILD, Amanda J.; MACKINNON, David P. A general model for testing mediation and moderation effects. *Prevention Science*, 2009, 10.2: 87-99.

[139] TAKAHASHI, Hiromitsu; MATSUYAMA, Akira. An approximate solution for the Steiner problem in graphs. *Math. Japonica*, 1980, 24.6: 573-577.

[140] WINTER, Pawel; SMITH, J. MacGregor. Path-distance heuristics for the Steiner problem in undirected networks. *Algorithmica*, 1992, 7.1-6: 309-327.

[141] SADEGHI, Afshin; FRÖHLICH, Holger. Steiner tree methods for optimal sub-network identification: an empirical study. *BMC bioinformatics*, 2013, 14.1: 1.

[142] JAHID, Md Jamiul; RUAN, Jianhua. A Steiner tree-based method for biomarker discovery and classification in breast cancer metastasis. *BMC genomics*, 2012, 13.6: 1.

[143] DOWELL, Robin D., et al. The distributed annotation system. *BMC bioinformatics*, 2001, 2.1: 1.

[144] YATES, Andrew, et al. Ensembl 2016. *Nucleic acids research*, 2016, 44.D1: D710-D716.

[145] AKEN, Bronwen L., et al. The Ensembl gene annotation system. *Database*, 2016, 2016: baw093.

[146] TWEEDIE, Susan, et al. FlyBase: enhancing Drosophila gene ontology annotations. *Nucleic acids research*, 2009, 37.suppl 1: D555-D559.

[147] LI, Guoliang, et al. ChIA-PET tool for comprehensive chromatin interaction analysis with paired-end tag sequencing. *Genome biology*, 2010, 11.2: 1.

[148] The 3D Genome Browser: http://www.3dgenome.org.

[149] LÉVY-LEDUC, Celine, et al. Two-dimensional segmentation for analyzing Hi-C data. *Bioinformatics*, 2014, 30.17: i386-i392.

[150] SERVANT, Nicolas, et al. HiTC: exploration of high-throughput ‘C’ experiments. *Bioinformatics*, 2012, 28.21: 2843-2844.

[151] LUN, Aaron TL; SMYTH, Gordon K. diffHic: a Bioconductor package to detect differential genomic interactions in Hi-C data. *BMC bioinformatics*, 2015, 16.1: 258.

[152] ENCODE Consortium portal: https://www.encodeproject.org/

[153] PRÜFER, Kay, et al. The bonobo genome compared with the chimpanzee and human genomes. *Nature*, 2012, 486.7404: 527-531.

[154] WANG, Zhong; GERSTEIN, Mark; SNYDER, Michael. RNA-Seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 2009, 10.1: 57-63.

[155] CONESA, Ana, et al. A survey of best practices for RNA-seq data analysis. *Genome biology*, 2016, 17.1: 1.

[156] CORE, Leighton J.; WATERFALL, Joshua J.; LIS, John T. Nascent RNA sequencing reveals widespread pausing and divergent initiation at human promoters. *Science*, 2008, 322.5909: 1845-1848.

[157] ROBINSON, Mark D.; MCCARTHY, Davis J.; SMYTH, Gordon K. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 2010, 26.1: 139-140.

[158] ROBINSON, Mark D.; SMYTH, Gordon K. Small-sample estimation of negative binomial dispersion, with applications to SAGE data. *Biostatistics*, 2008, 9.2: 321-332.

[159] ROBINSON, Mark D.; OSHLACK, Alicia. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome biology*, 2010, 11.3: 1.

[160] CROSETTO, Nicola, et al. Nucleotide-resolution DNA double-strand break mapping by next-generation sequencing. *Nature methods*, 2013, 10.4: 361-365.

[161] ZENG, Xin, et al. jMOSAiCS: joint analysis of multiple ChIP-seq datasets. *Genome biology*, 2013, 14.4: 1.

[162] BURNHAM, Kenneth P.; ANDERSON, David. Model selection and multi-model inference. *A Pratical informatio-theoric approch. Sringer*, 2003.

[163] WEDDERBURN, Robert WM. Quasi-likelihood functions, generalized linear models, and the Gauss—Newton method. *Biometrika*, 1974, 61.3: 439-447.

[164] COX, David Roxbee; SNELL, E. Joyce. *Analysis of binary data*. CRC Press, 1989.

[165] SUGIURA, Nariaki. Further analysts of the data by akaike's information criterion and the finite corrections: Further analysts of the data by akaike's. *Communications in Statistics-Theory and Methods*, 1978, 7.1: 13-26.

[166] FAIRCHILD, Amanda J.; MACKINNON, David P. A general model for testing mediation and moderation effects. *Prevention Science*, 2009, 10.2: 87-99.

[167] QUINLAN, J. Ross. Induction of decision trees. *Machine learning*, 1986, 1.1: 81-106.

[168] DENG, Houtao; RUNGER, George; TUV, Eugene. Bias of importance measures for multi-valued attributes and solutions. In: *International Conference on Artificial Neural Networks*. Springer Berlin Heidelberg, 2011. p. 293-300.

[169] LIAW, Andy; WIENER, Matthew. Classification and regression by randomForest. *R news*, 2002, 2.3: 18-22.

[170] LÓPEZ-RATÓN, Mónica, et al. OptimalCutpoints: an R package for selecting optimal cutpoints in diagnostic tests. *J Stat Softw*, 2014, 61.8: 1-36.

[171] HEINZ, Sven, et al. Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. *Molecular cell*, 2010, 38.4: 576-589.

[172] FERRARI, Raffaele, et al. Frontotemporal dementia and its subtypes: a genome-wide association study. *The Lancet Neurology*, 2014, 13.7: 686-699.

[173] LI, Miao-Xin, et al. GATES: a rapid and powerful gene-based association test using extended Simes procedure. *The American Journal of Human Genetics*, 2011, 88.3: 283-293.

[174] WU, Michael C., et al. Rare-variant association testing for sequencing data with the sequence kernel association test. *The American Journal of Human Genetics*, 2011, 89.1: 82-93.

[175] MA, Shuangge; DAI, Ying. Principal component analysis based methods in bioinformatics studies. *Briefings in bioinformatics*, 2011, 12.6: 714-722.

[176] FERRARI, Raffaele, et al. A genome-wide screening and SNPs-to-genes approach to identify novel genetic risk factors associated with frontotemporal dementia. *Neurobiology of aging*, 2015, 36.10: 2904. e13-2904. e26.

[177] PEPE, Daniele; PALLUZZI, Fernando; GRASSI, Mario. Sem Best Shortest Paths for the Characterization of Differentially Expressed Genes. In: *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*. Springer International Publishing, 2014. p. 131-141.

[178] PEPE, Daniele; PALLUZZI, Fernando; Perturbed pathway communities for the characterization of putative disease modules; *ECCB/ISCB 2015; DOI: 10.*7490/f1000research.1110064.1

[179] YOSHIHARA, K., et al. The landscape and therapeutic relevance of cancer-associated transcript fusions. *Oncogene*, 2014.

[180] AMBROSIO, Susanna, et al. Cell cycle-dependent resolution of DNA double-strand breaks. Oncotarget, 2016, 7.4: 4949.

[181] Salvi E, Kutalik Z, Glorioso N, Benaglio P, Frau F, Kuznetsova T, Arima H, Hoggart C, Tichet J, Nikitin YP, Conti C, Seidlerova J, Tikhonoff V, Stolarz-Skrzypek K, Johnson T, Devos N, Zagato L, Guarrera S, Zaninello R, Calabria A, Stancanelli B, Troffa C, Thijs L, Rizzi F, Simonova G, Lupoli S, Argiolas G, Braga D, D'Alessio MC, Ortu MF, Ricceri F, Mercurio M, Descombes P, Marconi M, Chalmers J, Harrap S, Filipovsky J, Bochud M, Iacoviello L, Ellis J, Stanton AV, Laan M, Padmanabhan S, Dominiczak AF, Samani NJ, Melander O, Jeunemaitre X, Manunta P, Shabo A, Vineis P, Cappuccio FP, Caulfield MJ, Matullo G, Rivolta C, Munroe PB, Barlassina C, Staessen JA, Beckmann JS, Cusi D. *Genomewide association study using a high-density single nucleotide polymorphism array and case-control design identifies a novel essential hypertension susceptibi lity locus in the promoter region of endothelial NO synthase*. Hypertension. 2012 Feb;59(2):248-55. Epub 2011 Dec 19. doi: 10.1161/HYPERTENSIONAHA.111.181990. PubMed PMID: 22184326; PubMed Central PMCID: PMC3272453.

URLs:

[U1] http://genetics.bwh.harvard.edu/pph/FASTA.html
[U2] https://en.wikipedia.org/wiki/FASTQ_format#cite_note-Cock2009-1
Cock, P. J. A.; Fields, C. J.; Goto, N.; Heuer, M. L.; Rice, P. M. (2009). "The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants". Nucleic Acids Research. 38 (6): 1767–1771. doi:10.1093/nar/gkp1137. PMC 2847217. PMID 20015970.

[U3] https://en.wikipedia.org/wiki/Chromatin
[U4] https://blast.ncbi.nlm.nih.gov/Blast.cgi
[U5] https://genome.ucsc.edu/FAQ/FAQblat.html#blat3
[U6] https://sourceforge.net/projects/bfast/
[U7] http://bowtie-bio.sourceforge.net/index.shtml
[U8] http://bio-bwa.sourceforge.net/
[U9] http://maq.sourceforge.net/maq-man.shtml
[U10] http://mrfast.sourceforge.net/
[U11] http://compbio.cs.toronto.edu/shrimp/
[U12] http://www.sanger.ac.uk/science/tools/smalt-0
[U13] http://soap.genomics.org.cn/soapaligner.html
[U14] http://www.sanger.ac.uk/science/tools/ssaha2-0
[U15] http://www.bcgsc.ca/platform/bioinfo/software/abyss
[U16] http://amos.sourceforge.net/wiki/index.php/AMOScmp-shortReads-alignmentTrimmed
[U17] https://sourceforge.net/p/mira-assembler/wiki/Home/
[U18] http://sharcgs.molgen.mpg.de/
[U19] http://soap.genomics.org.cn/soapdenovo.html
[U20] https://omictools.com/ssake-tool
[U21] https://sourceforge.net/projects/vcake/
[U22] http://www.ebi.ac.uk/~zerbino/velvet/