1 Introduction and Outline

1.1 Background

Bioinformatics, the blend of computer science and biology, has played a dominant role in almost all major discoveries in the fields of genetics and subsequently in life sciences over the last decade. The main reason for this lies in the very nature of modern genetics research. The latest technological advancements have allowed the massive profiling of genetic data. These data include gene expression, genotypes, DNA-sequencing, and RNA-sequencing to name just a few. The quantity, complexity and heterogeneous nature of these data are such that handling, storage and, most importantly, their processing require advanced computational techniques. The need to develop these techniques is the reason why bioinformatics have become a vital part of modern genetic research.

Bioinformatics began as an experimental field that provided auxiliary help to geneticists, mainly for tackling algorithmic and organizational research tasks [22]. As genetics has become a more data-driven science, its reliance on bioinformatics has steadily increased. The level of this dependence is such that today we are experiencing two unprecedented events. The first is that although bioinformatics and genetics are tightly intertwined, with advances in one giving rise to progress in the other, today it is the bioinformatics that is guiding the progress in genetics [15]. Of course, the opposite also happens but on a smaller scale. The reason for this is that the computational challenges of genetic data are on the leading edge of computer science and technology in general [32]. It is not an exaggeration to claim that modern computer infrastructure is just not powerful enough to qualitatively process the amount of genetic data being generated on a daily basis. This is mainly because the pace of progress in sequencing technology is far higher than progress in computer technology [33], [24]. Moreover, there are also considerations such as noise, quality assurance and data provenance.

The realization of these shortcomings has led to the second unprecedented event. Because of the extreme computational demand of many genetic experiments, there are bioinformatics considerations in the formulation of genetic research hypotheses [15]. Thus, the complete intellectual process of shaping new ideas that might shed new light on the understanding of genetic regulation and disease mechanisms is now governed,
and sometimes hindered, by bioinformatics limitations. For example, the files that contain the complete nucleotide sequence of the genome (Whole Genome Sequencing) of a single sample at an adequate quality level (usually at 30X coverage which is the average number that each nucleotide has been sequenced) are approximately 350GB [21]. These files are enough to completely fill the hard drive of a commodity computer. Consequently, regardless of the low sequencing cost, a research team has to employ advanced data management techniques in order to test hypotheses that require the sequencing of even a modest number of samples [3]. These techniques may be more expensive than the sequencing. Another example can be seen in hypotheses that require either the integration of multiomics datasets or to locate multiple genetic factors that simultaneously play a role in complex diseases. In the first case by adding an extra layer of data that corresponds to a more complete picture of the disease, the dimensionality of the problem increases [6]. In the second, the search space of possible combinations of multiple genetic factors is so big that even modern techniques require extremely long computational time [19] (assuming of course that there are adequate numbers of samples). In both cases, the existing statistical analysis methods lie at the leading edge of modern bioinformatics research and the specifics of current implementations largely determine the formulation of currently testable hypotheses.

These are some of the main points that underline the significance of bioinformatics in genetics research and in healthcare in general. It also makes obvious that even small enhancements in the design of a bioinformatics pipeline can have enormous impact on the progress of a genetic experiment [5]. So far, bioinformatics have managed to engineer solutions for extremely challenging problems at such level that we can confidently state that the genomic revolution now lies in the past and that we are entering the post-genomic era. So despite this seemingly dark present picture, we can envision that bioinformatics will continue to be at the forefront of scientific discovery in the life sciences. But in order for this to happen we first need to analyze the inner characteristics and the philosophy behind the development of current bioinformatics pipelines, locate points for improvement, and attempt to implement them.

1.2 Bioinformatics done right

1.2.1 Maturity model

If we attempt to divide the history of bioinformatics so far, we can see three distinct generations. These generations, briefly described in Table 1.1, were formed mainly by the computational needs of the research society through time. The first is the ‘algorithmic’ generation, where special algorithms needed to be crafted for computation
problems inherent in genetics. For example, one of the most demanding computa-
tional needs at that time was how to efficiently compare a genomic sequence with a
database of available sequences. This gave rise to the Needleman–Wunsch algorithm
(1970) that performed exact sequence alignment (or else global alignment). This was
followed with the introduction of the Smith–Waterman algorithm (1981), which is a
variation of Needleman-Wunsch that allowed sequence mismatches (local alignment).
The Smith–Waterman algorithm also allowed for the computational detection of se-
quence variants (mainly insertions and deletions). It is interesting that the demanding
time requirements of these algorithms gave rise to the prominent BLAST and FASTA
algorithms (1985) that are approximately 50 times faster (although they produce
suboptimal results). Today these algorithms are available either as online web services
or programs implemented in low-level programming languages that make optimal uti-
лизation of the underlying computer architecture. These algorithms and their variations
are the foundational components of modern Next Generation Sequencing techniques.

The second generation appeared as a need not so much to implement new algorithms
but rather to manage large amounts of heterogeneous data [2]. The starting point of
this generation was the Human Genome Project (HGP) [23] that lasted from 1990
until 2003 and delivered the first almost complete (92%) sequence of human DNA.
At the cost of approximately 3 billion dollars, it remains until today the largest
scientific collaboration in biology. HGP managed to bring together 20 sequencing
research facilities from 6 countries, although more than 200 labs from 18 different
countries participated in the analysis. The successful outcome of HGP was also based on
generating and establishing common analysis protocols and data formats. Additionally,
HGP pioneered the establishment of initial public data release policies and brought
forward ethical, legal and social issues of genetics. This was also the era that large
datasets started becoming public, mainly from large organizations (i.e. EMBL at
Europe and NCBI in the United States). New notions appeared in bioinformatics
such as security, data sharing, data modeling, distributed computing and web services.
HGP succeeded in not only filling one of the most eminent gaps in biology, that of
the sequence of the human genome, but also in paving the way for other projects to
continue its work. Right after the end of HGP, the HapMap project [11] was started
(in October 2002) with the goal of inferring the haplotype structure of the human
genome. The haplotype structure contains information regarding the regions of the
genome that tend to be co-inherited and recombination rarely happens. With this
information any genetic variant that is known (or suspected) to be involved in a disease,
can be associated with a complete region, thus offering many ways of investigating
additional variants or even mechanisms of action. To accomplish this, the HapMap
project genotyped 1184 individuals belonging to 11 different populations, investigating
more than 3.1 million SNPs (Single Nucleotide Polymorphisms) [17]. The project released data in three phases (2005, 2007 and 2009) and since the cost of sequencing technology (at that time) was prohibitive for population studies of this size, it mainly used genotyping techniques. The main difference between genotyping and sequencing is that genotyping is a technique to assess the combination of alleles that an individual has at a specific site, whereas sequencing reveals the complete nucleotide sequence of the individual’s DNA. Despite being more affordable, the main drawback of genotyping is that it requires a priori the knowledge of the genotyped location. Therefore, as a technology, it has a limited value in investigating novel variants. Fortunately, by 2008, sequencing had advanced well enough to be used in a large population study. The 1000 Genomes Project [10] (1KGP) was launched at that time to investigate human genetic variation by employing mainly sequencing techniques. 1KGP was revolutionary in many aspects. It sequenced 1092 individuals from 26 different population regions, covering a significant part of the world. It practically tripled the number of known variants of the human genome (from 3.1 million to 10 million), increased the analysis of the haplotype structure, and provided insights regarding the evolutionary processes that guided the distribution of these variants. Additionally, it used many sequencing technologies (whole-genome, exon-targeted) on varying platforms and also included trios (mother-father-child). At the peak of 1KGP, it was sequencing 10 billion bases per day, which is more than 2 whole human genomes. After the completion of 1KGP (in 2012), biologists at last had a comprehensive catalogue of genetic variants for various populations and also a tested set of software, pipelines, benchmarks and good practices for conducting large-scale sequencing studies. In parallel, the completion of HGP sparked many investigations regarding the functional effect of these variants. Perhaps the most notable effort towards this is the ENCODE project [9], which is an ongoing international study into the functional genomic elements of DNA. This project is building a comprehensive functional map of human DNA. For example, it annotates regions that are transcribed in RNA, regions that are mostly transcribed in particular cell types, and also regions that are translated into certain types of proteins.

Apart from being milestone contributions to genetics, these projects have also pioneered the open philosophy behind data publication and the willingness to receive feedback from the research community by publishing data and methods even at an early stage of the experiments. Nevertheless, the greatest contribution of these projects may be the huge momentum that they have created: many national organizations have designed experiments to investigate the genetic identity and diversity of their populations. Today (2017), there are more than 20 countries with national sequencing projects in various stages of completeness [16, 13]. These countries cover 29% of the world population (18% is from China) and as this percentage grows we can expect the
genetic structure of nearly the whole human race to become known in the next few decades.

From these various projects, the Genome of the Netherlands [31] (GoNL) stands out. Although GoNL was a Dutch project that started in July 2009, only 1.5 years after the start of 1KGP, it represented a significant advance in population sequencing projects. Despite having approximately the same number of samples, it used better sequencing quality (\(\sim 12 \times\) coverage) than 1KGP (\(\sim 3 \times\) coverage). Additionally, GoNL’s samples were composed of only related individuals: specifically it contained 231 trios (mother, father, child), 8 quartos (mother, father, 2 siblings) with dizygotic twins, and 11 quartos with monozygotic twins. The relatedness information (i.e kinship) is a valuable resource that helps to validate the sequencing information. For example, a nucleotide, say ‘C’, at a specific location in the child’s genome should also be evident in the same location of at least one of the parents’ genomes (except for the highly unlikely event of a de novo mutation). But most importantly, by exploiting the relatedness information we can deduce the haplotype structure of the population in superb quality. Finally, all the samples from four different Dutch regional biobanks were pooled. These biobanks contained extensive phenotype information and additional biological samples for all the participating individuals. We can confidently state that if 1KGP demonstrated how to perform genetic studies on international populations, then GoNL demonstrated how to study the genetic structure on a national scale by applying high quality and cost-effective methods.

It is interesting to take a look at the progression of the order of magnitude of included samples by looking at some high-profile national projects. In March 2010 the UK10K project [18] was set up to sequence 10,000 individuals of UK descent whereas in June 2017 the Precision Medicine Initiative project [1] in the United States started enrolling the first of the 1,000,000 million planned individuals.

The main ‘product’ of this second generation phase is the wealth of knowledge that these projects have created. From a bioinformatics point of view, this generation also produced an extremely rich set of open-access data, tools, pipelines and web services. Today, it is difficult to list even the main products of this generation since they range from files with identified genetic variants (e.g. VCF files from GoNL), simple portals (e.g. ArrayExpress, PubMed, dbSNP), genomic browsers (e.g. UCSC, ENSEMBL) and generic web platforms (e.g. MOLGENIS) to name just a few. A thorough discussion of the quantity and quality of existing data, tools and pipelines is given in Chapter 2.

We are currently experiencing the transition from the second generation to a new third generation. As with the first two generations, there are new computational needs that cannot be fulfilled by the existing ‘schools of thought’. The new necessities are open and reproducible protocols, methods and data. We also require more social and
more ‘extrovert’ behavior from researchers at all stages of development (maybe even beforehand, from the fund-raising stage through crowd-funding [8]). Although these requirements sound more behavioral, they bear particular computational demands. These are: the adoption of more readable programming languages, a minimum dependence on proprietary software in favor of open source implementations, good data stewardship, and a smooth transition towards nationwide or international computer infrastructures, such as the Grid and the cloud.

<table>
<thead>
<tr>
<th>Generations</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithmic</td>
<td>Build essential algorithms</td>
</tr>
<tr>
<td></td>
<td>Introduce data mining, AI, machine learning, visualization</td>
</tr>
<tr>
<td></td>
<td>Biostatistics</td>
</tr>
<tr>
<td></td>
<td>Introduce terminology/nomenclatures</td>
</tr>
<tr>
<td></td>
<td>Moderate computational needs</td>
</tr>
<tr>
<td></td>
<td>Languages: C, C++, Fortran, Pascal</td>
</tr>
<tr>
<td>Data Management</td>
<td>Web services</td>
</tr>
<tr>
<td></td>
<td>Large/open datasets</td>
</tr>
<tr>
<td></td>
<td>Build formats/models/ontologies</td>
</tr>
<tr>
<td></td>
<td>Data exchange</td>
</tr>
<tr>
<td></td>
<td>Introduce databases, software engineering to genetics</td>
</tr>
<tr>
<td></td>
<td>Compute on local clusters</td>
</tr>
<tr>
<td></td>
<td>Languages: Java, C#, Perl, R, DSL</td>
</tr>
<tr>
<td>Social</td>
<td>Data provenance</td>
</tr>
<tr>
<td></td>
<td>Open code/data/articles</td>
</tr>
<tr>
<td></td>
<td>Reproducibility/Readability</td>
</tr>
<tr>
<td></td>
<td>Social coding/Crowdsourcing</td>
</tr>
<tr>
<td></td>
<td>Introduce Big Data to genetics</td>
</tr>
<tr>
<td></td>
<td>Compute on Grid/Cloud</td>
</tr>
<tr>
<td></td>
<td>Languages: Python, Ruby, GO, Scala</td>
</tr>
</tbody>
</table>

Table 1.1: The main characteristics of the three generations of bioinformatics development.
1.2.2 Moving forward

Although bioinformatics have matured and many lessons have been learned during these generations [14], there are still many points to improve. We discuss four points below.

As a first consideration, today we have repositories that host or describe thousands of tools and databases in the bioinformatics domain. Additionally, there are more than 100 meta-tools to ease the integration and inter-connection of these tools in complex analysis pipelines. Yet, in most cases, this bridging requires extensive technical knowledge that can be done only by skilled IT (Information Technology) personnel [34]. Unless this bridging becomes easy enough for a geneticist or biologist with basic IT knowledge to implement it, this vibrant landscape will become fragmented and hostile towards innovative and ground-breaking ideas. There is therefore a prominent need for technologies that can bring the very people who generate the research hypotheses into the foreground, rather than have them depend on an unfamiliar technical domain.

Second, today we can be assured that open source is finally the norm for code distribution. Nevertheless, open source does not automatically make the research community favor a certain tool. Apart from easy access, an open source tool is not guaranteed to have readable source code, tests, documentation, and a team or community able to provide support [36]. Moreover, the development plans and roadmap of the software are not always known, leading to qualitative implementations that are abandoned and forgotten, for example, or to good ideas with great potential that the community hesitates to support for fear of future project discontinuation.

Third, the main criterion for career advancement in fields like bioinformatics right now is ultimately, the number of papers published and of their citations. The existing scientific publishing status quo has a limited interest in the open practices of the published solutions. Besides the obvious consequences [35], without this incentive, researchers perceive the adoption of open policies as an extra burden, even though this adoption can be highly beneficial either for the source code [4], the papers written [26] or the research data [30], [27].

A final fourth point has to do with the infrastructure. Developers often think locally when they implement their methods, tending to build solutions for the architecture of their laboratory or for the version of the problem that their fellow geneticists have formed. Adhering to open practices also includes developing generic solutions in common environments. This is one of the reasons why developing for the cloud or for a nationwide grid infrastructure should be a priority for modern bioinformaticians.

By overcoming these shortcomings, bioinformatics can fully enter into the emergent third generation. Moreover, these open challenges should be further polished, crystalized
and objectified into a set of principles that, at least for this thesis, can be called ‘bioinformatics done right’. These are the main principles that guide the creation of the methods presented in this thesis.

1.3 Integration as a service in genetics research

Integration is the key to overcoming the presented shortcomings and to reap all the potential benefits from the emerging third generation of bioinformatics. In this context, integration can be defined as the process of efficiently bridging together tools, data, research communities, and different computation resources for the purpose of conducting innovative science. The European Bioinformatics Institute (EBI) set two major goals for 2016; the first was data growth and the second was integration [12]. Data growth is a sine qua non objective for any major bioinformatics institute and relies mostly on the adaptation of modern sequencing (and other mass profiling) technologies. In contrast, integration entails some additional and intricate challenges.

In general, the major purpose of integration is to provide solutions that offer high performance, efficiency and usability as a result of a sophisticated combination of existing components. Alternatively, the purpose is to prove that “the whole is greater than the sum of its parts”. This realization has also been pinpointed as the anticipated philosophy that should govern most future studies [7], [37]. Apart from the issues presented in section 1.2.2, the proper integration of existing methods can help tremendously in overcoming problems like limited tool discoverability, merging of heterogeneous data and battling the replication crisis. Therefore, one crucial question in this regard is how can: (1) modern tool developers, (2) data curators, and (3) Workflow Management System engineers alter their practices in order to augment the connectivity and share-ability of their resources. To achieve this, the bioinformatics community should agree on adhering to some basic and easily applicable development guidelines and good practices for each one of these groups. In order to do this, we first need to simply observe and note some of the common pitfalls and omissions that impede the process of integration. Subsequently, we need to distill these observations into clear and easy-to-follow guidelines and we should also underline how and to what extent the bioinformatics community can benefit by adopting them. This is perhaps the most important challenge - it basically lays the groundwork on which subsequent challenges have to be addressed.

One of these challenges is to properly and sufficiently communicate all the minor and major information that is essential for the seamless connection of different tools and data that comprise a bioinformatics pipeline. This is the Documentation challenge. This
effort becomes more challenging if the pipeline contains complicated, error-prone or very resourceful components. This description has to cover all the technical instructions, including troubleshooting directions, system requirements and methods to measure the adequacy of the input and the quality of the results. Moreover, the instructions should be clear enough to cover a target audience with a wide range of IT and genetics literacy. An example of a pipeline of this kind is genotype imputation [29], which is the process of stochastically enriching the set of markers of a genotype experiment with additional markers from a more densely sequenced or genotyped population panel.

A subsequent challenge is Wrapping, or how to offer ‘one-stop’ solutions to bioinformatics pipelines that have been sufficiently described in prior studies. The objective in this case is to provide a single, open-source tool that wraps all the software, installation scripts, and system configuration commands. This tool should undertake the tasks of setting up an environment, installation of software, applying quality control on input data, optimally splitting the processing tasks into executable jobs, submission of jobs on a wide selection of high-performance computing (HPCs), and finally assessing the quality of results. The design philosophy of this tool should be user friendly, highly customizable, documented and easily embed-able in an external pipeline for upstream and downstream analysis.

An additional consideration is Collaboration, which covers user involvement and engagement. The challenge in this case is to offer an online environment that promotes user collaboration and provides incentives to a diverse community of bioinformaticians to participate in creating, editing, documenting and testing bioinformatics pipelines. A pioneering method to achieve this objective is through ‘crowdsourcing’ [20, 25]. Crowdsourcing is the process where an online, loosely-coupled community collaborates in order to achieve a cumulative result, with absent or little central moderation. User participation takes the form of ‘edits’. Each edit can be anything from a minor typographic correction, a simple reference addition or a major contribution that immensely improves the result’s quality. Perhaps the most successful example of the crowdsourcing concept is Wikipedia, which contains, on average, high quality encyclopedic articles for almost all areas of interest. Therefore, an open question is if crowdsourcing can also be used for the purpose of creating high quality bioinformatics computational solutions. A crowdsourced bioinformatics resource environment could serve concurrently the purposes of a repository, a content management system, a workflow management system, an execution environment, and also a social experiment to test how the bioinformatics community can collaborate under the ‘wiki’ philosophy [28].

A common issue that usually rises during the process of building a bioinformatics pipeline is when there is an abundance of available tools for a given task, all produce
good results for some part of the input data but none is able to process all the possible variations of the input. This problem is usually coupled with the facts that the input is not well defined and that the community rarely uses formal directions or even good practices when generating these data. I call this the *Composition* challenge. A pipeline that attempts to tackle a problem of this category has to apply strict quality checks but also has to be permissive in erroneous input. Additionally, it should treat the component tools as an inventory and use the correct combination of tools for different kinds of input. Therefore, one challenge is to overcome the existing perceptions of pipelines as simple and rigid input/output connections between different tools. On the contrary pipelines should contain inherent logic (even sometimes fuzzy logic) that performs the optimal tool combination and alleviates possible incomplete tool implementation or even erroneous input data.

In Figure 1.1, I present a visualization of these challenges as pieces of a puzzle. By assembling this puzzle we can transform a simple analysis workflow into an active component that is fully immersed in the upcoming third generation of bioinformatics. In this Ph.D. thesis, I present some well-grounded approaches to these challenges that are being used to tackle various problems in genetics. Of course, since this generation is currently under formation, there may be additional considerations I am unaware of, I therefore expect future studies will complement my work by identifying and describing them.
Figure 1.1: The main Integration techniques presented in this Ph.D. thesis. On top there is a typical view of an analysis workflow (Data → Tool → Results). Making this workflow as integrate-able as possible for the wider bioinformatics community entails four basic challenges. These are the availability of proper Documentation, the inclusion of methods that Wrap the workflow in an embeddable one-step solution, the smart Composition of available tools that handle incomplete implementations or erroneous data, and the adoption of receptive policies that enhance user Collaboration.

1.4 Outline

Below I outline the chapters in this thesis, the particular research questions addressed, and how the chapters are related.

In Chapter 2, I present a set of practical and easy-to-follow best practices and guidelines for tool developers, data curators and Workflow Management System engineers. These guidelines can help these groups to significantly augment the re-usability,
connectivity and user friendliness of their Research Objects. This chapter also includes a discussion of the expected benefits that the wider biomedical community can gain by following these guidelines.

In Chapter 3, I propose a modern bioinformatics protocol for the imputation of genetic data. I discuss all the practical considerations and present good practices that should be applied when performing this task.

In Chapter 4, I present a bioinformatics implementation of the good practices of chapter 3 in a new imputation pipeline. The pipeline has been designed so that it requires minimum effort from a user to install and submit it in a High Performance Computation (HPC) environment. It is also open and easily adapted to external workflows and HPCs. The solution uses Molgenis-compute as a pipeline management system.

In Chapter 5, I discuss a perspective on the characteristics and the problems of modern bioinformatics pipelines. Specifically, I concentrate on the reproducibility and openness of published methods and I suggest that crowdsourcing can improve existing solutions. I propose adopting PyPedia, which is a wiki platform for the development of bioinformatics pipelines with the python language. To demonstrate its use, I have implemented parts of the imputation pipeline using PyPedia.

In Chapter 6, I address a problem that is more often seen in the area of clinical genetics. The majority of newly discovered mutations are presented in the scientific literature in one of the oldest and most widely used nomenclatures in genetics, called HGVS(Human Genome Variation Society). Although the main purpose of HGVS is the unambiguous and concise reporting of mutations, many authors disregard the official reporting guidelines. This renders some of the reported mutations ambiguous and hinders their location and validation in existing NGS studies. To remedy this problem I implemented MutationInfo, which combines 11 different tools in an exhaustive pipeline in order to locate the chromosomal position of an HGVS mutation with the highest possible confidence.

Finally in Chapter 7, I present some concluding notes and remarks.
Bibliography


