

DOTTORATO DI RICERCA IN
Oncologia e Patologia Sperimentale
Ciclo XXVII°

Settore Concorsuale di afferenza: 06/A2
Settore Scientifico disciplinare: MED/04

***Data management and data analysis in the large
European projects GEHA (GEnetics of Healthy
Aging) and NU-AGE (NUtrition and AGEing):
a bioinformatic approach***

Presentata da:
Dottor Dario Vianello

Coordinatore Dottorato
Prof. Pier Luigi Lollini

Relatore
Prof. Stefano Salvioli

Correlatore
Prof. Claudio Franceschi

*To all the people
who put up with me
in this very long journey.*

Science may set limits to knowledge,
but should not set limits to
imagination.

Bertrand Russell (1872 - 1970)

CONTENTS

1	INTRODUCTION	1
1.1	Systems biology of aging	1
1.2	The NU-AGE Project	2
1.2.1	NU-AGE technical infrastructure	3
1.3	Genetics, a focus on	5
1.3.1	Humans as metaorganisms	5
1.3.2	Mitochondria & Mitochondrial genome	5
1.3.3	Mitochondrial genome	6
1.3.4	mtDNA and aging	7
1.3.5	The GEHA study	7
1.3.6	The interacting mitochondrion	7
2	AIM OF THE STUDY	9
3	MATERIALS AND METHODS	11
3.1	Data	11
3.1.1	NU-AGE Data	11
3.1.2	GEHA Data	12
3.1.3	DbSNP Data	13
3.1.4	Eurostat Data	13
3.2	Programming Languages and Libraries	13
3.2.1	Python	14
3.2.2	Python Libraries and Packages	14
3.2.3	IPython Notebook	17
3.2.4	R	19
3.3	Databases and Web Interfaces	19
3.3.1	Text files	19
3.3.2	SQLite	20
3.3.3	PostgreSQL	20
3.3.4	Dbnp	20
3.3.5	LabKey	20
3.4	Programs and algorithms	21
3.4.1	Phred, Phrap and Consed	21
3.4.2	ClustalW2 and Clustal Omega	22
3.4.3	FastQC	22
3.4.4	Swiss-army knives for genomics data management	23
3.4.5	GATK	23
3.4.6	HaploFind	24
3.4.7	EpiData Entry	24
4	RESULTS	25
4.1	NU-AGE - the data challenge	25
4.1.1	The wealth of data	26
4.1.2	Why we must fight errors	28
4.1.3	How to fight errors	28
4.1.4	Striving to document	29
4.2	NU-AGE - Validating the data	29
4.2.1	Data entry interface	29
4.2.2	The need of a new solution	34
4.2.3	NU-AGE Validator - a quick introduction	34
4.2.4	NU-AGE Validator - The implementation	35
4.2.5	NU-AGE Validator - How it works	47
4.2.6	Validating continuous data, without fixed boundaries	48
4.2.7	Crunching everything - document to rule them all	50
4.3	NU-AGE - The @Bologna database	51
4.3.1	Why?	51
4.3.2	Which database?	51

4.3.3	Database deployment	51
4.3.4	Database description	52
4.4	NU-AGE Systems Biology	52
4.4.1	Where to start the journey from	53
4.5	GEHA and the Epistasis	64
4.5.1	Obtaining the sequences	64
4.5.2	Analyzing them	65
4.5.3	Single SNP association	69
4.5.4	Multiple SNPs association - Epistasis	70
4.6	Ion Torrent Data Analysis, an excursus of	75
5	DISCUSSION	81
5.1	The Systems Biology challenges	81
5.2	The NU-AGE study	82
5.2.1	Data management	82
5.2.2	Preliminary data analysis	83
5.3	The GEHA Project	84
5.3.1	Background and methods setup	85
5.3.2	Single variants association	86
5.3.3	Multiple variants association - Epistasis	86
5.4	Next Generation Sequencing data analysis	89
6	CONCLUSIONS	91
	Author's publications	93
	References	95

LIST OF FIGURES

Figure 1	Map of the subjects enrolled by NU-AGE	4
Figure 2	Mitochondrial DNA molecule	6
Figure 3	IPython Notebook folder	18
Figure 4	IPython Notebook screenshot	18
Figure 5	FastQC Screenshot	23
Figure 6	Validator repository size in line of codes in time	36
Figure 7	Example of test PDF report produced by PDFWrapper	46
Figure 8	Example of test Excel report produced by ExcelWrapper	47
Figure 9	Probability density function plot of a parameter suffering from systematic errors	49
Figure 10	NU-AGE variables documentation file	50
Figure 11	NU-AGE main page in LabKey platform	52
Figure 12	NU-AGE menu in LabKey platform	53
Figure 13	NU-AGE docs in LabKey platform	54
Figure 14	NU-AGE study navigator in LabKey platform	55
Figure 15	NU-AGE Systems Biology - The ideal enrolled subjects' situation at T0	56
Figure 16	NU-AGE Systems Biology - The ideal enrolled subjects' situation at T1	56
Figure 17	PCA on Nutrients Data - Fraction of explained variance by each PCA component	58
Figure 18	PCA on Nutrients Data - First component.	59
Figure 19	PCA on Nutrients Data - Second component.	60
Figure 20	PCA on Nutrients Data - Third component.	61
Figure 21	PCA on Nutrients Data - Fourth component.	62
Figure 22	The NU-AGE Cloud.	63
Figure 23	Epistasis analyzed with an entropy-based approach.	67
Figure 24	Trend of 90+ people percentage in the whole population	68
Figure 25	Bayes Factor versus Log Odds Ratio - non syn, aa level	70
Figure 26	Bayes Factor versus Log Risk Ratio - non syn, aa level	71
Figure 27	Bayes Factor versus Log Odds Ratio - all, nt level	71
Figure 28	Bayes Factor versus Log Risk Ratio - all, nt level	72
Figure 29	Manhattan plot based on Bayes Factor values - all, nt level	72

LIST OF TABLES

Table 1	NU-AGE questionnaires and assays schema	12
Table 2	GEHA sequence divided by Country and Status	13
Table 3	NU-AGE questionnaires and assays schema	26
Table 4	NU-AGE Centralized assays	27
Table 5	NU-AGE Validator Rules	40
Table 6	Threshold values for K interpretation	69
Table 7	Epistasis - Possible interaction models	74
Table 8	Results of epistasis analysis on mtDNA AA couples	76
Table 9	Comparison of variants detected by Sanger sequencing, Ion Torrent variant caller and GATK (against rCRS)	79

Table 10 Comparison of variants detected by Ion Torrent variant caller and GATK (against RSRS) 79

INDEX

1.1	Systems biology of aging	1
1.2	The NU-AGE Project	2
1.2.1	NU-AGE technical infrastructure	3
1.3	Genetics, a focus on	5
1.3.1	Humans as metaorganisms	5
1.3.2	Mitochondria & Mitochondrial genome	5
1.3.3	Mitochondrial genome	6
1.3.4	mtDNA and aging	7
1.3.5	The GEHA study	7
1.3.6	The interacting mitochondrion	7

Data management is the *art* of collecting, curating and properly storing data. It usually represents the introductory step to data analysis, which is the *art* of extracting meaningful results out of the high quality datasets obtained in the previous phase. Notwithstanding these two disciplines are commonly perceived as two faces of the same coin, they are in fact devoted to two consecutive but sharply separated tasks.

The work hereby described encompasses both these arts, applied in two different context: data management within the FP7 EU project NU-AGE (SANTORO et al. 2014) and data analysis on the mtDNA data produced by the GEHA project (FRANCESCHI et al. 2007a). NU-AGE aim is to assess if a fortified diet, tailored on elderly's needs, may be involved in modeling the aging process. GEHA, on the contrary, focused on understanding the genetic determinants of longevity.

Both studies require an holistic approach to the data to reach meaningful results: longevity is well known to be a complex phenotype, where many, possibly independent, factors play a role in determining the final outcome. *Systems biology* represents the corner stone around which all analysis should be planned and developed.

1.1 SYSTEMS BIOLOGY OF AGING

Aging is characterized by a progressive decrease in the fitness of interconnected physiological systems at all the levels of physiological organization, from single molecules up to the entire organism. A strong relationship has been highlighted between chronic, low level inflammation and aging, a condition named "inflammaging" (CEVENINI et al. 2013; FRANCESCHI et al. 2000, 2007b). Inflammatory mediators involved in this condition was proposed as the "common soil" where age-related pathologies may develop and flourish (BLAGOSKLONNY et al. 2009; SALVIOLI et al. 2013). If from one side inflammaging can be considered a physiological phenomenon, it is also true that it can rapidly become detrimental if the process goes out of control (DE MARTINIS et al. 2006), causing an excess of local and systemic inflammatory response. Tackling inflammaging can thus be considered a priority, representing an effective strategy to reduce the inflammatory status, a striking risk factor for elderly people, slowing down the development of all those harmful processes that have in the inflammatory response their main pathogenesis. Building mechanistic mathematical models describing how biological systems interact and eventually allow for the establishment of inflam-

maging, in response to internal and external stimuli altogether, will help in further dissect the mechanism behavior and possibly help to define effective interventions to counteract the chronic inflammation (CEVENINI et al. 2008; RATTAN 2014).

The modeling of the NF- κ B pathway represents an outstanding example of this working hypothesis, where not only models help in fully understand pathways but also suggest where succeeding experiments should be aimed at (BASAK et al. 2012; TIERI et al. 2012). In fact, while the original scope of the work by Basak et al. was to comprehend the roles different I κ B isoforms play in the temporal response to inflammatory stimuli, it also propelled two other "side" achievements: first, a careful revision of the literature available on the topic, done to implement the model itself, and second, thanks to computational sensitivity analyses, the discovery that different degradation rates of free and bound pools of I κ B are part of a cross-regulation mechanisms contributing to the robustness of the system response (O'DEA et al. 2007). Finally, it was also suggested that NF- κ B response may also be influenced by intracellular available metabolic information, an hypothesis which is now under study (TORNATORE et al. 2012).

Besides inflammaging, aging also causes the loss of the so-called *metabolic flexibility* (DIPIETRO 2010), the capacity of cells and tissues to adapt themselves from lipid oxidation in fasting conditions to the increase of glucose uptake, oxidation and storage after a meal or in insulin-stimulated conditions (GALGANI et al. 2008). Micronutrients imbalance, as well as overnutrition, may affect metabolic flexibility in several ways: glucose excess causes the production of ROS that in turn increases oxidative stress (NUNN et al. 2009; PONUGOTI et al. 2012); Moreover, overnutrition impacts in the capacity adipose tissue has to cope with lipids in excess, eventually causing lipids accumulation and lipotoxicity, and may cause cells to fail shifting towards glucose oxidation due to impaired translocation of GLUT4. Metabolic flexibility is also involved in system-wide metabolism dysregulation, for example in type-2 diabetes (T2d) (CORPELEIJN et al. 2009).

Immunity and metabolism are deeply intertwined partners. Overnutrition, disrupting energy storage and utilization, propels local inflammation and impairs metabolic function (HOTAMISLIGIL et al. 2008), whereas macronutrients deficiencies are suggested to hamper the immune response, somewhat removing the fuel needed to sustain it in terms of energy and substrate needed for proper immune defense. A precise, quantitative, modeling of these complex interactions may thus foster the development of a diet providing the proper nutritional income to avoid igniting any of these pathological and pre-pathological states (CALÇADA et al. 2014; CEVENINI et al. 2010).

1.2 THE NU-AGE PROJECT

NU-AGE, "*New dietary strategies addressing the specific needs of elderly population for an healthy aging in Europe*", (BERENDSEN et al. 2014; SANTORO et al. 2014) is a FP7 European project that aims to fill the gap caused by the lack of an integrated approach in the field of nutrition as a mean to contrast, or at least slow down, inflammaging (BIAGI et al. 2012; CLAESSESON et al. 2012; FRANCESCHI 2007; JEFFERY et al. 2013; RIBARIČ 2012).

Eurostat (<http://epp.eurostat.ac.europa.eu>) predictions forecast that the proportion of 65+ people will severely increase in the EU27 (the 27 Member States of the European Union) from 17% to 30% by 2060. A similar increasing trend is also predicted for 80+ people, with a 3-fold increase from 21.8 million in 2008 to 61.5 million by 2060. This dramatic rise has direct and important implications in terms of the health costs countries will be called to sustain while the proportion of elderly people progressively increases in the population. Any strategy able to slow down, reduce or, even better,

postpone the onset of aging-related diseases, thus improving the health of elderly European citizens, would be warmly welcomed. A good measure of how EU considers this issue critical can be provided by the fact that the whole EU Horizon 2020 program for research and innovation funding is devoted to increasing of two healthy years the life expectancy of EU citizens, based on interventions on physical activity and diet.

NU-AGE builds on the outcomes of previous studies focused on analyzing single nutrients intakes (BOUWENS et al. 2009; GUO et al. 2009) and pushes on to embrace the concept of "*whole diet*" as a mean of targeting the metabolism in its entirety and investigate the sophisticated interconnections that may exist among different nutrients and their impact on metabolism and, ultimately, to health status.

The project enrolled in total 1250 sex-balanced free living elderly people, ranging from 65 to 79 years, free of major overt diseases, in 5 European countries (Italy, United Kingdom, Netherlands, Poland and France). A map showing the geographical location of each subject is visible in Figure 1 on the following page. The trial is a 1-year randomized, single blind, controlled trial with two parallel arms (control vs "diet"). Subjects, after initial characterization, were randomly assigned to each arm in equal number. Five standardized questionnaires were then administered both before and after the trial, encompassing the following domains: dietary habits, life-style and physical activity, health and medication and other relevant data such as the functional cognitive status. Blood, urine and faecal samples were collected before and after the trial to allow for:

- the analysis of immune, inflammation, metabolic, genetic and epigenetic status;
- measures of the intake of sodium, nitrogen, creatinine and potassium;
- definition of microbiota composition.

Furthermore, NU-AGE will take advantage of "omics" analyses such as transcriptomics, metagenomics and HITChip arrays to investigate to the deepest possible level the cellular and molecular mechanisms underpinning the dietary effect and to identify specific nutritional targets in the elderly.

1.2.1 NU-AGE technical infrastructure

NU-AGE will collect (and in part already has) a massive amount of data that requires particular care to be efficiently exploited. My role within this project was to help building the technical infrastructure to support the collection process, from designing the data entry interface to validating the data and building the database and the needed documentation. Indeed, a nicely set data management plan represents the best achievement a research project can obtain to ensure long life to its data.

There are 4 tightly interconnected steps to be considered when talking about data management: data entry and transmission, data validation, data documentation and data storage. Failing to attain even one of them may pose the whole project at risk of failing its objectives in the long term. Each step, however, requires dedicated resources, i.e. databases and graphical interfaces, which all together can be, and *should be*, considered as a result of the project itself, given the amount of time that must be devoted to these tasks.

These challenges become more difficult to tackle as a function of the number of involved recruiting centers. NU-AGE, with its 5 recruitment centers with different backgrounds, was thus in a critical position. Controlling this risk required a lot of preliminary work to harmonize how information was collected and the development of ad-hoc solutions to detect and address inconsistencies in the shortest possible time, to reduce their propagation within the data.



Figure 1: Map of the subjects enrolled by NU-AGE. Point width represents the number of subjects enrolled in each geographical location, data aggregated at municipality level. Map reconstructed with Google Maps API (<https://developers.google.com/maps/>) and Basemap (<http://matplotlib.org/basemap/>).

There is a last issue that, common in most of the data management steps, becomes predominant in the last one: *data security*. There is an innate dichotomy here: security in first place means that information must be stored in protected repositories: data are the final result of the millions of euros spent to obtain them. Second, but first from an ethical point of view, data must be secured to protect *subjects privacy*, which is the cornerstone around which clinical research is based upon. Missing to comply with this golden rule should represent enough evidence to sustain that involved parties weren't able to cope with the challenges they were facing, even before being considered a matter to be discussed by attorneys in a court. NU-AGE took data security seriously and promoted all the technical and non-technical precautions to prevent data disclosure, a briefly sum of which you will see in later chapters.

Unfortunately, it is still too soon to draw conclusions from NU-AGE data. At the time of writing, a significant part of the data that will constitute the future database is still in the data entry phase. In spite of this, you will be able to get from this dissertation at least a feeling of the challenges that analyzing the data produced by the project will pose, some of which we are already discussing and, whenever possible, solving.

1.3 GENETICS, A FOCUS ON

While NU-AGE main goal is to understand how the diet will impact on subjects as a whole, the project is also laying strong foundations for less "diet-centric" genetics analyses. All subjects will be genotyped with an Illumina OmniExpress chip containing 750K probes, eventually building a remarkable database describing the european genetic variability, on which scientific hypotheses may be tested.

1.3.1 Humans as metaorganisms

However, limiting the analysis to a panel of mainly nuclear SNPs such as those comprised in OmniExpress chips may not be enough. In recent years, researchers started to think longevity as the result of three separate genetics, reflecting three separate genomes: the nuclear genome (nDNA), the mitochondrial genome (mtDNA) and the gut microbiome (GM), the last composed by the whole set of bacteria living inside our gut with their own set of genes (GARAGNANI et al. 2014). From this point of view, human beings should be consider not as a single organism, but as a "*metaorganism*" in which three different microbial ecosystems act together in response of each other and of external stimuli. Moving from the same premises, the result of the aging process itself should be considered as the final outcome of a complex interaction of biological and nonbiological factors, the latter further subdivided in environmental and stochastic factors. In developing this thesis we focused our efforts in investigating one of three genomes: the mitochondrial DNA.

1.3.2 Mitochondria & Mitochondrial genome

In the cell, mitochondria oversee several key functions: they produce energy in form of ATP by mean of oxidative phosphorylation (OXPHOS), they are involved in most metabolic pathways and in the apoptotic process, of which they represent the first step through the release of cytochrome-c. Diseases such as Alzheimer, diabetes and most degenerative diseases, along with AIDS progression (HENDRICKSON et al. 2008) seem linked to this organelle. Finally, ROS production and mtDNA mutagenesis are the two solid links connecting mitochondria to the aging process (LAGOUGE et al. 2013), albeit

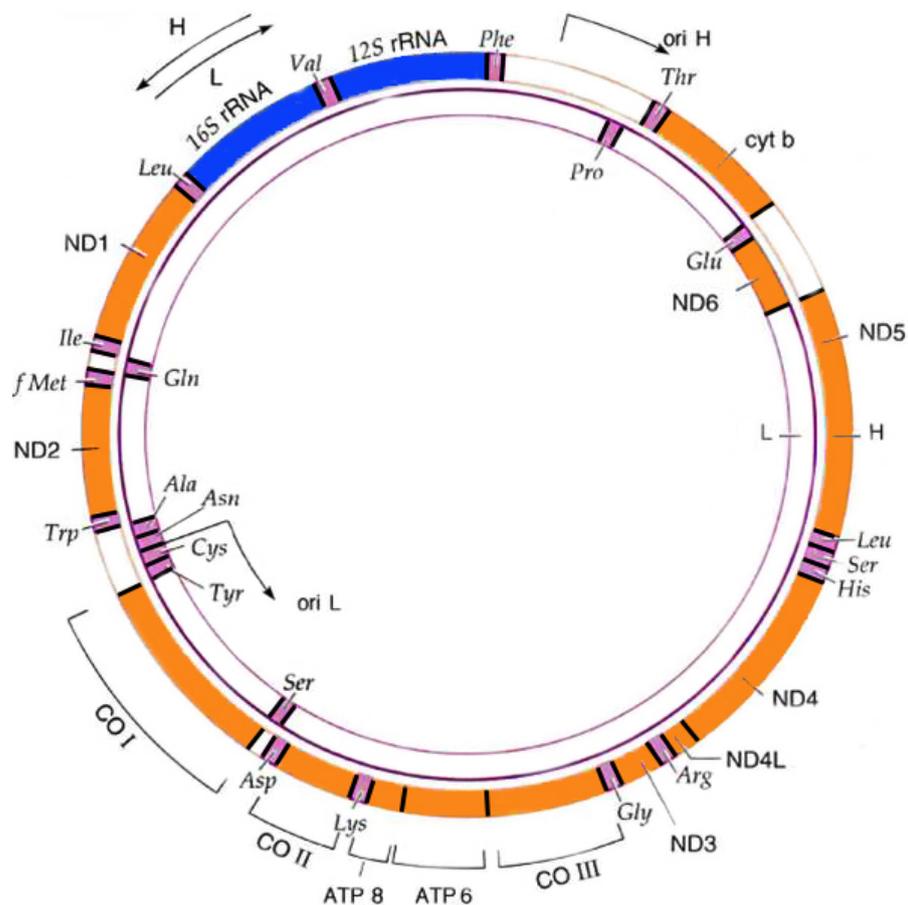


Figure 2: Mitochondrial DNA molecule.

recent studies suggested that increase of ROS production merely represents a consequence of the age-dependent accumulation of mtDNA damage, and not a cause of aging itself (KENNEDY et al. 2013; SHOKOLENKO et al. 2014; TRIFUNOVIC et al. 2004).

1.3.3 Mitochondrial genome

The mitochondrial genome is constituted by a single, circular, chromosome of 16569 base pairs, present in a variable number of copies in each mitochondrion. At the time of writing, there's a huge debate in the scientific community regarding the reference to be used in mtDNA studies (BANDELT et al. 2014), counterposing advocates of rCRS (ANDERSON et al. 1981; ANDREWS et al. 1999), the formerly undisputed reference, to researchers supporting the adoption of RSRS, a chimera sequence ideally placed at the very top of the mtDNA phylogenetic tree, as the new standard (BEHAR et al. 2012). In this thesis, whenever possible, we choose RSRS as our reference sequence, since we believe it represents a more "neutral" reference point, being it an ancestral rather than a "phylogenetically peripheral" and modern mitogenome from Europe, a definition that instead perfectly fits rCRS.

Out of the ~1500 genes originally coded by the mtDNA, only 37 are still present: 12S /16S rRNAs, 22 tRNAs required in mitochondrial protein synthesis and 13 polypeptides belonging to the OXPHOS system (WALLACE 2005). The remaining genes progressively moved to the nucleus, probably to reduce the risk of their degeneration caused by the Muller's ratchet (WALLACE 2007). However, this continuous flow of copies of mitochondrial genes, some of which are not under selective pressure in the nucleus, is the root cause of the most debated problem in sequencing and assembling mitochondrial genomes: *numts* (Nuclear Mitochondrial DNA sequences). Numts are sequences in the nucleus that show a very high similarity to mitochondrial

sequences and represent a very good explanation for all the misidentifications suffered by several studies focused on mitochondrial diseases and phylogeny reconstruction, and should be thus treated with particular care (CANNINO et al. 2007; HAZKANI-COVO et al. 2010; M. LI et al. 2012; PICARDI et al. 2012).

1.3.4 mtDNA and aging

The progressive accumulation of mutations in the mtDNA causes an impairment of the energy production, which is in turn contrasted by the cell increasing the number of mitochondria. At the same time, studies suggest that mutated molecules do have a replicative advantage on wildtypes (REARDON 2002). This phenomena, in combination with the progressive turnover of older mitochondria, provokes an enrichment of mutated forms due to the genetic drift. Sooner or later, the fraction of damaged mtDNA will reach a breaking point where the decrease in energy production causes an increase in ROS production that drives the cell toward the activation of the apoptotic process. At the tissue level, this prolonged cell loss will eventually disrupt functionalities, igniting pathological states (WALLACE 2005). Neurodegenerative diseases, such as Alzheimer, Parkinson and Huntington syndrome (SANTORO et al. 2010; YANG et al. 2008), as well as sarcopenia (KUJOTH et al. 2006), AIDS progression (HENDRICKSON et al. 2008) and aging (SALVIOLI et al. 2008) were all linked to mitochondrial inefficiencies caused by mtDNA damage. Recent association studies focused on age-related diseases identified nuclear loci involved in mitochondrial bioenergetics (PPAR γ , PGC-1 α and UCPs), strengthening the evidences of a role of mitochondria in these pathological states (CAPRI et al. 2013; J. J. JIA et al. 2010; RAULE et al. 2014; WALLACE 2005, 2013).

1.3.5 The GEHA study

In recent years, results coming from the GEHA study (FRANCESCHI et al. 2007a; SKYTTHE et al. 2011) emphasized the role recurrent and sporadic mtDNA mutations may have in longevity. The project sequenced the control region of 3000 samples and the complete mtDNA of 1292 samples (646 ultranonenarians, 646 controls) from subject enrolled in Denmark, Finland, Southern Italy and Greece (RAULE et al. 2014). An initial haplogroup-based analysis detected an association between haplogroups H2 and T2 with longevity in females, while J2 resulted associated only with males, contradicting previous results (DOMÍNGUEZ-GARRIDO et al. 2009; NIEMI et al. 2003). However, none of these associations reached the significance level after proper multiple testing correction was applied. To investigate if rare variants participate in determining longevity, a Rare-Variant association testing for sequencing data (WU et al. 2011) was performed. Obtained results proved that the number of non synonymous mutations in mtDNA genes coding for subunits composing OXPHOS complexes I, III and V significantly differs between 90+ and controls. Going into details, mutations in complex I may be beneficial for longevity, while the cooccurrence of mutations on both complex I and III or on both I and V might be detrimental for longevity. The really interesting finding of GEHA, until now, is this very last results: mutations do interact to determine the phenotype (long-lived or not).

1.3.6 The interacting mitochondrion

As stated above, more than the 90% of the components needed by mitochondria to work are nuclear-encoded. Thus, there's a tight interplay between the two genomes, nDNA and mtDNA, to preserve biological function and

cell activities (BAR-YAACOV et al. 2012; KENYON et al. 1997; RAND et al. 2004). This "communication" is bidirectional, with information travelling from the mitochondrion to the nucleus, and from the nucleus back to the mitochondrion. The "channels" through which this communication can take place are multiple: protein-protein interaction, protein-RNA interaction in the mitochondrial ribosome and nuclear factors-mtDNA recognition sites interactions in the transcription and replication processes (BAR-YAACOV et al. 2012; LUNG et al. 2006; RACKHAM et al. 2011). It is well established in literature that keeping these interactions stable and balanced is fundamental for the cellular homeostasis and even small alterations lead to stress and accelerated aging. However, little is known on how this communication channels are regulated.

The investigation of epistatic effects such as the ones described above represented a nearly-impossible challenge (TRANAH 2011) until the advent of NGS techniques and the increase of available computing power. NGS provoked a steep fall in the cost of obtaining a sequence, both in terms of money and working time, at the same time skyrocketing the volume of data to leverage in the analyses. However, as the amount of information to process increases, it is also mandatory to develop hardware and methods able to cope with this tremendous amount of data to be processed in reasonable time. *High performance computing* (HPC) is now able to reach a computing power that was not even imaginable some years ago. In particular, GPU (*Graphics Processing Unit*) computing is progressively revolutionizing the field, delivering groundbreaking performances when small but massively parallel calculations are needed. Indeed, this is the kind of computations required to perform epistasis analysis (GOUDEY et al. 2013; PÜTZ et al. 2013). An important drawback GPU computing has is the complexity of programming algorithms on these platforms, which requires dedicated – and complex – programming languages that are difficult to master without a pure computer science background. Additionally, dedicated and somewhat expensive hardware is needed, since using off-the-shelf cards may deliver performances way below those achieved with consumer-targeted CPUs.

It's from these foundations that we moved to build our approach to investigate epistasis. As you may have understood by now, GEHA possess all the data needed to perform this kind of investigation, which may provide useful insights on the mtDNA role in the longevity mechanism. Methods to analyze these interactions already exist (CORDELL 2002; EMILY 2012; MOORE 2015; MOORE et al. 2014; RITCHIE 2015), but we preferred to develop a new method based on Bayesian statistics that, in our opinion, was better suited to detect these subtle relations, helped in the job by carefully crafted *priors*, obtained by analyzing available sequences databases. Still, conducting these analyses is really complex: even analyzing a handful of mutations requires to evaluate all the possible interactions among them. Limiting the analysis to couples, the problem increases quadratically, n^2 , where n represents the number of mutations under analysis. For this reason, we initially tested our method to detect single variants associations in GEHA mtDNA sequences.

We then analyzed all the non synonymous variants at AA level, removing those positions where more than two different AA were found. This expedient allowed us to avoid having the computational problem exploding in our hands, while retaining a good exploration of the possible epistatic interactions. While results of this pilot study will be explained in details in the following chapters, we would like to stress since now that this method, once further optimized in terms of computing time, will be ready to be used to investigate epistasis in the full set of mtDNA variability discovered in GEHA, as well as to be exploited to explore the very same phenomena including nuclear variants, finally shedding a bit of light on the nuclear-mtDNA crosstalk.

2

AIM OF THE STUDY

Aging and longevity, as well as inflammaging, are complex phenomena requiring high quality data and cutting-edge analyses to be even partially understood. At the same time, there is a growing necessity to tackle these problems, given by the fact that they do have a severe impact on people's health and, to be blunt, on the costs each country must sustain to maintain its own population healthy. The aim of this study is to build on previous knowledge to at least start unraveling the complex net tightly interconnecting these three phenomena, providing both the components needed to solve the equation: data and analysis methods.

Data will be provided in the context of the NU-AGE study, which aims to understand the role diet has in regulating the inflammatory status. In particular, we will design and implement the project data management pipeline. To be successful, this effort requires to combine several independent but tightly interconnected parts covering the whole life of the data, spanning from the very first moment of their creation, the data entry process, to their long term storage in properly configured and secured databases. Fulfilling these requirements, however, is far from being easy: as we will see, it requires to mix together completely different backgrounds, ranging from software programming, to medicine and data documentation, down to web applications deployment and monitoring. We will also discuss some of the preliminary data the project has produced, with a particular focus on the nutrient intakes.

NU-AGE will collect a plethora of data encompassing several different fields, that will have to be combined together to provide a reliable and definitive answer to the scientific questions the project is based upon. As for all intervention studies, a critical part is represented by the evaluation of the subjects compliance. In our case, this compliance is extremely difficult to measure: we're not administering a new drug, we're administering a diet that each subject may decline in his or her own way, eventually diverging from the expected behavior. Moreover, controls may change their behavior while enrolled in project in response to external stimuli, adopting a healthier diet on their own. It is thus fundamental to precisely measure how each subject was compliant to its branch, to weight its data accordingly. To address this key need, in this study we will propose a way of obtaining a reliable index to measure compliance in diet-based intervention studies.

New analysis methods, on the contrary, will be provided in the form of new approaches to detect and measure the epistatic interaction in mitochondrial DNA, an unanimously recognized player in the longevity arena. These methods, based on entropy analysis and Bayesian statistics, will be first tested on the GEHA mtDNA sequences to verify if they will be able to detect the association of single variants to the longevity phenotype, and then used on the very same sequences to understand the effects that interacting mutations may have in influencing the aging process, if any. Additionally, we will begin to pave the way to the analysis of the crosstalk that occurs between the mitochondrial and nuclear DNA, which was already hypothesized in literature but is yet to be proven. Eventually, we will also see how the obtained results, if meaningful, may be used to provide a direct feedback on the probability of a subject of becoming long-lived or not.

As a side topic, we will cover the realization of a pipeline to analyze NGS data coming from a Ion Torrent PGMTM platform, stimulated by the poor performances achieved by stock softwares provided by the manufacturer in calling variants harbored by the mitochondrial DNA, which is well known

to represent one of the most tough molecule to be sequenced and analyzed due to its innate peculiarities, such as the widespread presence of poly-n stretches.

3

MATERIALS AND METHODS

INDEX

3.1	Data	11
3.1.1	NU-AGE Data	11
3.1.2	GEHA Data	12
3.1.3	DbSNP Data	13
3.1.4	Eurostat Data	13
3.2	Programming Languages and Libraries	13
3.2.1	Python	14
3.2.2	Python Libraries and Packages	14
3.2.3	IPython Notebook	17
3.2.4	R	19
3.3	Databases and Web Interfaces	19
3.3.1	Text files	19
3.3.2	SQLite	20
3.3.3	PostgreSQL	20
3.3.4	Dbnp	20
3.3.5	LabKey	20
3.4	Programs and algorithms	21
3.4.1	Phred, Phrap and Consed	21
3.4.2	ClustalW2 and Clustal Omega	22
3.4.3	FastQC	22
3.4.4	Swiss-army knives for genomics data management	23
3.4.5	GATK	23
3.4.6	HaploFind	24
3.4.7	EpiData Entry	24

Different data, languages, systems and programs were used in this PhD thesis, depending on the specific task to be accomplished.

Different programming languages may be differently suited to deal with certain type of data, and the same holds for the numerous *Relational Database Management System* (RDBMS) now in the wild. Usually, the final choice is driven by the attempt to reduce the overhead on the analyses due to the chosen infrastructure, being it a language or a database.

The aim of this chapters is to list and detail all the technologies and information used throughout this study.

3.1 DATA

As its title says, this thesis is pretty much a deep voyage in data management and analysis. In is thus important to note that the work described in the following chapters draws fully from years of efforts taken by several people in mainly two different European-funded project: GEHA (FRANCESCHI et al. 2007a) and NU-AGE (BERENDSEN et al. 2014; SANTORO et al. 2014).

3.1.1 NU-AGE Data

NU-AGE (BERENDSEN et al. 2014; SANTORO et al. 2014) is a FP7 European project with the main task of understanding if the Mediterranean diet can influence, and to which extent, elderly's health. The project recruited 1250 subjects in the age range 65-79, half of which followed a diet specifically tailored on the elderly's needs for 1 year.

Table 1: NU-AGE questionnaires and assays schema

Measurement	Time 0	Follow-up 4m	Follow-up 8	Time 1
Blood	x			x
Urine	x			x
Faeces	x			x
Anthropometric	x			x
DEXA-scan	x			x
Physical Activity	x			x
Physical Performance	x			x
Cognitive status	x			x
General questionnaire	x			x
Follow-up questionnaire		x	x	x
Dietary intake ^a	x	x	x	x

^a Dietary intake at month 4 and 8 is assessed only in participants in the diet group.

Understanding the effect that diet may have on the people's health is a topic that clearly falls in the list of problems you must address with a System Biology approach: hundreds of parameters may play a small but significant role in this phenotype, and not considering even a small fraction of them may be detrimental for the study outcome. Moving from this premises, NU-AGE adopted the most comprehensive panel of questionnaires and assays that was technically feasible, a schema of which can be seen in Table 1.

As the reader may imagine, the amount of information produced with this setup is overwhelming: we're talking about roughly 2000 variables, and that is just a small part of the whole database, since storing results of the planned "omics" analyses, such as

- genomics
- lipidomics
- metabolomics
- transcriptomics
- microbiome analysis via deep sequencing

will be the really groundbreaking challenge here.

Details on how we solved the data management problems will follow in the succeeding chapters.

3.1.2 GEHA Data

GEHA (FRANCESCHI et al. 2007a) aim was to identify candidate genes involved in healthy aging and longevity. The consortium, composed by 24 European partners and the Beijing Genomics Institute from China, recruited in 5 years 2650 long-lived 90+ sibpairs and 2650 younger, ethnically matched, controls from 11 European countries, of which DNA and healthy status was obtained.

While further details about the study setup can be found in the cited article, for the sake of this thesis it is important to note that UNIBO was responsible for the mtDNA sequencing of the enrolled subjects, which was only in part carried out in our lab, with the biggest batch of samples sent to the BGI for sequencing. In total, GEHA obtained the complete mitochondrial DNA sequence for 1292 subjects, subdivided as reported in Table 2 on the facing page, and ~2500 D-loop sequences equally subdivided in sibs and controls.

Additionally, a subset of the samples were genotyped with an Illumina Human OmniExpress chip containing ~730K probes. This batch originally

Table 2: GEHA sequence divided by Country and Status

	Denmark	Finland	Calabria	Greece	By Status
Sibs	422	149	66	8	645
Controls	429	146	58	14	647
By Country	851	295	124	22	1292

comprised 370 subjects from Bologna, of which we now have information about their nuclear variants.

Finally, in 2014 the phenotypical database (SKYTTE et al. 2011) was transferred to a secure server hosted in the UNIBO IT infrastructure. The database contains the questionnaire information gathered from sibs and controls, with ~200 and ~100 variables respectively.

3.1.3 DbSNP Data

Microarray results are as good as their annotation is sound and updated. GEHA microarray data dated back to 2011, and the mapping of the variants could have been changed significantly in this time interval. Thus, before proceeding with further analyses it was mandatory to update the original annotations. However, getting fresh data for 700K+ variants is not something you can do by hand. DbSNP (SHERRY et al. 2001) offers a "Batch submission" system that accepts up to 30.000 rs codes per request and sends back the updated records after a time that varies between a couple of minutes to 24 hours. In our case, using this approach would have meant to chop our list of variants in 25 batches, submit them separately and then glue the results back together. Luckily, Ensembl released in 2014 a brand new REST API (YATES et al. 2014) that allows to query their variants databases (that are mainly a mirror of DbSNP content) for updated information. We thus built a small script based on Python Request library (see 3.2.2) to automate this task, and retrieved the needed information in a couple of hours.

3.1.4 Eurostat Data

Dealing with data flowing in from different European countries poses several difficult to answer questions when you merge the information together: country is, obviously, a confounding factor you must consider when analyzing the data. But how strong this factor is for longevity? To answer to this question we used Eurostat public data to estimate the proportion of 90+ subjects in the population of countries participating in the GEHA project.

3.2 PROGRAMMING LANGUAGES AND LIBRARIES

Programming languages are the language you can talk to be understood by a computer. There are many of them, developed in different times and to address different problems, but they are usually divided in two main categories: *compiled* and *interpreted*. The first approach is well known for its speed, which is usually order of magnitude faster compared to the second, while the latter allows for greater flexibility and platform independence (i.e. Java).

Compiled languages are much faster because the source code is first "compiled" in *machine language* and then interpreted by dedicated hardware or software at time of execution. This distinction does not exist in interpreted languages, in which both processes are done on the fly. However, this speed gap can be partially filled by *just-in-time* compilation: the code is compiled

by the interpreted just before execution, with a gain in speed that can eventually be considerable, depending on the program itself (KLUGER 2008).

3.2.1 Python

The great part of the work described in this thesis was done in Python (Python Website), a language implemented by Guido van Rossum. Python is an *interpreted*, object-oriented language, which is thus interpreted on-the-fly by the Python Interpreter, usually referred to as CPython. A project, codenamed PyPy (pypy Website), is now developing a new implementation of the Python language that contains a Just-In-Time compiler, allowing programs to run faster and with a lower memory footprint.

Python gained a considerable momentum in many fields, primarily because of its easiness of use and low learning curve. It has an innate predilection for strings (thus sequences) processing, making it a top choice adopted by bioinformaticians all around the world, but also allows to solve most tasks exploiting dedicated libraries: numerical calculus (VAN DER WALT et al. 2011), data exploration and analysis (MCKINNEY 2010; PÉREZ et al. 2007), advanced plotting (HUNTER 2007), web development (Django Website) and biological data management (COCK et al. 2009).

3.2.2 Python Libraries and Packages

For most of the question you pose to it, Python has a library to answer with. That's particularly true in the biology/bioinformatics world, thanks to the growing adoption Python has in these fields. Hereby follows a list of all the libraries used in this thesis, with a short description for each of them.

BioPython

BioPython (COCK et al. 2009) does all the heavy lifting of reading (a.k.a. parsing) biological formats such as FASTA, ClustalW or GenBank (BENSON et al. 2013) and interacting with NCBI's Entrez databases to download various type of information, i.e. article abstracts, sequences or GEO datasets (BARRETT et al. 2013).

In the context of this thesis, BioPython was used to pre-process GEHA mtDNA sequences, both at the quality-check step and before starting epistasis analyses.

SciPy

SciPy (SciPy Website) can be seen in two different ways:

- a package grouping together mathematical algorithms and convenience functions based on NumPy (VAN DER WALT et al. 2011);
- an ecosystem of open-source Python packages devoted to science, mathematics and engineering.

Most of the packages that follow in this list are indeed part of the SciPy ecosystem, which is every bit as good as the acclaimed R ecosystem.

Pandas

Pandas (MCKINNEY 2010) is a python package that provides two fundamental parts of the data processing equation:

- proper data structures, to load, filter and slice the data. The most notable structure is the *DataFrame*, that closely resembles a R dataframe;
- highly efficient but basic tools for exploratory analysis, i.e. average, percentiles, grouping and so on.

These two parts, merged together, allow for a quick exploration of new datasets, and provide a solid base for more advanced analyses. Pandas is build upon numpy (VAN DER WALT et al. 2011), a Python library for scientific computing.

Matplotlib

Matplotlib (HUNTER 2007) , a 2D plotting library, aims to, as stated in the project website

“make easy things easy and hard things possible”

and usually fulfill this duty in a very elegant way. The library literally puts the most common plots (i.e. histograms) at the programmer fingertips, and more complex plots, such as hexbins, just a couple of lines of codes away.

All the scientific plotting contained in this thesis was powered by matplotlib, helped by libraries such as seaborn (WASKOM et al. 2014) or basemap (Basemap Website), to improve plots appearance and for map projections, respectively.

Scrapy

If you need to extract – or *scrape*, to use the correct computer science term, information from a website automatically, Scrapy (scrapy Website) is the way to go. Python-based, Scrapy is an application framework to crawl web sites and extract structured data to be later processed for other task, i.e. for data mining.

The framework allows to build specialized crawlers that, starting from a set of url base addresses, dive deep in each website, following the links you instruct them to, selecting only the pieces of information you need and exporting them in an easy to use comma or tab delimited format. It also keeps track of already visited urls to avoid being stuck in a loophole and can automatically adjust the crawling speed to accommodate both the needs of the website you’re crawling and the computer you’re running the crawlers on. If more sophisticate processing and storage is needed, Scrapy can perform post-processing of the information, in form of pipelines, and directly store data into databases.

For the scope of this thesis, Scrapy was used to retrieve the *Anatomical Therapeutic Chemical* (ATC) nomenclature to validate the ATC codes entered by NU-AGE recruitment teams in the data entry process.

Requests

Request is, following the author’s claim,

“HTTP for Humans”

In other words, it’s a Python library to perform HTTP requests, the same type of request any browser, i.e. Firefox or Chrome, does every time you access a web page.

What’s the reason behind it in science, you may ask. We’re not visiting web pages, after all! The reason is that Ensembl (CUNNINGHAM et al. 2014) recently released its own REST APIs (YATES et al. 2014) (where API means *Application Programming Interface* and REST *Representational State Transfer*), eventually overcoming the long standing problem of the lack of supported APIs for any programming language, with the exception of Perl. REST APIs are instead language-agnostic, and can be interrogated via any programming language via HTTP request.

To do a "real life" parallel, you can think about REST APIs as a librarian you ask for a book (a dbSNP rs code, SHERRY et al. 2001). If your request is semantically correct (you're asking the right thing at the right address), he will reply with the book (data encoded in some format, i.e. JSON), or gently ask you to rephrase your question (something was wrong in your request). If you ask too many books together, he will gently tell you to come back later (too many requests together will be blocked).

A more bioinformatics-related example may be the following: you need to get updated information about a SNP of which you have only the rs code. You have many ways to do this, but the two main choices are:

- going to the dbSNP website, write the code in the search box, wait for the result, read throughout the record to find the information you need, and use it;
- ask to Ensembl the data via the REST APIs, have a program digest the reply and extract only the information you need, and then use it.

If you need annotations for just a couple of rs codes, both processes are doable. On the contrary, if you need to update thousands of them there is really no choice: a programmatic way is the only approach you can follow.

To implement such a script you can use Python Request library to connect to the Ensembl Variation Endpoint, at the URL address "http://rest.ensembl.org/vep/human/id", and perform a POST request. POST request are request that asks the server to accept the payload in the message body, in this case a JSON-encoded list of rs codes, to be processed. The server will thus process the list of codes and send you back the requested information, encoded in JSON. You can then decode the data and extract the bits you need. All these steps can be successfully done in 2 lines of Python code:

```

1 request = requests.post(endpoint, headers={"Content-Type": "
    ↪ application/json"}, data=json.dumps({'ids' : rs_list
    ↪ }))
2 variations = request.json()

```

where line 1 asks for the data (rs codes are contained in the `rs_list` variable) and line 2 parses back the JSON contained in the reply to native Python data structures.

In this thesis Request was used to obtain gene annotations of a set of variants contained in a OmniExpress Illumina chip containing ~730K probes.

Plumbum

Python has many built-in and external libraries to interact with external commands and redirect their output, but what makes Plumbum (Plumbum Website) outshine its competitors is the easiness of use.

Need to launch a complex-and-very-long command? Just shove all the arguments in the right order inside a list, and you're ready to go. You need to launch the same command many times with different arguments? Save the root command separately and combine it on-the-fly with the arguments you need each time.

Finally, in a very long pipeline, such as the pipeline we built for NGS variants calling, it is fundamental to have an exact control on what's going on, and this comes down to two things: logging, to have a clear trace of what happened, and exit code management, to halt the pipeline as soon as an error occurs, without losing time and resources processing dirty data. Well, Plumbum make this straight easy.

SQLAlchemy - The Database Toolkit for Python

SQLAlchemy (SQLAlchemy Website) is a Python library to connect to most DBMS, making easy to load and query data independently from the underlying infrastructure. It consists of two main components: the Core and the

Object Relation Mapper. The Core is a fully featured abstraction toolkit, which swiftly removes the problem to connect to different database implementations and allows to map and query Python Objects to database types. At the same time the *Object Relation Mapper*, or ORM, allows to associate user-defined Python Classes to database tables and instances of these classes to rows in the corresponding tables, taking care of continuously syncing them. In this way, the programmer can store and retrieve data from the DMBS without performing additional conversion steps or leaving the comfortable Python language.

To give you an explicit example of how the ORM allows for such a level of customization, here is how we defined an object to store information about each recruiting center:

```

1 class RecCentre(Base):
2     """
3     Table to store info on Recruiting Centres
4
5     """
6     __tablename__ = 'recruiting_centres'
7
8     id = Column(Integer, primary_key=True, autoincrement=
9         ↪ True)
10    cid = Column(Integer, nullable=False) # Center ID
11    name = Column(String, nullable=False)
12    address = Column(String, nullable=False)
13    country = Column(String, nullable=False)

```

This object is transformed in a SQLite (see 3.3.2 on page 20) table and instances automatically pushed to the database.

SQLAlchemy was adopted in this thesis to interact with the SQLite database at the basis of our validation pipeline.

3.2.3 IPython Notebook

IPython Notebook (PÉREZ et al. 2007) can be easily seen as your data best friend: it allows to combine code execution, documentation, text and image visualization in a single, easily shareable, page. Notebooks files are text files you can move around on your computer, send to colleagues to double-check your analyses, or make freely available on the web, with just a couple of clicks.

The IPython Notebook Folder of NU-AGE, grouping together all the notebooks used in this thesis, can be seen in Figure 3 on the following page. Notebooks are denoted by the ".ipynb" extension, while other rows refer to nested folders, containing both raw data or other notebooks. On the top, you can see buttons to access the running notebooks, or enable IPython cluster computing. You can also easily create a new notebook with the dedicated button.

An example of a running IPython Notebook can be seen in Figure 4 on the next page. Here you can appreciate IPython uttermost feature: an interactive experience nailing together code development and execution with image visualization and prompt access to libraries documentation. In the Figure you can appreciate at the top the code used to plot the violinplot shown just beneath it and also have a glimpse of how easy is to get help if, for example, you forget how to invoke a command: just write the command and append a "?" mark at the end, as shown in the last line, and IPython will fetch and display the help for you in no time!

All the plotting, and most of the data post processing, in this thesis was accomplished via IPython Notebooks, each time combining one or more of the previously described libraries.

IP[y]: Notebook



Figure 3: IPython Notebook Folder - Screenshot of a IPython Notebook folder. Files with extension .ipynb are IPython Notebook, while other entries are folders containing either Notebooks or data.

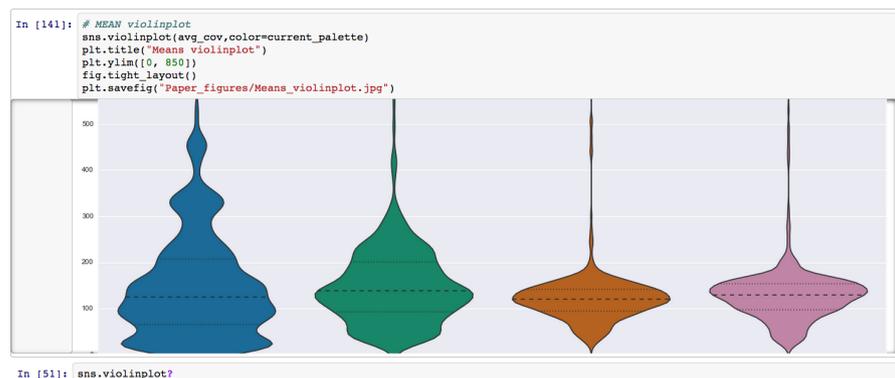


Figure 4: IPython Notebook screenshot. On the top we have the code used to produce the central figure. The last line shows how easy is to obtain commands documentation: append a "?" mark at the end, and IPython will get it for you, displaying it a dedicated box.

3.2.4 R

R (R FOUNDATION FOR STATISTICAL COMPUTING, VIENNA 2008) probably represents the most acclaimed open source programming language for data exploration and statistics. A great part of its reputation derives from the impressive number of packages available to "easily" analyze the most disparate data, developed and maintained by scientists, or companies, all around the world. Accessing these packages is usually straightforward: many, if not all, of them are stored in the *Comprehensive R Archive Network* (CRAN, CRAN Website) and can be downloaded and installed with a single command. On the contrary, in biology and bioinformatics the reference archive is Bioconductor (Bioconductor Website), which containing hundreds of packages devoted to the analysis of high throughput data.

R was initially adopted in this thesis for the pre-processing and analysis of raw data coming from the Ion Torrent platform. However, used to the way higher flexibility of Python, we abandoned R after some initial test and moved the analysis to the IPython platform.

3.3 DATABASES AND WEB INTERFACES

Properly and efficiently store data is the first step to easy processing and advanced (and useful!) analyses.

There is no catch-all answer to which system should be adopted: the final decision must carefully weights the kind of data you're storing and the overhead you can sustain to provide a higher performance storage backend. Sometimes, a complex *Database Management System* coupled with an advanced web interface represents the way to go, in other cases a text file is more than enough to cope with the problem.

3.3.1 Text files

The reader may be disappointed to find text files within a section entitled "Databases and Web Interfaces". When talking about databases, the imagination immediately skyrockets to extremely complex systems with terabytes of data ready to be queried with a single click. Sorry to say, this may hold strong in Google's world, but biology, with some notable exception (i.e. high throughput raw data, image processing, and so on), usually flies at a much lower altitude. We're not dealing with "big data" here.

That's the reason why a text file can be enough to store whole databases and share subsets of them. The VCF Format (VCF Format Specification website), the mainstream format to store variants identified via Next Generation Sequencing (NGS), is a *text file format*. You can compress it to save space, but it remains a (very well designed) text-based format. When several people need to access the same resource at the same time, or the curator frequently updates the data, that's the case in which something more advanced is needed, and databases kick in.

There is also an innate benefit in text files: they are just - as the name says - plain old easy text files: even in a very distant future, a reader for this format will exist, probably the grandson of the notepad installed by default in all your computers. So, a scientist needing your data in 10 years from now will have no problem in reading them. If the same data are locked in a proprietary abandoned file format, that information would be lost. Think about it! Always use the easiest and most standardized format you can afford for your data.

This quite long introduction to say that most of the data used in this thesis were, at some point of the chain, contained in text files. NU-AGE pipeline for data management includes steps in which data are sent in text files: it's indeed the main format we used to send data from recruiting centers and

labs to the central database. GEHA sequences were in FASTA format, that's again a text-based format, while genotypes were stored in PLINK (PURCELL et al. 2007) MAP and PED formats, and, imagine what? Text-based formats again!

3.3.2 SQLite

SQLite, citing the official website is

a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine

In a nutshell, SQLite provides you with a fully functional relational database within a *single file*. Advantages are striking: you can have your data stored in a database supporting many SQL queries without the overhead of running a DBMS on a dedicated server and losing time configuring countless configuration files. This comes with the price of no concurrent access - only a single person or process can access the file at once - and limited support to some SQL constructs. In conclusion, SQLite can be seen as a good compromise between text files and more complex DBMS.

For the scope of this thesis, SQLite was used to implement a small database to store information about NU-AGE to empower tracking and management of data failing the validation checks. You can easily convince yourself how this was the ideal situation to adopt SQLite: no need for concurrent access - a single process writes and reads the data - and for complex SQL constructs (i.e. certain kind of joins SQLite doesn't support).

3.3.3 PostgreSQL

PostgreSQL (PostgreSQL Website) (a.k.a. Postgres) is an Open Source DBMS with more than 15 years of active development. It supports most of the SQL:2011 standard, is ACID-compliant and allows for complex SQL queries with indexing methods not available in other implementations. It also has a particular focus on scalability, with latest version supporting database replication to reliably share the workload on multiple servers.

In this thesis, Postgres was used as database for LabKey (see 3.3.5), a data integration platform we deployed to address NU-AGE data management needs.

3.3.4 Dbnp

The "Nutritional Phenotype database" (Dbnp) (VAN OMMEN et al. 2010) is an open source effort to produce an application suite to store biological studies. Although, as the name suggests, it was designed to store nutrition-related data, it's module-based architecture allows for a very high degree of flexibility, allowing it to deal with other types of information such as phenotypical and omics data.

An instance of this application was (and is) used for long term storage of NU-AGE's data, hosted at the TNO, The Netherlands, a NU-AGE partner.

3.3.5 LabKey

LabKey (NELSON et al. 2011) represents one of the best examples of platforms for data sharing and scientists' collaboration. Implemented in Java, Labkey runs on multiple platforms (Windows, OS X and Linux), and requires:

- Apache Tomcat to run the web application;

- PostgreSQL (see 3.3.3 on the preceding page) as database backend;

A wiki page at the developers' website guides throughout the installation process that, unless particular configurations are needed (e.g. security requirements), doesn't require advanced computer science skills. Its developer team offers highly professional paid support, but the excellent user community forum gives a great way of troubleshoot common and uncommon problems for free.

After initial configuration, LabKey provides an easy and flexible system where to upload, store, analyze and plot biological data. Studies can be organized in nested folders, each one with specific modules enabled and customized permissions. Case/control studies are particularly well supported by LabKey, which provides specific functions to manage data produced within this type of projects.

Moreover, additional modules are available to improve the analysis and searchability of assays such as:

- Genotype data;
- Luminex data;
- Mass spectrometry data;
- Flow cytometry;

Finally, LabKey can also forward jobs to a Galaxy Server (BLANKENBERG et al. 2010; GIARDINE et al. 2005; GOECKS et al. 2010), transforming it in a highly efficient genomics analysis platform.

In this thesis, LabKey was deployed on a server provided by the University of Bologna's IT Department to aid NU-AGE data validation process and act as a temporary storage for the data that, once finalized, will be transferred to the dbnp (see 3.3.4 on the facing page).

3.4 PROGRAMS AND ALGORITHMS

Implement something from scratch when somebody else already did the job is useless (and somewhat silly), and this concept holds both for programming languages libraries and for whole softwares. The development of this thesis took advantage from many freely available algorithms to address some of the tasks, and hereby follows a list of them.

3.4.1 Phred, Phrap and Consed

Building a consensus sequence out of tens of chromatograms coming out of a Sanger sequencer can be a tough work, even when the target is a small molecule such as the mtDNA. Many different suites exist to reduce the burden of this work, implemented by sequencers manufacturers or independent developers. Our lab adopted for this task SeqscapeTM (Applied Biosystems Inc. part of Life Technologies, Carlsbad, CA), but it lacks support for some chromatograms formats used in the GEHA project. This forced us to move to a different pipeline, based on Phred, Phrap (EWING et al. 1998a,b) and Consed (GORDON et al. 1998).

The process is rather straightforward:

1. Phred loads and analyzes the raw files coming from the sequencer, calls the bases and computes a *quality value* for each of them, as a logarithmic function of the error probability.
2. Phrap reads Phred outputs and assembles the contigs using the mitochondrial DNA reference sequence (ANDERSON et al. 1981; ANDREWS et al. 1999) as a scaffold.

3. problematic sequences can then be further manually refined with *Consed*, a *graphical user interface* (GUI) to view Phrap output, if needed.

Phrap strong point is represented by the calculation of additional quality indexes that, supported by those produced by Phred, allows it to use a higher fraction of each read, as long as the quality is high enough to be aligned to the other reads. This helps increasing the throughput of the process, in turn lowering the need of human intervention: there's no more need for an experienced operator to recover erroneously discarded fragments to link otherwise separated contigs, an common issue with other softwares where reads are trimmed aggressively. If needed, *Consed* GUI can help in finalizing particularly tricky sequences.

In this thesis, this suite was used to reconstruct GEHA sequences starting from raw data produced by the Beijing Genomics Institute (BGI), China.

3.4.2 ClustalW2 and Clustal Omega

A direct way to evaluate the quality of a consensus sequence, apart of the quality values produced by the assemblers, it's comparing it to the reference sequence: the identification of many mismatches, far more than expected, is usually a very bright indication that something went wrong, possibly at the sequencing or assembling step.

However, to compare two sequences we must align them first. Aligning together many samples, usually against the reference sequence, in a single alignment would help the operator in assessing the quality of each of them in a single shot, dramatically reducing the time needed to check the whole batch. This kind of alignment is called *Multiple Sequence Alignment* (MSA), and, just to avoid overlapping, there are a very long list of softwares to do so, each one with its own advantages. After careful testing in terms of time and alignment accuracy, we eventually found that ClustalW2 (LARKIN et al. 2007), albeit a little slower than Muscle (EDGAR 2004), produced more consistent results. When ClustalW2 was officially no longer maintained, we switched to Clustal Omega (SIEVERS et al. 2011), an excellent implementation of the next generation of multiple sequence alignment algorithms.

ClustalW2 and Clustal Omega were used within this thesis to evaluate the quality of the GEHA mtDNA sequences produced in Bologna and at the BGI.

3.4.3 FastQC

The difficulty of evaluating the quality of sequences obtained via Sanger sequencing simply fades away when compared to the complexity of quality control in *Next Generation Sequencing* (NGS), when thousands – or hundreds thousands – small reads are at play. Luckily, with the progressive, in some way explosive, adoption of these technologies tools were designed to help researcher and technicians in evaluating the quality of this flood of data.

A very good example of this is FastQC (FastQC Website), which provides an integrated – and easy – way to perform basic quality control checks on the raw data coming out from the NGS sequencer. The user has just to load the file (usually a BAM file) and the program will compute common key statistics and display them in a friendly interface. An example of a (bad) run can be seen in Figure 5 on the next page, with quality values dropping low just few bases after the reads start.

FastQC was used to evaluate the quality of the data produced by an Ion Torrent platform while optimizing a protocol for complete mtDNA sequencing.

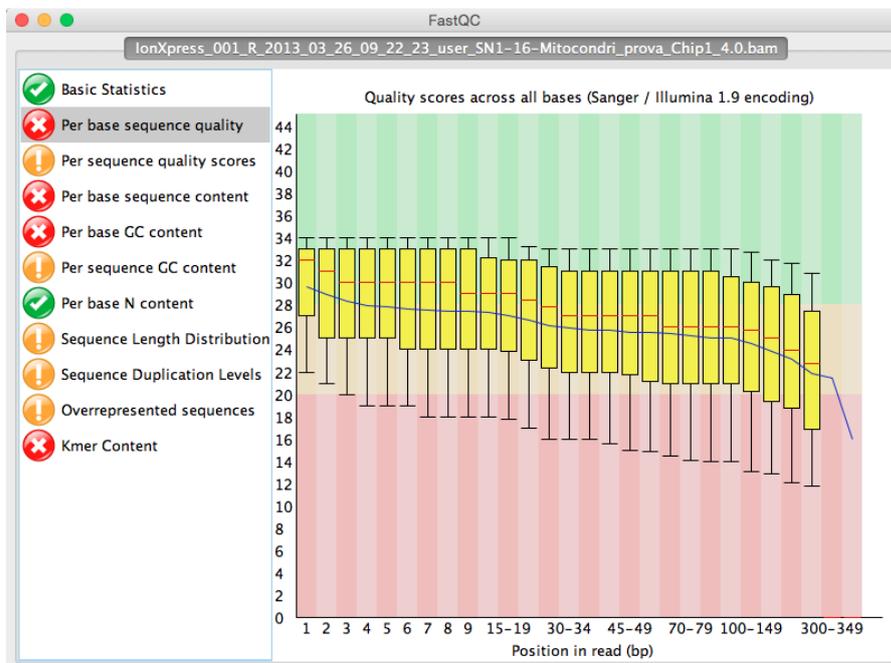


Figure 5: FastQC screenshot. On the left we have the list of available statistics, with semaphore-based colors depending on the statistics results themselves. On the right a clearly bad run, with mean quality scores dropping below optimal values (green shadow) early in the reads.

3.4.4 Swiss-army knives for genomics data management

Samtools (H. LI et al. 2009b), bedtools (QUINLAN et al. 2010) and Picard (Picard Website) are toolkits to manage, view and analyze high-throughput sequencing data. Combining them you get a fully-featured Swiss-army knife that can sort out many – if not all – the problems you may encounter in selecting, merging, splitting, indexing and viewing data in SAM and BAM, the most adopted formats for raw NGS data.

Detailing all the task they can perform will result in a 10 pages and counting long list. Eventually, only Picard was used in this thesis, but all of them were thoroughly tested. You will get a complete view of where we used Picard in the next chapter but, for the moment, just know that we used them in our quest of improving the variants called by an Ion Torrent platform.

3.4.5 GATK

Getting the data out of the sequencer and view them is the first step in a long pipeline. What you're really looking for when sequencing are the variants each subject harbors in his or her genome. Technically speaking, you must *call the variants* in that genome. Given the very high number of reads NGS sequencers produce, the amount of information you must consider to call a single variant may turn out to be enormous. You clearly need advanced algorithms to condense all this information in a single nucleotide call.

The *Genome Analysis ToolKit* (GATK) (DEPRISTO et al. 2011; MCKENNA et al. 2010) disentangles this problem combining complex statistical modeling with efficient implementations of the algorithms to increase the sensitivity and specificity of the calls, while also keeping low the CPU and memory footprint of the system. To help first-time users, the GATK development team published and constantly updates Best Practices for different use cases, with comprehensive examples on how to execute and master each command the pipeline is composed of (VAN DER AUWERA et al. 2013).

GATK HaplotypeCaller was used in this thesis to perform variant calling on samples sequenced with the Ion Torrent Platform.

3.4.6 HaploFind

HaploFind (VIANELLO et al. 2013) is a web application developed at our lab to perform high-throughput SNP discovery and haplogroup assignment of complete mitochondrial DNA sequences. Implemented in the context of my master thesis, the web application is now gaining a small but significant momentum among researchers, with a constantly growing user base.

The application allows to load a set of mtDNAs in FASTA format that are then automatically aligned against the mitochondrial reference sequence (RSRS, BEHAR et al. 2012) to detect variants. Once the mutational signature of each samples is defined, HaploFind compares it to the Human mtDNA Phylogenetic tree (PhyloTree, VAN OVEN et al. 2009) to find the best match between detected mutations and known variability. Results are then sent back to the browser and displayed in a Javascript-based web interface for navigation, refinement and exporting.

The whole process is usually very fast, with a lower bound of 0.09 seconds/sequence for the snp discovery and haplogroup assignment step, which increases to ~0.9 seconds due to the overheads caused by data insertion/retrieval in the database and job queue management.

HaploFind carried out the SNP discovery and mtDNA haplogroup assignment on GEHA sequences.

3.4.7 EpiData Entry

Epidata Entry (EpiData Entry Website) is, citing the developers' website, a software for

simple or programmed data entry and data documentation.

It allows to build complex yet easy to use interfaces for the data entry in the context of scientific projects and beyond. Validation of the data can be coded exploiting a dedicated programming language, that allows for the creation of advanced checks with relatively small efforts. It also has built-in function for data export and backup.

In this thesis we adopted EpiData Entry to build the data entry interface for the NU-AGE project.

4

RESULTS

INDEX

4.1	NU-AGE - the data challenge	25
4.1.1	The wealth of data	26
4.1.2	Why we must fight errors	28
4.1.3	How to fight errors	28
4.1.4	Striving to document	29
4.2	NU-AGE - Validating the data	29
4.2.1	Data entry interface	29
4.2.2	The need of a new solution	34
4.2.3	NU-AGE Validator - a quick introduction	34
4.2.4	NU-AGE Validator - The implementation	35
4.2.5	NU-AGE Validator - How it works	47
4.2.6	Validating continuous data, without fixed boundaries	48
4.2.7	Crunching everything - document to rule them all	50
4.3	NU-AGE - The @Bologna database	51
4.3.1	Why?	51
4.3.2	Which database?	51
4.3.3	Database deployment	51
4.3.4	Database description	52
4.4	NU-AGE Systems Biology	52
4.4.1	Where to start the journey from	53
4.5	GEHA and the Epistasis	64
4.5.1	Obtaining the sequences	64
4.5.2	Analyzing them	65
4.5.3	Single SNP association	69
4.5.4	Multiple SNPs association - Epistasis	70
4.6	Ion Torrent Data Analysis, an excursus of	75

In the last 3 years, I've been involved in many different projects – sometimes just an idea of a project – a part of which made their way in the Results of this thesis. The same premise made in Material and Methods holds strong also here: each project had its own requisites, which were addressed with the most appropriate computational answer. Sometimes 20 lines of script did the job, in other cases entire pipelines were coded from scratch. I will first start discussing the biggest challenge I personally faced, and then move on to the other topics.

4.1 NU-AGE – THE DATA CHALLENGE

NU-AGE aims to understand the role diet, in particular a fortified diet focused on elderly's needs, may play in longevity and healthy aging. To achieve its objective, the project collected an unprecedented amount of data, deeply characterizing 1250 subjects before and after a one year-long diet intervention followed by half of the enrolled cohort. This fine characterization encompassed different areas, from phenotypical data such as physical and cognitive indexes to DXA scans and cutting-edge omics analyses.

Cracking open the inner mechanisms that diet may use to impact on longevity must go through efficiently leveraging the collected data to find the needle in the haystack. NU-AGE, apart from the innate complexity of the intervention itself, represented a big challenge even for data managers. Tackling the data problem required to act at different levels, to ensure data were entered and stored consistently.

Table 3: NU-AGE questionnaires and assays schema

Questionnaire	Datasets	Timepoints
Admission 1	Admission 1	Time 0
Admission 2	Admission 2	Time 0
General	General Time 0	Time 0
	Prescribed Medicines To	Time 0
	General Time 1	Time 1
	Prescribed Medicines T1	Time 0
Interview	Interview Time 0	Time 0
	Interview Time 1	Time 1
Blood & Urines	Blood & Urines Time 0	Time 0
	Blood & Urines Time 1	Time 1
Energy Expenditure	Energy Expenditure Time 0	Time 0
	Energy Expenditure Time 1	Time 1
Supplement	Supplement Time 0	Time 0
	Supplement Time 1	Time 1
Follow-up	Follow up Month 4	Month 4
	Prescribed Medicines Month 4	Month 4
	Follow up Month 8	Month 8
	Prescribed Medicines Month 8	Month 8
	Follow up Month 12	Month 12
	Prescribed Medicines Month 12	Month 12
Vitamin D ^a	Vitamin D	Months 4, 8 and Time 1
DXA	DXA Time 0	Time 0
	DXA Time 1	Time 1
Nutrients	Nutrients Time 0 (7 days average)	Time 0
	Nutrients Month 4 (3 days average)	Month 4
	Nutrients Month 8 (3 days average)	Month 12
	Nutrients Time 1 (7 days average)	Time 1
Exit conditions	Exit conditions	Time 1

^a Contains Month 4, Month 8 and Month 12 data together.

4.1.1 The wealth of data

In defining the pipeline that would have dealt with NU-AGE data, the first step was to exactly define which information the project planned to collect, down to the level of each single variable. There's a first sharp distinction to be made here, separating:

- Phenotypical data, obtained via questionnaires;
- Data produced by assays, such as fresh blood analyses, DXA and omics assays.

which need to be managed in different ways.

There is also a second distinction regarding data: some are obtained by each recruiting center independently (i.e. questionnaires, DXA, fresh blood analyses), while others are centralized to reduce the biases different instruments and labs may cause. Table 3 provides a list of the firsts, while Table 4 on the next page details the centralized assays.

Roughly speaking, the number of variables datasets comprised in the first table will produce is ~2040. We still can't estimate the same number for the centralized analyses: for some of them the output cannot be exactly determined upfront, while for others involved labs are redefining the data formats on-the-fly to improve later usability.

After obtaining a clear picture of the data the project was collecting, we proceeded to design a proper *data management plan* to address the issues we

Table 4: NU-AGE Centralized assays

Assay	Timepoints	Analyzed subjects
Cytokines / Interleukines	Time 0	1250
	Time 1	1250
Citomegalovirus	Time 0	1250
Genetics (Illumina 750K)	Time 0	1250
Microbiota – Phylogenetics	Time 0	1250
	Time 1	1250
Ferritin – Soluble transferrin receptor	Time 0	1250
	Time 1	1250
Microbiota – UCC	Time 0	120
	Time 1	120
Serum – UCC – Calcium, Vitamin D	Time 0	1250
	Time 1	1250
Lipids	Time 0	1250
	Time 1	1250
Metabolomics – INRA	Time 0	120
	Time 1	120
Microbiota sequencing – HITChip	Time 0	120
	Time 1	120
Plasma – IFR – Hormones	Time 0	1250
	Time 1	1250
Plasma – IFR – Cytokines	Time 0	120
	Time 1	120
Metabolomics on serum	Time 0	120
	Time 1	120
Metabolomics on urine	Time 0	120
	Time 1	120
Metabolomics on feces	Time 0	120
	Time 1	120
Transcriptomics	Time 0	120
	Time 1	120
Proteasome / Immunoproteasome	Time 0	120
	Time 1	120
Serum – ORU – Liver functionality	Time 0	1250
	Time 1	1250
Telomere length on PBMC	Time 0	120
	Time 1	120
Telomere length on whole blood	Time 0	1250
	Time 1	1250

found, making life easier for people that would come later, data analysts on top of the others. This last consideration gently pushes us to the next section: errors, and why we must fight them.

4.1.2 Why we must fight errors

Massive projects, if not carefully managed, usually produce massive amount of dirty data requiring extensive cleaning afterwards, if any cleaning is feasible at all at that point. We all experienced the bad feeling of a shiny new dataset that promises wonders, only to find that some of the variables are missing, some shows suspiciously different behaviors depending on the recruiting center (SOPs problems) and, maybe, some are encoded in a unknown way. Sum small and not-so-small problems like these in all the datasets obtained within a project and you've a screwed research project that will end in a very big and thick smoke cloud.

The reason we must fight errors from the very beginning is obvious: not wasting efforts, time and, ultimately, money. NU-AGE is a complex project that aims to combine together data coming from completely different areas and must rely on a clean database to build its foundations upon. We can't think about advanced techniques if even a one-floor building will collapse in no time.

This leaves us with a big, red-flashing question: how to fight errors?

4.1.3 How to fight errors

Clean datasets require good plans on how to detect and fix errors in the data. In NU-AGE we committed ourselves to be quite aggressive against errors, acting at many different levels:

1. Data entry was strictly done via a data entry interface developed to validate the data on-the-fly, unless they were automatically produced by a software or lab instrument (i.e. DXA). In this second case, datasets were accepted without further checks.
2. Data gathered in with the data entry interface were exported and sent to UNIBO, where an in-house developed data validation pipeline re-validated the data from scratch, applying checks beyond the data entry interface capabilities.
3. Continuous data not having clear expected ranges (again, DXA is a good example here), which thus cannot be easily validated against a set of fixed rules defined *a priori*, were validated visualizing their distribution and pointing out outliers.
4. Each center data was uploaded in separate folders in the database, and merged together only afterwards. This allows for full control on information history (all uploads are retained in the database) while ensuring only the latest (and hopefully most correct) version of the data is available to authorized staff.
5. Data requiring specific skills to be validated were forwarded to partners within the project with recognized experience in the field of interest. Cleaned datasets were then sent back to the data management team and uploaded in the database.

As you may have already understood, the first and only aim of this multi-level validation pipeline was to detect errors quickly and as close as possible to the data entry time. It's much more easier to correct, or at least double-check, a value when source documents are still around, than if the same request comes in 3 whole years after the subject inclusion, with all the "paperwork" locked somewhere else.

4.1.4 Striving to document

Suppose in 5 years from now – after the project end – someone asks for the heart beat, or handgrip, values in the cohort. Which was the variable encoding the value? And, why subject 789 has a so low value? Was him bradycardic or something went wrong in the measurement? Should a data analyst throw away the subject as an outlier, or there's a reason behind the value and he should keep him? What about subject 566, which has the same problem? That's exactly the reason why documentation in scientific projects is a must have: data are valuable, and people should be able to use them in the future without getting mad asking themselves if a value valid or not.

NU-AGE obviously faced this issue: with 1250 enrolled subjects, it was only a matter of time before some strange - but valid - value made its way into the database. We thus decided to plan ahead: we gave recruiting centers an easy way to flag false positives they may encounter, provided they had a meaningful reason form them, while on our side these reasons were stored to track why failed checks were silenced. In this way, when the data entry process will be completed, we will be able to provide to data analysts an handbook detailing each exception along with its reason. At the same time, we're are assembling documentation about the variables, detailing from where each variables comes from, the logical block in which it was originally comprised, and, if applicable, question text, expected ranges and dictionaries used to encode the answers.

All together, these files will constitute a sort of "NU-AGE data manual" that will hopefully answer most of the questions researchers will have when playing with the data for the first time.

4.2 NU-AGE - VALIDATING THE DATA

What we just saw is the theory of data management. However, going from theory to practice needs a lot of efforts to translate these rules in softwares and, possibly, documentation. What will follow is an accurate description of how we implemented this theory in the NU-AGE project.

4.2.1 Data entry interface

Checks at data entry level are the first line of defense against errors in the datasets. It's probably the most effective measure: if the medical doctor or the researcher enters a value outside what was expected there's some sort of notification that immediately warns him about the issue and the information can be verified on the source documents in a matter of seconds. However, there's an intrinsic limitation in this approach: the systems behaves well only when there are clear sharp bounds that depends only on data already entered into the system. We can safely throw out an alert if someone tries to enter information about an abortion for a male (given we already know the subject is a male), but saying if that hearth beat is outside what it is normally expected, well, that may be a little bit trickier.

Anyhow, building a data entry interface requires a suite to base it upon, which should ideally have two key features:

- Easy to code *Case Report Forms* (CRFs, sometimes also referred to as eCRFs, where e stands for *electronic*);
- Easy to fill forms;

When we evaluated how to implement the CRFs for NU-AGE we had to consider the following points:

- No server was available to host a dedicated web application (i.e. OpenClinica, OpenClinica LLC and collaborators, Waltham, MA, USA, www.openclinica.com).

OpenClinica.com.) and acquiring a new one was off the table for several reasons;

- Time was an important factor, since recruiting was about to start and data entry should have started soon thereafter;
- We didn't have a solid background in this field, thus an easy to learn system was mandatory;
- Due to the short time to deploy, we had for sure to fix bugs after deployment, and we needed a plan for that;
- Ideally, concurrent usage by multiple users could have been useful, but it's very difficult to obtain it without a client-server architecture in place;
- Different centers had different needs, so flexibility in development and deployment was a must (this later forced us to maintain 3 different platforms).

After carefully evaluating the candidates, and with all the server-based excluded *a priori* due to server unavailability, we chose EpiData Entry (EpiData Entry Website) to develop our interface. This software had several advantages:

- Coding questions and validation checks was straightforward, being the whole system based on a trivial language and text files;
- Deployment could be done manually by each user, or automatically by a setup program;
- Each center had its own interface, so different needs could be addressed separately;
- Data were stored locally and then exported to the database. In this way, recruiting centers had immediate access to their own data, as per project agreements;
- Bugs could have been solved on a test interface in Bologna, and patches deployed via remote connection powered by TeamViewer (TeamViewer GmbH, Göppingen, Germany, <http://www.teamviewer.com/>);
- Being the interfaces physically separated, any technical problem occurring in one of them, i.e. not foreseen edge cases, would have affected only the data collected in that given center;

The only problem still open was to allow for the concurrent usage of the platform by different users at the same time. However, concurrency is quite a complex feature to obtain without relying on a client-server architecture. Indeed, EpiData Entry does not support this natively: once files are opened by a process (a user) they are locked for opening by others. However, after some discussion with the partners, we realized that the feature we were missing was indeed the possibility to access the system from different computers, and not concurrent access. In this case a workaround does exist: both the software and the configuration files were moved to a network share where authorized users had write access to and, since the program doesn't need to be installed to work, it was possible to launch the interface directly from within the share. This setup also has a side-advantage: most of the network shares provided by universities undergo backups on a regular basis, and moving the interface on those systems ensures that all the entered data are backed up automatically without additional effort.

So, granted that EpiData was probably the most adequate answer to our needs in that particular moment, we still had to code the CRFs. When we started developing the interface, back in 2012, there were two different implementations of EpiData available: an older, stable version with a lot of

field testing and a new, completely rewritten version that was in its early stage of development (usually referred to as *alpha* and *beta*). The newly developed platform was really promising: it solved most of the chronic problems the previous suite suffered from and made CRFs creation even easier. However, the codebase was so unstable at that point that the developers themselves discouraged the adoption in production use. Additionally, some features required to fully implement the NU-AGE interface were still missing. We were thus forced to adopt Epidata Entry v3.1, the last version available of the older Epidata implementation.

Implementing eCRFs

Implementing a basic CRFs in Epidata is relatively easy, provided that the developer has some programming rudiments. A data entry form is made of two files: .qes and .chk, where the first defines the questions and links them to the proper variable name and format and the second defines the validation checks to be used.

An example of how we declared the fields storing subject code, name of interviewer and date of administration in Admission 1 questionnaire (.qes file) is the following:

1	code	Subject Code	#####
2	interadm1	Interviewer	
	↪	-----	
3	interadm1d	Date of interview	<dd/mm/
	↪	yyyy>	

There's already some information in these lines: each variable is declared (first string, i.e. code) and linked to the corresponding question and the proper field type and length is defined, where:

- "#" stands for a figure, so 6 "#" symbols in a row mean an integer value of maximum length 6. Adding a dot, such as "#.#", codes for float values, in this case with a single decimal digit;
- "_" stands for a character, and the number of "_" is directly proportional to the maximum string length;
- <dd/mm/yyyy> codes for a date.

If no additional validation rules are needed in this form, the CRF is ready to be used. However, that's not the way things usually go: there are additional checks we need to apply to our data to detect wrong values. In EpiData these checks can be declared in the check file (.chk), and the checks counterpart for the questionnaire above is the following:

```

1 code
2   NOENTER
3   KEY UNIQUE
4   TYPE STATUSBAR "IDCODE = "
5 END
6
7 interadm1
8   MUSTENTER
9 END
10
11 interadm1d
12   MUSTENTER
13   AFTER ENTRY
14     IF interadm1d > Today then
15       HELP "Dates in the future are not allowed" TYPE
16         ↪ =ERROR
17       GOTO interadm1d
18     EXIT
19   ENDIF

```

```

19     END
20 END

```

There are three main blocks in this piece of code: each one starts with one of the variables we saw in the previous snippet, ends with a capitalized "END" and defines the validation checks for the variable whose name opens the block. Indentation should help in identifying them.

Skip the variable "code" for now – it has a special meaning – and focus on "Interadm1", which stores the name of the interviewer. This may be a valuable information in case discrepancies due to different ways of administering the questionnaires among the recruiters arise later on. Anyhow, there's really nothing we can validate here, apart of the fact that this field can't be blank, and that's exactly the job the "MUSTENTER" command does: it throws an error if the field is left empty. For "interadm1d", which holds the date of the questionnaire, we can instead do something more: again, this field can't be blank, but we can also make sure that the date is not in the future. Data entry is done *after* a questionnaire is filled (in the ideal situation on the same day), but never before. Thus, a date in the future doesn't make any sense in this context. EpiData gives us the bricks to build this check: the "AFTER ENTRY" block, which is invoked every time the field goes out of focus, i.e. the person moved to the following field, and the special variable "Today", which always contains the current date. To implement our check we compare "interadm1d", which stores the date the person just entered, to the current date to verify if it is greater (to be read: it is after). If the comparison is true, we throw out an help box of type "ERROR" with text "Dates in the future are not allowed" and, when the box is closed, we move the focus back on the "interadm1d" field with the GOTO instruction, so the error can be fixed. The EXIT command grants that, if an invalid value occurs, no further processing is done, preventing further errors to happen. Finally, the ENDIF instruction terminates the IF block.

We now go a step further analyzing the variable "code", which allows us to introduce a new concept: *relational eCRFs*.

Relational eCRFs

PostgreSQL, the DBMS we saw in section 3.3.3, can also be defined as a RDBMS, where R stands for relational. Briefly, relational databases are systems in which tables storing different type of information are linked together by mean of an unique key that is present in all the tables. Keys are the glue that keeps the whole database together. With EpiData you can do more or less the same: subdivide information within different questionnaires, and glue everything using a key. In our implementation, the variable "code" represents the key. Going back to the previous code snippet, you can see that the first command for that variable is "NOENTER", which prevents the user entering the field, since the code is automatically filled by EpiData and we don't want the user to fiddle with it. The next instruction is then "KEY UNIQUE", which means "this variable is a unique key, so the string entered here must be unique among all the records", exactly as a patient ID should behave. The last code line customizes the interface, displaying the current PID the user is working on in the bar at the bottom.

For NU-AGE, we exploited this relational architecture and built a bogus form that was used as a entry point to access all questionnaires. There are only two fields in this fake form: the PID the user is going to enter the data of and a integer field ranging from 1 to 17, which corresponds to a progressive id assigned to each questionnaire. Each CRF was implemented separately and hooked back to the main page via Epidata command "RELATE". Once the user enters the PID of the subject and the questionnaire id he or she wants to enter, the program automatically loads the CRF and, if a record with that PID already exists, displays the stored data. Otherwise, a new blank record ready to be filled is created.

The advantages this approach guarantees are obvious:

- Every questionnaire is separated from the others and must be accessed on purpose to modify the data;
- Only the selected CRF pertaining to the selected subject is editable. All other records are locked, decreasing the risk of data entry errors;
- Every questionnaire constitutes a separate dataset, that can be exported and managed independently. When you're dealing with 2000+ variables, this is an important plus.

Exporting data

After being entered in the interface, data must be exported in a format compatible with the rest of the data management pipeline. Since this step is carried out by the user, the process should be as user friendly as possible: lower complexity directly relates with fewer errors. Unfortunately, EpiData Entry does not provide any way of exporting all the CRFs linked to our main interface as a whole. Instead, the user would have to open each questionnaire one by one, export the data and then close it. It's not a only matter of repetitiveness: the critical point is that in this timeframe the questionnaire would be open for modifications, without the safe boundaries given by the relational architecture described above.

Luckily, the EpiData Association, the associations that steers the present and the future of EpiData, developed the EpiData Command Application (EpiC). This utility, that can only be invoked from a DOS command prompt, can export the data contained in an EpiData Questionnaire without external assistance in many different formats. In our case, we chose a text-based TAB-delimited format (see 3.3.1). Nonetheless, EpiC can do the job, but still one questionnaire at the time. EpiData Entry toolbox, however, wasn't empty yet: the "EXECUTE" command permits to invoke external programs from within a CRF. We thus coded a .bat file, a list of DOS commands that are executed one after the other once the file itself is executed, that did two jobs:

1. Delete any previous export for the given questionnaire;
2. Export the data for the questionnaire.

The file was then connected to the main page thanks to the "EXECUTE" instruction: when a user enters 17 as questionnaire id, no CRF is opened but the export procedure is triggered and exports build automatically in a previously configured folder.

Complexity is bad - how to configure everything easily?

Complexity, at least on the user side, is bad for a thousand reasons, some of which we detailed above. Once properly deployed and configured, the interface was very easy to use, even by people without specific training. Yet, properly configure the system remained a complex task, especially when dealing with the settings needed to configure the paths where the interface was supposed to export the data and store backups. Trusting in the user for these tasks was off the table, and we thus decided to implement an installer based on the excellent Nullsoft Scriptable Install System (NSIS), a professional open source software used to create Windows installers. The result was a easy-to-use installer that took care of asking users where exports, backups and the interfaces itself were to be installed and of customizing center-dependent checks (i.e. first digit of the NU-AGE PID is center-dependent, and we had a check on that). We also wrote a manual covering all the daily operations users may had to do, which also doubled as a first-aid help in case of minor issues.

All the initial interface deployments were powered by this automatic installer, but it was soon clear to us that that wasn't a "shoot and forget" problem. Most of the centers needed further assistance, so we adopted TeamViewer to fix issues and deploy updates remotely.

We briefly mentioned backups: EpiData do implements a backup routine, and we instructed it to do so each time it was closed, aggregating the data at day level. In this way, if needed, we were able to track the data entry process down to day-by-day changes.

4.2.2 The need of a new solution

Epidata was probably the wisest solution to address NU-AGE needs given the strict requirements we had, but was not downsides-free.

The first and biggest was (and is) the lack of central control once the interface is deployed. We're getting back again at the problem of having a server to manage the study and to the fact that, if proper resources are available, that approach represents the golden standard that should be followed. Monitoring the data entry process as it goes by, immediately spotting errors in the data, being them systematic or not, and pushing out fixes in a single place instead of chasing countless partners to schedule a TeamViewer connection are just some of the advantages a central management point offers.

Adopting Epidata Entry, we faced serious problems in building advanced checks due to the intrinsic limitations of the language it is based upon. In some cases, we were forced to omit a check because it was too complex to implement, or required writing a dedicated C library. Also easier checks may stumble on the very same limitations: you can't use data in questionnaire1 to validate a value in questionnaire2, since cross-questionnaire access is not supported. If you have a sex-dependent test in questionnaire2 and the sex of the subject is in questionnaire1? You can't alter the questionnaires structure obviously, so you must either rely on the data entered by the user, or ask him or her to fill the gender twice in both questionnaires. Please note: double entry is not a bad thing *per se*, but double entry to overcome problems like these is *pure evil*.

Another important requisite to keep in mind is the possibility to track the data failing the validation. Having values out of normal ranges is common, a light cold can alter people's blood values out of what is normally expected. We knew that recoding these discrepancies, together with the reasons for them, would later become of striking importance. Epidata, I would say rightly given the different scope the software has, doesn't support that. You can add subject-wide notes to each CRF, but without any structured information.

Finally, the biggest limitation we struggled against: checks in EpiData can be silenced simply using the mouse. Move along fields with the mouse, and no check will be executed. This was the result of a precise, and *sensible*, design decision: if you end up with a badly written check that blocks the rest of the form you can recover from the situation moving out of the troublesome field with the mouse. However, this had a sense when no updates or patches were possible, but in 2015 we have emails and remote connections, so this is much more a damage than a gain. Until you have a working internet access, of course.

From these considerations we started wondering if implementing a custom solutions was worth the costs in time and resources, and we eventually concluded that the answer was affirmative.

4.2.3 NU-AGE Validator - a quick introduction

NU-AGE Validator was the answer to all the problems we faced using EpiData as our data entry interface. Entirely built in Python, the software can parse the .qes and .chk files we wrote to define EpiData questionnaires and

use them as a source to rebuild its own representation of the CRFs. Once the structure is built, the systems loads the datasets and validates all subjects one by one. Each questionnaire, in Validator, can contain one or more questions and harbor questionnaire-wide checks. In turn, each question can define one or more checks, made from pre-defined or custom Rules.

Each check (hence, each Rule) produces a Validation Result (VR) independently if the check was successful or not. VRs are stored in a local database to be later used to provide a feedback on which checks didn't pass and why, in forms of two almost identical reports, based on the PDF and Excel format respectively. The PDF report is made by tables detailing the errors occurred in each questionnaire and, once printed, helps in skimming through the source documents searching for the original information. The Excel report, on the contrary, supports a more structured access and fulfills an additional, very important, role by mean of two additional columns: Authorize and Reason. These columns are part of the authorization mechanism that allows recruiting centers to flag false positives as such and represents an important piece at the basis of the handbook of NU-AGE data we will build once data cleaning will be completed.

The functioning of our authorization system is pretty easy from the users perspective: they need to respectively fill the two aforementioned columns with the string "Yes" and with the reason why they what to authorize the exception for each VRs they request the silencing. Validator then loads back these authorization requests and, provided that the reason is long enough, 10 chars or more, silences the corresponding VRs.

4.2.4 NU-AGE Validator - The implementation

NU-AGE Validator can be subdivided in several different libraries, each of them answering to a precise need of the pipeline:

- *Parsers*, which hosts parsers to read questionnaires and checks from configuration files;
- *Models*, which hosts the definition of all the objects needed for modeling questions and questionnaires;
- *Rules*, which hosts all the rules that can be used to build checks;
- *Wrappers*, which host wrappers around famous Python libraries, i.e. ReportLab to build PDFs, to make them more easy to use for our task;
- *Validator*, which is the script that glues everything together and carries out the work;

There's a pretty decent amount of code here: we're talking of ~10000 lines written in about 2 years of development, as visible in Figure 6 on the following page. For libraries that, for their complexity or pivotal role have a critical importance in the pipeline we developed specific *unittests*, which are automatic tests to verify that the code behaves as expected. This is part of the "best practices" in code development, since it allows to quickly verify if the latest changes a developer did on the code broke it somewhere else. A best practice is also to track these changes, to immediately rollback the nasty ones. In our case, we adopted git (Git Website), a distributed revision control system, to fulfill this requirement. After the git repository is initialized, code changes can be committed to it, thus obtaining a clear history of when the codebase was changed, how, and by who. Being honest, Git can help in all the situations in which text files are continuously modified and changes must be tracked. An example? This thesis, which was written in Latex (<http://www.latex-project.org/>), was tracked in a git repository!

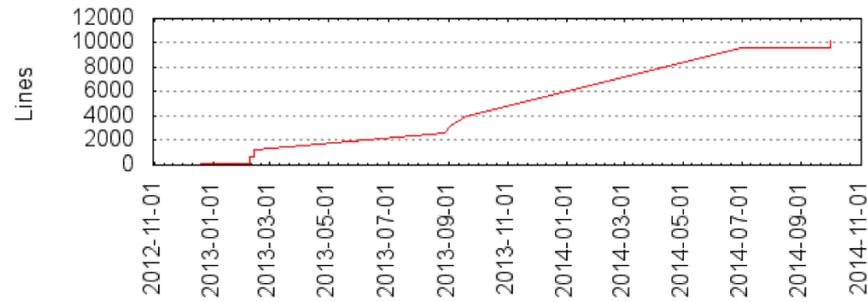


Figure 6: Validator repository size in line of codes in time. The development started in early 2013, and ended in November 2104. Small revisions to fix bugs or add functionalities are still developed and pushed out.

Models

The first step to comprehend Validator is to understand how the objects modeling questionnaires and rule were implemented. Let's start with the smallest unit we have in our construction set: questions.

QUESTIONS A question is the smallest brick we have in a questionnaire set. A question can belong to a structured test battery, or just be self-standing. Nonetheless, this is the level at which the information is linked to a variable and to its own space in the database. We can isolate a common structure among all questions, which we can resume as:

- A question text or, to use a computer science term, the representation of the question text in the form of a string;
- A variable, which represents where the data will be saved;
- A flag that marks the answer as mandatory;
- A progressive ID, which is the progressive number of the question within a questionnaire.

All these properties, taken together, can be seen as the core of an object modeling a question. Obviously, that's not the whole thing, but these are the *attributes* each question must have at minimum. How many types of questions do exists? From our point of view, we have 3 of them:

- Numeric question, which expects an answer in form of a number (integer or float);
- String question, which expects an answer in form of a string;
- Date question, which expects an answer in form of a date.

So we have three different theoretical models, each one acting differently in some cases, but which still have common attributes. Rephrasing the concept, we have a model inside a model, or, even better, a "parent" model from which the three inherit from. Python, as most object-oriented programming languages, allows for objects that inherit from other objects. To be more specific, allows for classes – that are custom data types defined by a developer – to inherits from other classes. Object, in *Object Oriented Programming* (OOP), are instance of a class, i.e a variable containing a string is a object of the Python built-in string class. The approach we followed was to implement a generic Question class owning the common attributes and then define three different classes (the three question types), which inherits from the parent class the common core.

Why three different objects to modeling a single object, a question? The reason behind that is that, depending on the expected answer, the object

should act differently. The maximum allowed length for the answer is a good example of this need: if we're dealing with a string, it's just the number of characters in it, but if it is instead a float number, it can be to the length of the whole number or the number of figures before the decimal point. Also, if we expect a string, we should throw out a warning if other data types come in. Particular care must be used when dealing with dates, since there are many ways of representing them (i.e. 12/02/2014, 12022014, 12/02/14 – same date, different representation) and in some edge cases it can be difficult to distinguish between correctly formatted dates and, for example, numbers.

Some validation checks are part of the question definition itself: mandatory/non mandatory flag, maximum length and wrong data type detection are good examples of these "by definition" checks. Each check returns a result (a VR object), which is not only a True or False flag, but also contains additional information regarding where and why validation failed. These objects must be collected within the question and then transferred on in the chain to be further processed. Thus, each question possesses an attribute, *check_results*, which is nothing more than a list (one of Python built-in data types) of VRs. Its companion, the attribute *checks*, holds the checks to be applied.

Questions also have *methods*, which are functions that help them do their work. The most important one is (fairly obviously) "*validate*", which triggers the execution of all the checks contained in a question. Other helper methods were coded, such as those used to add rules to a question or reset the object to validate the next subject, but they will not be described here due to space constraints. We will instead discuss two "special" methods that are used to deal with specific events in the questionnaire logic: jumps and missing codes. The first happens when a specific answer to a question causes the interviewer to skip one or more succeeding questions. Coping with this requires a tight interconnection between questions, the place where these deviations from the expected path may occur, and questionnaires, which in our implementation are the locations where the validation progress is monitored and, if needed, altered. Missing codes, on the contrary, are questionnaire-wide codes used when the interviewed subject is unable to give an answer or refuses to do so and the recruiter needs to report this condition. This event is managed checking the entered value against a list of legal missing codes stored in the Questionnaire object.

LABELS Numerical answers can be the result of numerically-coded questions, in which a list of options has a unique number assigned to each of the items. They're fairly common in the questionnaire world, primarily because they help to reduce data entry errors: it's much more easier to enter a single number than copy a whole phrase.

As EpiData, also Validator supports Labels, a named list of numerically-coded items a user must select from. In theory, since the numerical key is made of progressive numbers without any missing, a possible solution to this problem could be to define a range check with as boundaries the first and last number of the list. However, this provides poor support in case the validation fails: which was the item linked to the failing number? Which were the options available?

Sticking to the EpiData implementation of Labels, we implemented a class as much as possible similar to the definition EpiData gives of Labels. Let's look at the code:

```

1 class Label(object):
2     """
3     Class defining a Label class.
4
5     Label class is used to validate questions that have
    ↪ numerically-coded answers.
```

```

6
7     """
8
9     def __init__(self, name, labels_dict):
10        """
11        __init__ function to instantiate a Label object
12
13        labels_dict should be like:
14        labels_dict = {
15            1 : 'label 1',
16            2 : 'label 2',
17            3 : 'label 3'
18        }
19
20        :param name: label name
21        :type name: unicode
22        :param labels_dict: dictionary with keys and
23            ↪ corresponding labels
24        :type labels_dict: dict
25
26        """
27        self.name = name
28        self.labels_dict = labels_dict

```

In Line 1 we define the class with name "Label" that inherits from the class "object", which is the base class all programmer-defined classes should inherit from unless they are subclasses of other classes. We then have some greyed-out lines: these are documentation lines that are used to explain to people reading the code what that chunk of the program is doing. Documentation is extremely important, since the developer itself may not recall what that class, or method, was meant for at the beginning. Further following the code, line 9 says "def __init__(self, name, labels_dict)". We are defining a method here, in this case the special method "*__init__*", which is called when a new object must be instantiated. This method requires some arguments passed to it, namely:

- self, an argument all methods within a class require, represent the object itself;
- name, the name we would like to assign to the label;
- labels_dict, a dictionary in the form *key : value* where the key is the progressive number and the value the string of the corresponding option.

Once the "*__init__*" method is called, it carries out two operations: assign the content of the variable "name" to the class attribute *self.name* and the content of "labels_dict" to *self.labels_dict*. This piece of code is why the "self" argument is needed when invoking the function: it must be possible to set (or read from) class attributes, that are part of the class it-"self".

Obviously, there are other methods defined in this class, but we're not discussing them here for space reasons. For the sake of understanding how the object works, just know that they are used to verify if a given number is comprised within the label and to properly format the label object when printed.

QUESTIONNAIRE We now go one level up in our modeling, to the Questionnaire, which is in turn composed by one or more questions and some additional information. The attributes this class defines are many:

```

1     self.name = name
2     self.valid = False
3     self.pid = pid
4     self.qid = qid

```

```

5     self.record = record
6     self.questions = {}
7     self.checks = []
8     self.check_results = []
9     self.labels = {}
10    self.jump_to = None
11    self.validate_jump = False
12    self.ref_set = None

```

You may recognize some old friends in here, namely *checks* and *checks_results*, but there are also new entries. *Name* holds the string with the questionnaire name, *valid* is a boolean flag that states if the questionnaire is valid or not, *pid* is the famous Patient ID, *qid* a progressive id of the questionnaire within a questionnaire set, *questions* a dictionary (we just saw them) that contains all the questions, *labels* contains the Label classes available in the questionnaire, while *jump_to* and *validate_jump* are need to deal with jumps management and, last but not least, *ref_set*, which points to the questionnaire set (more on that in a moment) the questionnaire belongs to.

We skipped an attribute, *record*, because it deserves an additional explanation. Usually, each subject will go through a single questionnaire per type. However, there are cases in which this condition does not hold, for example when filling the list of prescribed medicines: we have multiple records for a single subject, all with identical IDs, but we still need to track to which entry we are referring to. Here is where the *record* attribute kicks in, keeping track of the progressive id each entry has within each subject record.

Questionnaire class, as all the other classes we have seen until now, defines many helper methods, that we will not list here for sake of brevity. They encompass multiple functions, ranging from enabling and disabling questions to adding and removing rules and labels. Finally, there's the validate methods, which in this case deserves a few more words: questionnaire can hold checks that, due to their nature, are impossible to assign to a single question, and that is the reason why also this class has the *checks* attribute.

QUESTIONNAIRE SET Last but not least, we have the class `QuestionnaireSet`. The reason for this class to exist is to group together questionnaires that are meant to be validated together, provide some functions to properly manage or update the set and have a single point from which validation can be started. `QuestionnaireSet` allows, as `Questionnaires` and `Questions` do, to define `QuestionnaireSet` checks.

Rules

Validating data can be a tough work, with many different checks to be done, each one depending on different rules. NU-AGE, with its 22 questionnaires (counting the same questionnaire administered at different times twice), needs a fairly high number of rules to cover all the required checks. Some of these rules follow a reusable logic (i.e. bigger than, in range), while others cannot be easily transferred to other cases (i.e. PASE score calculation). Developing Validator, we decided to separate reusable rules, which are hosted in the codebase, from case-specific rules that are kept in an auxiliary file, the content of which is loaded at runtime. This is pretty much a semantic distinction, the inner working of both is very similar, but helps keeping the repository clean of unneeded code.

Validator defines 16 core rules, which are reported in Table 5 on the next page. As for Questions, also rules are divided in 3 sub-types: Numerical, String and Dates, plus an additional type called "Conditional", which is used to deal with rules that must check several conditions⁸ at once.

VALIDATION RESULT The same library also defines the VR (Validation Result) class, which is returned by all Rules to describe the outcomes of the

Table 5: NU-AGE Validator Rules

Rule Type	Rule Name	Checks if...
Numerical	GreaterThan	the entered value is greater than the base value.
	SmallerThan	the entered value is smaller than the base value.
	EqualTo	the entered value is equal to the base value.
	InRange	the entered value is between the defined range.
String	StringEqual	the entered string is equal to the base string.
	StringDifferent	the entered string is different from the base string.
	StringShorter	the entered string is shorter than the base string.
	StringLonger	the entered string is longer than the base string.
	StringLengthEqualTo	the length of the entered string is equal to the base value.
	StringLengthBetween	the length of the entered string is between the provided bounds.
Dates	Before	the entered date is before the base date.
	After	the entered date is after the base date.
	SameDate	the entered date is identical to the base date.
	BetweenDatesRange	the entered date is comprised within the provided dates.
Conditional	Conditional	a list of conditions in a questionnaire are true. If yes, the questionnaire is invalid.
	ConditionalEmpty	a list of condition in a questionnaire is true and a specific field is empty. If yes, the questionnaire is invalid.

validation process. Each VR needs some mandatory information to be created, which can be provided in form of the Rule Object originating the VR from which the init method will autonomously extract the needed data, or as single arguments. Required parameters are:

- pid, the subject pid;
- qid, the numerical code of the questionnaire where the VR was produced;
- record, the record where the VR was produced;
- variable, the variable producing the VR;
- prog_id, the progressive id of the question producing the VR;
- help_text, an help text to provide to the user to help him fixing the error;
- rule_id, an unique id composed by a combination of the Rule name and defined constraints.

If, and only if, the VR reports a failure it is also mandatory to provide the failing value. Taken together, all the attributes stored in a VR do represent an unique firm that clearly defines which rule is failing in which subject: in other words, a unique *key*.

EXTERNAL RULES Some of the rules we defined to validate NU-AGE data were too specific to be included in Validator core and are kept in a separate library, loaded at runtime. In this way, Validator codebase remains slim, while developers can add all custom checks they need. In NU-AGE, a total of 27 external rules were defined.

An External Rule must fulfill all these condition to be considered valid:

- Inherit from Rules class;
- Possess the methods:
 - `__init__`;
 - `validate`;
 - `build_id`, to build the unique Rule ID used in the VR creation.
- Always return a VR as the final result of the validation process.

Defining an external rule is just half of the job: developers must also tell the system where these rules must be applied to have a working check. To streamline this process Rules library contains the ExternalRule class, which has two methods: one for initializing the class (the "`__init__`" we already saw) and one for loading the rules into the appropriate questions.

The initialization function accepts two arguments: a list of destination questionnaires with the variable where checks should be loaded into, and the Rule object of the custom rule. On the contrary, the loading function accepts only a single argument: the QuestionnaireSet containing the destination questionnaires. This second function iterates over the provided destinations, loads the questionnaires from the QuestionnaireSet and injects the Rules in their destination questions.

For example, the following piece of code

```

1 ExternalRule(destinations_list=[
2     (['NU-AGE Interview Questionnaire - Time 0',
3       'NU-AGE Interview Questionnaire - Time 1'],
4       'antbmi')],
5     rule_object=BMI(variable='antbmi')
6 ),
```

loads the Rule BMI (computes the BMI starting from height and weight) in the variable "antbmi" inside the questionnaires "NU-AGE Interview Questionnaire - Time 0" and "NU-AGE Interview Questionnaire - Time 1".

Loading checks in this way can be helpful in a second case: suppose you've a set of questionnaires in EpiData and you load them in Validator thanks to the parsers library. However, the CRF designer left out a check you now want to add and the corresponding rule is already comprised in Validator core. This need can be easily fulfilled by defining an ExternalRule, as described in this example:

```

1 ExternalRule(destinations_list=[
2     ('NU-AGE Interview Questionnaire - Time 0',
3      'NU-AGE Interview Questionnaire - Time 1'),
4     'sys1']),
5     rule_object=InRange(90, 200, help_text="Systolic value
6     ↪ should be within 90 and 200", variable='sys1')
7 ),

```

Here we have an InRange rule, which is coded in Rules library, which will be added to the variable sys1 (systolic pressure) on top of the configuration originally defined in the CRF. This is a real world example: in NU-AGE we intentionally avoided checks on blood pressure values after receiving suggestion in not doing so by recruiting centers. However, when we realized that in this way we offered an easy game for absurd values to reach the database, we added a range check in Validator, without losing time in going back to each EpiData interface.

Parsers

We have now understood the basics of how a Questionnaire is modeled in Validator, how checks can be defined starting from Rules and even how to add specialized Rules that are not comprised in Validator core. We still, however, need to load the configuration defining questionnaires, rules and so on.

Validator has an automatic way to accomplish this task using parsers. Parsers are pieces of code that read a format and returns back native, in this case Python, objects. NU-AGE data entry interface was made with EpiData Entry, so we designed a parser to interpret that format and create Validator objects starting from EpiData files.

Later down the project we realized that it would have been helpful to validate also datasets not coming from questionnaires, i.e. DXA assays results. One possible solution was to configure a fictitious questionnaire in EpiData format, but spending time in writing a now abandoned format (EpiData Entry is no more supported, in favor of the new version of EpiData) seemed us awkward. We thus decided to draw from the long list of serialization formats to choose the one that better matched our needs, and we eventually adopted YAML (YAML Website), a human-friendly data serialization standard for which Python has a nice library to interact with. On top of this library we built our parser, which allows to quickly and easily define new datasets to be validated, along with the needed checks, in a single YAML file.

EPIDATAPARSER Questions and checks are contained in two separate files in the EpiData format: it is thus mandatory to parse both to fully reconstruct the questionnaire structure. However, correctly understand the schema of a .qes file can be really complicated, since it also defines questions formatting. EpiData Entry can build, combining the .qes and .chk files together, a .not file, which includes most of the information we had in the two initial files within a single source. Here's a snippet of it:

```

1 Fields in data file:
2

```

No.	Name	Variable label	Field type	Width	Checks
1	code	Subject Code	Number	6	Key unique 1 NoEnter
2	interblur	Interviewer	Text	20	Must enter
3	interblurd	Date of interview	Date	10	Must enter More: See check file

It's rather easy: a table detailing most of the information we need to build our Questionnaires. However, this format has several shortcomings:

1. Columns are separated with spaces, not tabs. To be precise, each column is separated from the next one by a different number of spaces, which makes parsing the file extremely complex;
2. If the text of the question ("Variable label" in the example above) is too long, it gets truncated. So, to have the full question text we must in any case parse the .qes file.
3. Not all the checks are reported here. The "More: See check file" string in the example says exactly that: there are custom checks for that variable, and you should read the .chk file as well.

The .not file is certainly of help, but does not resolve the situation.

So, resuming: we need to parse three different files, but most of them doesn't have a clearly defined structure. Given the requirements, regular expression was the answer to our problem. In computer science, regular expressions are sequences of characters defining a search pattern, i.e. we may look for something like "A number of three figures, followed by one or more tabs and by a word of length 3". We adopted this approach to parse the three files: regexp, helped by Python functions for more basics and predictable tasks, extract the information, which is then processed to determine which question or rule type to use depending on the situation, ultimately reconstructing each Questionnaire. All questionnaires are then grouped together in a QuestionnaireSet for manipulation.

`YAMLPARSER` `YAML` (or *YAML Ain't Markup Language*) is a data serialization format that focus on human friendliness. It is easy to read and write for humans and provides a fully fledged environment where to define a Questionnaire object along with its validation checks.

To prove you this claim, here's a piece of code defining the first two variables in our DXA dataset:

```

1 ## DXA TO dataset
2 questionnaire: NU-AGE DXA TO
3 missing_codes: [-9]
4 questions:
5   - variable: code
6     text: code
7     field_type: int
8     max_width: 6
9     mandatory: True
10  - variable: Android/gynoid fat mass
11    text: Android/gynoid fat mass
12    field_type: float
13    max_width: 20
14    mandatory: True
15    checks:
16      - type: InRange
17        lower_bound: 0.27
18        higher_bound: 1.11

```

```

19 help_text: Android/gynoid fat mass should be
      ↪ between 0.27 and 1.11

```

This is a clear example of a programming language that does its best to behave as nearly as possible like human language. The structure is straightforward: attributes are defined one per line with a "*attribute : value*" schema, whereas lists are defined simply indenting the items and prepending them with a "-".

Python has a library that automatically parses the YAML format and gives back its content in ready-to-use Python data types, making the whole shebang of reconstructing YAML-based questionnaires much more easier. For comparison, EpiDataParser is made of 824 lines of code, YamlParser of only 229. All the difference is made by the need of parsing from scratch the format in the first, while in the second we just need to instantiate the correct objects starting from the data already parsed by Python *yaml* library.

If YAML files are detected, Validator automatically parses them and adds the questionnaires to the QuestionnaireSet generated by EpiDataParser, or creates a new one if it doesn't exist already.

Wrappers

Wrappers in computer science are thin layers of code that translates an existing library interface into a more "compatible" interface for the problem you're addressing. Usually, these layers are created to lower the burden of complex libraries with thousands of options for novice users or for users that don't need such a level of customization.

There are three wrappers in Validator:

- DBWrapper, to interact with databases, built on top of SQLAlchemy;
- PDFWrapper, to create PDFs, built on top of ReportLab;
- ExcelWrapper, to create Excel files, built on top of openpyxl;

The role of each of them is to decline the library they are based upon in a way that nicely interacts with Validator classes and objects, making building a PDF or Excel report a matter of a couple of lines of code.

DBWRAPPER Probably the most important wrapper among the three, DbWrapper allows Validator to talk with a SQLite database to track and update the validation errors found in NU-AGE data.

The wrapper is based on SQLAlchemy (see 3.2.2) and uses the declarative approach to define three models used to store the information needed to build the reports:

- RecCentre, which hosts information regarding the recruiting centers;
- Questionnaire, which allow to link questionnaire IDs, the *qid* attribute we saw before, with questionnaire names and their correct ordering;
- VR, which is the database counterpart of the Python class we encountered in the Rules library.

As usual, let's start with an example of the code. VR models are defined in DBWrapper as follow:

```

1 class VR(Base):
2     """
3     Validation Result (VR) mimicking ad db level to store
      ↪ Validator produced VRs.
4
5     """
6     __tablename__ = 'validation_results'

```

```

7  __table_args__ = (UniqueConstraint('pid', 'qid', '
    ↪ record', 'rule_id', 'cid', 'variable', 'value',
    ↪ name='_error_code_uc'),
8      {'sqlite_autoincrement': True}
9  )
10 id = Column(Integer, primary_key=True)
11 pid = Column(Integer, nullable=False)
12 qid = Column(Integer, ForeignKey('questionnaires.qid'),
    ↪ nullable=False)
13 prog_id = Column(Integer, nullable=False)
14 record = Column(Integer, nullable=False)
15 rule_id = Column(String, nullable=False)
16 cid = Column(Integer, ForeignKey('recruiting_centres.
    ↪ cid'), nullable=False)
17 valid = Column(Boolean, nullable=False)
18 help_text = Column(String,)
19 variable = Column(String, nullable=False)
20 value = Column(String, nullable=False)
21 authorized = Column(Boolean, )
22 authorized_on = Column(DateTime)
23 active = Column(Boolean)
24 inactive_from = Column(DateTime)

```

If you bother going back to the VR definition in the Rules library (4.2.4), you can realize that most of the attributes are identical, except for 4 newcomers, namely:

- "id", the progressive id of the stored VR;
- "cid", which refers to the id of the center stored in the RecCentre table;
- "authorized" and "authorized_on", which track if a VR was authorized and when;
- "active" and "inactive_from", which track if a VR happened in the last validation, so it must be considered active or, alternatively, when we saw it the last time.

Before proceeding, let's focus a moment on attributes "qid" and "cid", which are declared differently from other attributes, as you can see in the code. That's because they are foreign keys pointing to records contained in tables Questionnaire and RecCentre, respectively. Foreign keys are the database parallel of the keys we saw in EpiData: unique ids that allow to link together information contained in otherwise unrelated tables.

We should also keep in mind that some error may occur when interacting with SQLite, and we don't want to have truncated or invalid data in our database. Luckily, databases allow for rolling back changes done in the current session, and SQLAlchemy supports this function. Within Validator, there's a small piece of code that monitors queries execution and, if any exception is raised, rolls back the changes and propagates the original exception for debugging. In this way, the database content is preserved from damage.

PDFWRAPPER Having validation results stored in a database is useless unless you get them out in some usable form. When we started thinking about which was the most useful way of telling centers what went wrong in the validation using a non-modifiable format, the first option we thought of was obviously a PDF file. Everybody has a PDF reader installed nowadays, and the fact of being a read-only format granted us that information would have reached the final user untouched. The de-facto standard for PDFs creation in Python is ReportLab (ReportLab Website), an easy to use library with thousands of tutorials available online.

Cutting a long story short, what you need to do to create your PDF is defining the paper format you want to obtain and then fill what ReportLab

NU-AGE Admission 2 Questionnaire (QID: 2)

The database currently stores data for **275** subjects.

Errors table (Total of 6 errors)

PID	Variable	Record	Value	Rule Id	Help text
100072	interadm2d	1	20/11/201	InvalidDate	Date is invalid
100161	p6weight	1	-9.0	MissingCodeApplies	Missing Code applies
100161	w6change	1	-9.0	MissingCodeApplies	Missing Code applies
100161	p12weight	1	-9.0	MissingCodeApplies	Missing Code applies
100161	w12change	1	-9.0	MissingCodeApplies	Missing Code applies
100224	interadm2d	1	06/06/201	InvalidDate	Date is invalid

Missing table (Total of 0 missing codes used)

PID	Variable	Record	Value	Rule Id	Help text
					No missing codes in this questionnaire! Hurray!

Figure 7: Example of test PDF report produced by PDFWrapper. Test page detailing errors and missing codes in the Admission 2 Questionnaire, ordered by PID and progressive id of the question.

documentation defines a "Story": a list of elements that will then be assembled together, being them images, titles or whole pages. PDFs produced by PDFWrapper are divided in two separate section with different aims. The first section is made by a summary page that gives a bird's view on the number of VR in the database, subdividing them in active, inactive, valid, invalid and authorized. The second section, on the contrary, hosts a long list of tables, two for each questionnaire, with the first containing detected errors and the second variables where missing codes are present. This second table is there for the sole purpose of having recruiting centers "touch" the amount of missing codes they're using, and double-check if they simply didn't remember to go back and fill the missing data.

Once all the elements are in the Story, ReportLab combines them producing a ready-to-use report ready to be sent back to the centers, a screenshot of which can be seen in Figure 7.

EXCELWRAPPER PDF reports are wonderful to display information, but when it comes down to provide an even minimal interaction with the data, they are quite poor. They are not designed to do so, after all! What is the most adopted format/program in the world for working with tables and data? Without any doubt, the first place goes to Excel.

Python has several libraries to interact with Excel files, but at the moment of writing the code only openpyxl (openpyxl Website) offered compatibility with the new XLSX format, and was thus adopted for our wrapper.

For most part the of job what ExcelWrapper does is identical to the job done by PDFWrapper, the only difference being that instead of different tables we here have different worksheets. Nonetheless, there is an important distinction to be done: each worksheet has two additional columns if compared to the tables built by PDFWrapper: Authorize and Reason. As previously said, these columns are used to allow centers to report back false positives, and part of the code inside ExcelWrapper is devoted to reading back these authorizations, verify if they are valid and, if the answer is affirmative, update the database accordingly. Once authorized, failing VRs will be ignored in following validations.

A screenshot of an Excel test report can be seen in Figure 8 on the next page.

	A	B	C	D	E	F	G	H	I
1	NU-AGE Admission 2 Questionnaire								
2	QID	2							
3									
4									
5	QID	PID	Record	Variable	Rule ID	Value	Help text	Authorize	Reason
6	2	100072	1	interadm2d	InvalidDate	20/11/201	Date is invalid		
7	2	100161	1	p6weight	MissingCodeApplies	-9	Missing Code applies		
8	2	100161	1	w6change	MissingCodeApplies	-9	Missing Code applies		
9	2	100161	1	p12weight	MissingCodeApplies	-9	Missing Code applies		
10	2	100161	1	w12change	MissingCodeApplies	-9	Missing Code applies		
11	2	100224	1	interadm2d	InvalidDate	06/06/201	Date is invalid		
12									
13									
14									
15									

Figure 8: Example of test Excel report produced by ExcelWrapper. Results for each questionnaire are splitted in different worksheets, while each row represents a VR. You can clearly see here the columns "Authorize" and "Reason" used to report false positives.

4.2.5 NU-AGE Validator - How it works

We spent quite a lot of time (and pages) in detailing the components upon which Validator is built, ranging from questionnaires loading to validation and reports building. It is now time to see how these small pieces work together to obtain the final result.

As many sci-fi films have proved you, serious people do work writing commands in a small black box called bash (honestly, bash is not the only option, but this would end up in a really, really, long discussion). Validator, being a respectable software, can only be invoked in this way. So, writing

```
validator.py
```

and pressing enter will show all the options you have to launch validator.

The first distinction the software asks you to do is if you want to validate new data or read back authorizations from an excel file. Obviously, required arguments do differ depending on the task you need to do. For brevity, we will ignore the part regarding authorizations, and stick with validating new data.

The list of parameters you need to specify in this second case is the following:

- `rules_folder`: where Epidata / YAML files to be use to create the QuestionnaireSet are stored;
- `data_folder`: where the files containing datasets referring to the questionnaires are stored;
- `database`: path to the SQLite database;
- `rec_center`: the progressive ID of the recruiting center to which data belongs to. This is a safeguard agains loading the wrong data by mistake;
- `external_checks`: path to the file where ExternalRules are defined;
- `metadata`: path to the file where the mapping between Questionnaires and datasets names and questionnaires order is stored;
- `promote_list`: path to the file that contains the list of variables that cannot have missing values. If a variable in this list has a missing value, it is listed as an error;
- `output_prefix`: a string prefix to prepend to the output files. This is handy when the destination folder of the reports is identical independently from the center under analysis.

Now that you have all your arguments set, you press enter. What is going on next? First of all, the program fetches, starting from the center id you provided, the name and location of the recruiting center, displays them for verification and awaits your authorization to proceed.

Once authorized, it proceeds by parsing questionnaires and checks definitions with EpiDataParser and YamlParser and building the QuestionnaireSet. Then, metadata are loaded and the number of listed questionnaires in the file is compared to the number of questionnaires within the QuestionnaireSet. If they match, the program continues loading into the database any new questionnaire the metadata may contain. Then, it's ExternalRules turn to be loaded and injected in the questionnaires. With this operation, the QuestionnaireSet is ready to be used to validate data.

Validator then scans the directory containing the datasets, looking for files with extension ".txt", ".xls" or ".xlsx", the three formats the software supports. Depending on the format, different Pandas (see 3.2.2) methods are used to load the data in memory while creating an index containing all the PIDs present in the files. This index will be used in the succeeding steps to validate subjects one by one.

The first real step of validation is the inactivation of all the stored VRs regarding the center under analysis. Then, iterating over the subjects index, data regarding each subject are extracted and progressively validated against the Questionnaires. Once a Questionnaire is validated, VRs are compared to those stored in the database: if a record is already present it is flagged as active and updated, otherwise a new record is created from scratch. When all subjects are validated, a DBWrapper method is invoked to set the inactive_from field of all newly inactive records to the current date and time. Finally, the session is committed and changes flushed to the database.

The last step is building the reports. Thanks to the wrappers that take care of the heavy lifting, building exports requires four lines of code:

```

1 report = PDFWrapper(args.database[0], args.rec_center[0],
2     ↪ promoted_variables)
3 report.build_report(args.output_prefix[0])
4 excel = ExcelWrapper(args.database[0], args.rec_center[0],
5     ↪ promoted_variables)
6 excel.build_report(args.output_prefix[0])

```

where the arguments are easily recognizable as the parameters we provided when launching the software.

The last validation of Italian data reached the whooping number of 147710 active VRs, that gives a pretty good idea of the quantity of work Validator does every time it is launched. The amount of time the process will take is not easily predictable, being heavily dependent on the number of errors, thus VRs, produced which is in turn function of the data amount and quality. To give you at least an idea of the order of magnitude, the Italian data validation reported above took about 13 minutes.

There's still plenty of room for improvement in the code that would reduce its running time, maybe consistently. However, Validator is just at the first iteration, a beta version if you want to define it so, which, as is, is already enough to answer NU-AGE needs.

4.2.6 Validating continuous data, without fixed boundaries

Not all data are categorical. Many important parameters are indeed continuous values, which cannot be validated just setting rules because of their innate characteristic: coming from all but predictable biological entities.

Data regarding blood pressure, height, weight and most of the values obtained via fresh blood assays can't be cleaned defining hard limits and

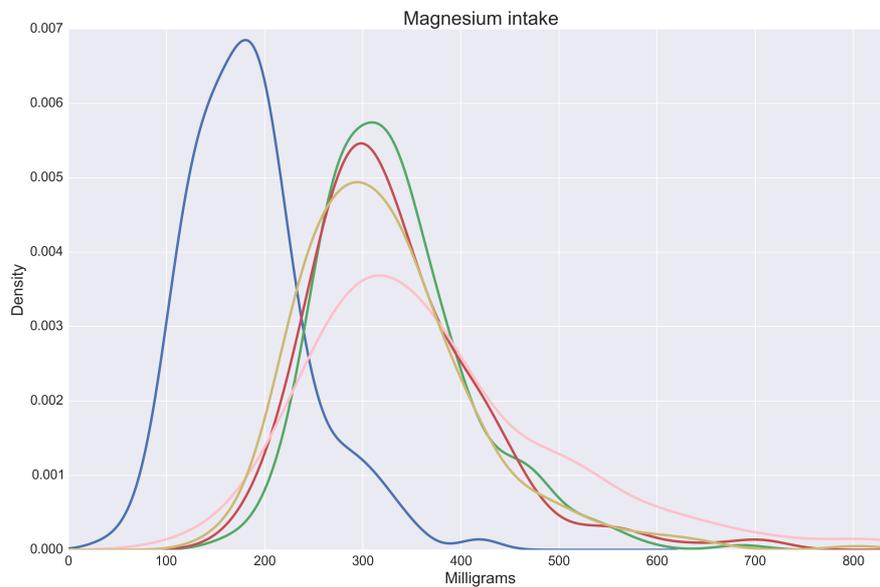


Figure 9: Probability density function plot of a parameter suffering from systematic errors. A center shows striking lower values for this parameter in all its subjects, while single outliers coming from several centers appear at very high values on the X axis.

throwing away everything that falls out. What we initially tried to do in NU-AGE was getting in touch with medical doctors in the 5 recruitment teams to understand if setting limits to these values was possible and if a consensus may have been built on which were the correct boundaries. However, it rapidly became clear to us that a sharp bound could not be defined.

Blood values, such as glucose and so on, have healthy ranges, but they do change from country to country. To partly overcome the problem, we collected these ranges, combined them to obtain a "general" interval of expected values and then set the data entry interface to accommodate all the values contained within this range. If a value fell outside, a simple warning was given and data entry allowed to continue. This was probably not the tightest possible way to proceed, but at least protected us against non-sense values.

Nonetheless, the described approach can only be used when a reference range exists. DXA and Nutrients, for example, are not so fortunate, and require a different treatment. Since in this context validating a single value is not possible, focus must be moved on validating the whole distribution of the values, and that's exactly what we did in NU-AGE. When the data entry of T0 values was completed, we plotted the distribution of the continuous variables looking for outliers. As you can see from Figure 9, it is rather easy to spot a systematic difference: one center does have striking lower values for this parameter in all its subjects, and single outliers from various centers appear at very high values in the X axis.

But, you may argue, this is a manual process, how to automatize it? Validator could do the job? Automatic outlier detection is an area in statistics that is still highly debated today. There are many methods, i.e. elliptic envelope fitting and One-class SVMs, but there is no general agreement on which represents the best answer to the problem, also because the answer depends on the distribution the values under analysis do have. An Elliptic Envelope can do wonders with a unimodal gaussian distribution, but is rapidly overtaken by SVM approaches when things get worse.

After some unsatisfactory initial tests in which we tried to find a "one-solution-fits-all" answer, we decided to leave automatic checks of this kind out of Validator. We instead computed for each continuous variable the 5th

	A	B	C	D	E	F	G	
1	Variable	Data Source	Hierarchical Level 1	Hierarchical Level 2	Field Type	Text	Structure	Range
2	code	NU-AGE DXA T0	CODE		int	Subject Code	#####	[None]
3	Whole body_Weight	NU-AGE DXA T0	Whole_body		float	Whole body weight (g)	Not regulated	[52468
4	Whole body_Fat mass	NU-AGE DXA T0	Whole_body		float	Whole body fat mass (g)	Not regulated	[10596
5	Whole body_Lean mass	NU-AGE DXA T0	Whole_body		float	Whole body non-bone le	Not regulated	[33015
6	Whole body_BMC	NU-AGE DXA T0	Whole_body		float	Whole body bone miner	Not regulated	[1595.:
7	Whole body_Soft tissue	NU-AGE DXA T0	Whole_body		float	Whole body soft tissue (Not regulated	[51117
8	Whole body_Fat mass %	NU-AGE DXA T0	Whole_body		float	Whole body fat mass pe	Not regulated	[16.75,
9	Whole body_Regional fat mass %	NU-AGE DXA T0	Whole_body		float	Whole body regional fat	Not regulated	[16.2, :
10	Whole body_BMD	NU-AGE DXA T0	Whole_body		float	Whole body bone miner	Not regulated	[0.87, :
11	Whole body_T-score	NU-AGE DXA T0	Whole_body		float	Whole body T-score	Not regulated	[-2.7, 2
12	Whole body_fat mass/lean mass	NU-AGE DXA T0	Whole_body		float		Not regulated	[0.2, 1.
13	Upper limbs_Weight	NU-AGE DXA T0	Upper_limbs		float	Upper limbs weight (g)	Not regulated	[5355.:
14	Upper limbs_Fat mass	NU-AGE DXA T0	Upper_limbs		float	Upper limbs fat mass (g)	Not regulated	[1057.:
15	Upper limbs_Lean mass	NU-AGE DXA T0	Upper_limbs		float	Upper limbs non-bone le	Not regulated	[3066.:
16	Upper limbs_BMC	NU-AGE DXA T0	Upper_limbs		float	Upper limbs bone miner	Not regulated	[201.0:
17	Upper limbs_Soft tissue	NU-AGE DXA T0	Upper_limbs		float	Upper limbs soft tissue (Not regulated	[5145.:
18	Upper limbs_Fat mass %	NU-AGE DXA T0	Upper_limbs		float	Upper limbs fat mass pe	Not regulated	[14.03,
19	Upper limbs_Regional fat mass %	NU-AGE DXA T0	Upper_limbs		float	Upper limbs regional fat	Not regulated	[13.91,
20	Upper limbs_BMD	NU-AGE DXA T0	Upper_limbs		float	Upper limbs bone miner	Not regulated	[0.55, :
21	Upper limbs_fat mass/lean mass	NU-AGE DXA T0	Upper_limbs		float		Not regulated	[0.16, :
22	Lower limbs_Weight	NU-AGE DXA T0	Lower_limbs		float	Lower limbs weight (g)	Not regulated	[16722
23	Lower limbs_Fat mass	NU-AGE DXA T0	Lower_limbs		float	Lower limbs fat mass (g)	Not regulated	[2852.:
24	Lower limbs_Lean mass	NU-AGE DXA T0	Lower_limbs		float	Lower limbs non-bone le	Not regulated	[10034
25	Lower limbs_BMC	NU-AGE DXA T0	Lower_limbs		float	Lower limbs bone miner	Not regulated	[551.9,
26	Lower limbs_Soft tissue	NU-AGE DXA T0	Lower_limbs		float	Lower limbs soft tissue (Not regulated	[16110
27	Lower limbs_Fat mass %	NU-AGE DXA T0	Lower_limbs		float	Lower limbs fat mass pe	Not regulated	[14.14,
28	Lower limbs_Regional fat mass %	NU-AGE DXA T0	Lower_limbs		float	Lower limbs regional fat	Not regulated	[13.73,
29	Lower limbs_BMD	NU-AGE DXA T0	Lower_limbs		float	Lower limbs bone miner	Not regulated	[0.9, 2.
30	Lower limbs_fat mass/lean mass	NU-AGE DXA T0	Lower_limbs		float		Not regulated	[0.16, :
31	Trunk_Weight	NU-AGE DXA T0	Trunk		float	Trunk weight (g)	Not regulated	[25230

Figure 10: NU-AGE variables documentation file. Each line represents a variable and provides information regarding the source dataset, logical grouping with other variables (if any), variable type, and so on.

and 95th percentile of its distribution and used them as expected ranges in simple InRange checks in Validator, for both T0 and T1 data. This approach may seem coarse at first glance, but it has proven to be a good compromise between false positives and false negatives rates, allowing to solve otherwise difficult to detect errors. The downside is that it's data dependent: you can't transfer the ranges obtained in NU-AGE to other studies, unless the same population and instruments are used. Independently from the success this approach had, there's a rule of the thumb that every person should keep in mind: when complex data are at play, nothing is better than a trained person in checking them. He or she can be helped by automatic systems such as Validator, but a final glance by a human eye is always needed.

4.2.7 Crunching everything - document to rule them all

Databases full of precious and extremely expensive data can be transformed into junk if nobody can understand how they're organized. It is probably the most boring and overlooked task in scientific research, but documenting the data is a key step to allow their usage in the long term. It happened to all of us: a thing you perfectly remembered until the minute before, then you can't recall even the tiniest piece of it. That's why the meaning of variables, the keys used to encode them and, in a broader sense, all the *metadata*, data about the data, that may be put at good use in the future must be clearly written down and made available to all interested parties.

In NU-AGE we followed this "creed" from the very first moment. As soon as data have begun to flow inside the database, questionnaires structure and checks were combined leveraging Validator parsers and relevant information stored in a Excel file, visible in Figure 10, ready to be exploited by data analysts. Each line in the file contains a variable – the order is given by its appearance in the QuestionnaireSet – and details from which dataset it comes from, any logical grouping with other variables, the type, and so on.

But this is just half of the story. What it is also important to know is how anomalous values should be treated: there's a reason for them, and thus they should be kept in the dataset, or we must consider them an outlier to be discarded? Here the database we're building with Validator will come into play. Once the validation will be deemed complete, authorized VRs along with their reason will be extracted and made available in a yet to define format to NU-AGE partners. This will empower a rational decision on

removing subjects from the cohort, improving the usual custom of throwing away everything that doesn't fit the initial idea of the data.

4.3 NU-AGE - THE @BOLOGNA DATABASE

At the beginning of 2014, when the data started to grow in the database and Validator reached a stable enough development point, we realized that having a copy of the NU-AGE database hosted in Bologna would have facilitated the validation work. Nonetheless, deploying an infrastructure capable of managing the data and ensuring their security against external threats, while keeping them available for the authorized staff, is a complex task that embraces many different domains.

4.3.1 Why?

The "why" behind this decision must be looked for in the following points:

- reduce the overhead. At that time we were pre-processing every update of the data twice: to validate them with Validator, and to send them to the dbnp (see 3.3.4).
- reduce the workload on the TNO. Every iteration of the data was processed and sent to be loaded in the database, that was instead supposed to store only their final revision.
- tracking. Dbnp does not support data versioning, mainly because it's focused on storing clean data only. Nonetheless, having older version of the information is important while you improve them, at least for comparison.
- reduce time-to-database. The time a single bit of information takes between being entered in the data entry interface and reaching the database ready to be queried is wasted time and should be reduced as much as possible.

All these point could have received an answer with a database in Bologna: single pre-processing, versioning, shorter time to database, and less work on top of TNO's shoulders.

4.3.2 Which database?

Database is a catch-all word that means a lot of things. Microsoft Access is a database, but is not exactly what you imagine when talking about databases. PostgreSQL is a RDBMS that could do the job, but it's just the shelf where to store information, but what about regulating access to this shelf, let alone support an *easy and friendly access*?

What we were looking for in reality was a whole platform for data exchange, not just a database, but a front-end to be made available online for people to access the data and administrator to monitor this access. We searched quite a lot to find a suitable candidate, and we eventually found LabKey (see 3.3.5). We quickly deployed it on a virtual machine for testing purposes, and soon realized that it was exactly what we were looking for.

4.3.3 Database deployment

Deploy means having a server to deploy into. However, we didn't have a server available, and we also lacked the skills needed to *harden* (a computer science term to say "make more secure") it. We thus started a profitable collaboration with the University IT Department, that provided us with a

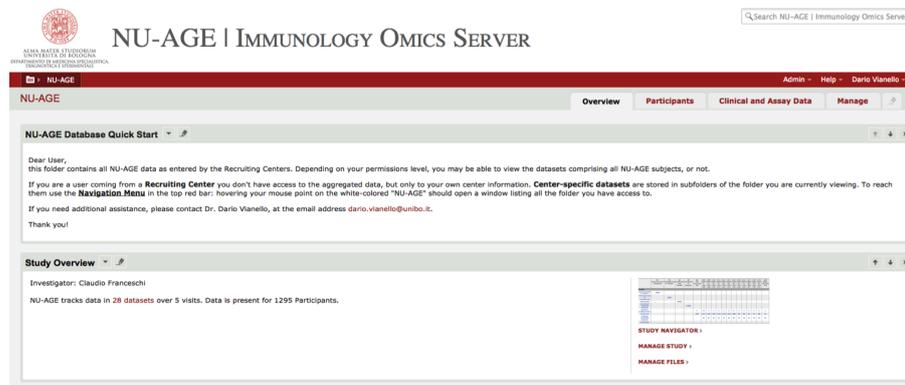


Figure 11: NU-AGE main page in LabKey platform. From here users with enough privileges may access the data subdivided by center, the datasets containing all the subjects, as well as the study admin panels.

virtual machine in their virtualized infrastructure in no time, and helped us in ensuring its integrity. These efforts eventually ended up in a 10 page long document analyzing all the weak points to be addressed to increase and maintain in time the security level of the resources, reaching a level assimilable to what it is normally requested by the ISO 27001 certification (<http://www.27000.org/>), the de-facto standard of data protection and integrity. We cannot provide here the details about all the steps undertaken to secure the server for obvious reasons. However, every component used by the interface, PostgreSQL, Tomcat and the operating system itself were analyzed and patches applied to reduce the risk of external threats.

LabKey, through Tomcat, publishes a web interface for users to access the platform resource via any common browser. Appropriate ports were opened to provide encrypted (https-only traffic) access to authorized personnel, and strict policies applied for password creation and expiry.

4.3.4 Database description

The access to the database, if correctly configured, is extremely easy. Once connected to the correct url, you're presented with the platform home page, from which the area dedicated to NU-AGE, visible in Figure 11, can be accessed. Through a menu, it is possible to select the subfolders hosting the data separated by center (see Figure 12 on the next page) and the documentation, which is made by a set of wiki-style pages containing all the information needed to navigate the data (see Figure 13 on page 54).

Lastly, If you're granted access to, you can access the location where the whole study is stored. LabKey provides numerous facilities to help dealing with longitudinal studies, ranging from monitoring datasets filling, to assays management, automatic subdivision of enrolled subjects in sub-cohorts (i.e. cases / controls) depending on the entered data, values plotting and advanced querying. In Figure 14 on page 55 you can see the so-called "Study Navigator", a page detailing the number of records the system contains for each dataset at each timepoint.

4.4 NU-AGE SYSTEMS BIOLOGY

One of the techniques that immediately pops out of a researcher's mind reading the NU-AGE project is, probably, systems biology. With the plethora of gathered data, some of which proxying the functionality status in fundamental physical and cognitive systems, building models on the most disparate interactions may be theoretically possible. But, are we there already?

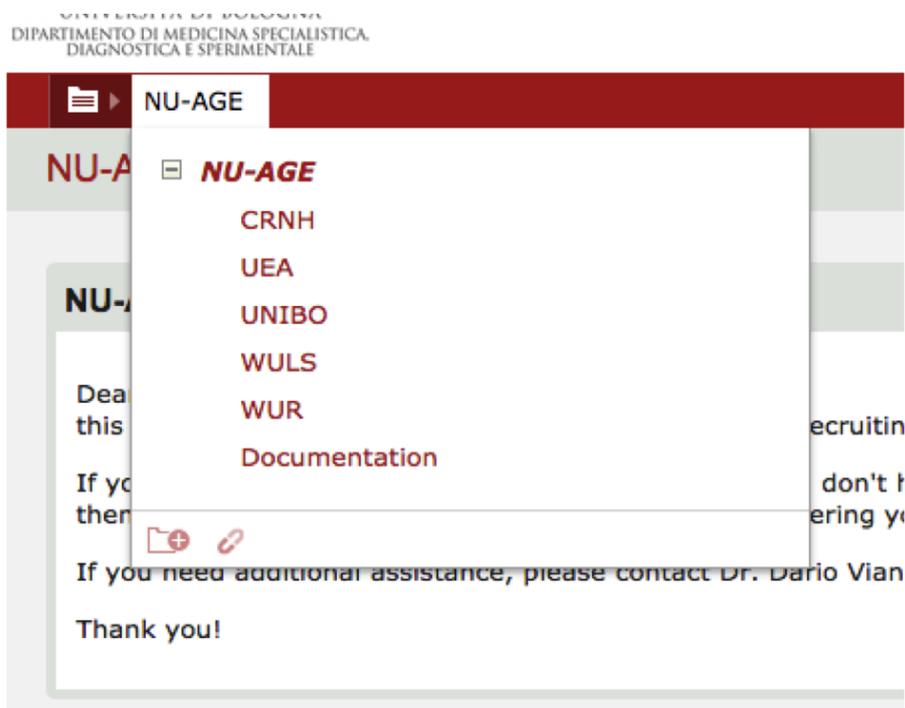


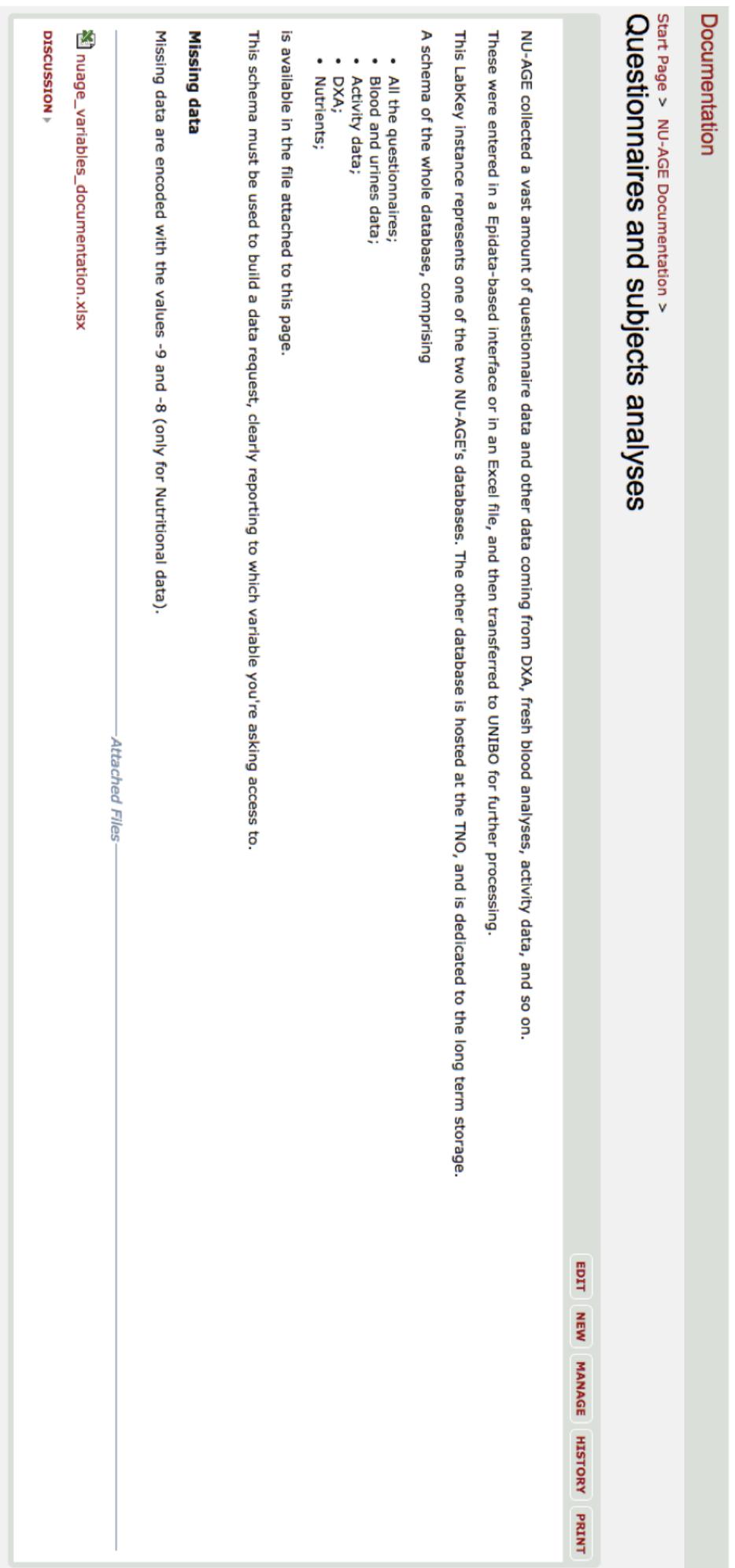
Figure 12: NU-AGE menu in LabKey platform. From here users with enough privileges may access pages regarding specific centers and NU-AGE documentation.

4.4.1 Where to start the journey from

The 1-year diet intervention is the planet around which the whole project orbits. We can't start even only speaking about systems biology in NU-AGE without understanding the diet first, in terms of:

- Diet was really followed by the subjects (*compliance*)?
- Which subjects in the diet group did not follow it?
- Controls were really good controls from a nutritional point of view?
- Controls were still good controls after 1 year, or they changed their habits?

These are difficult to answer points, but ignoring them could have severely undermined the study power and outcome. The reason lies in Figure 15 on page 56, and in assessing if it depicts the real situation or not. But let's go one step back. The ideal starting point in an intervention study is a condition where subjects that in the future will be enrolled in the diet and control branches do behave identically. That's the reason behind randomization, after all. Imagine to represent the NU-AGE diet as a point in a n -dimensional space, where the n dimensions are the nutrients dietitians take into account to evaluate NU-AGE subjects prior to the inclusion and throughout the project. Following the same approach, the diet each subject was following before being enrolled can be transformed in a point in the same n -dimensional space. In the ideal situation, subject's points should lie far away from the NU-AGE diet, possibly well clustered together. Keeping following our "ideal situation case", half of the people are randomized in the diet branch, and follows the intervention for a year. In this timeframe, their diet, thus their points in space, will hopefully move toward NU-AGE, as show in figure 16 on page 56. At the same time, controls does not alter their original nutrients intake and keep their points as they were originally. This is a very fabulous tale, but it depicts the real situation? All diet subjectd really followed the diet, and none of the control subjects changed their



The screenshot shows a web page titled "Questionnaires and subjects analyses" under the "Documentation" section. The page content includes an introduction to NU-AGE data, a list of questionnaire types, and a section on missing data. At the bottom, there is an "Attached Files" section with a link to "nuage_variables_documentation.xlsx" and a "DISCUSSION" link. On the right side of the page, there are navigation buttons for "EDIT", "NEW", "MANAGE", "HISTORY", and "PRINT".

Documentation

Start Page > NU-AGE Documentation >

Questionnaires and subjects analyses

NU-AGE collected a vast amount of questionnaire data and other data coming from DXA, fresh blood analyses, activity data, and so on. These were entered in a Epidata-based interface or in an Excel file, and then transferred to UNIBO for further processing.

This LabKey instance represents one of the two NU-AGE's databases. The other database is hosted at the TNO, and is dedicated to the long term storage.

A schema of the whole database, comprising

- All the questionnaires;
- Blood and urines data;
- Activity data;
- DXA;
- Nutrients;

is available in the file attached to this page.

This schema must be used to build a data request, clearly reporting to which variable you're asking access to.

Missing data

Missing data are encoded with the values -9 and -8 (only for Nutritional data).

Attached Files

 [nuage_variables_documentation.xlsx](#)

DISCUSSION ▶

EDIT NEW MANAGE HISTORY PRINT

Figure 13: NU-AGE docs in LabKey platform. Provided as a set of wiki-style pages, it contains all the information needed to safely navigate throughout NU-AGE data.

	All Visits	Baseline	Follow up - Month 4	Follow up - Month 8	Follow up - Month 12	T1
Assays						
Citomegalovirus	1221	1221				
Uncategorized						
Exit Questionnaire	252					252
Additional Indexes - T0	1271	1271				
Admission 1	1284	1284				
Admission 2	1283	1283				
Blood & Urines T0	1249	1249				
DXA T0	1247	1247				
Energy Expenditure T0	1247	1247				
Nutrients T0	1230	1230				
General T0	1281	1281				
General T0 - Prescribed Medicines	980	980				
Interview T0	1264	1264				
Supplements T0	1125	1125				
Follow up M4	1033		1033			
Follow up M4 - Prescribed Medicines	237		237			
Follow up M8	895			895		
Follow up M8 - Prescribed Medicines	197			197		
Follow up M12	395				395	
Follow up M12 - Prescribed Medicines	62				62	
Blood & Urines T1	458					458
DXA T1	245					245
Energy Expenditure T1	248					248
General T1	589					589
General T1 - Prescribed Medicines	385					385
Interview T1	576					576
Supplements T1	356					356
Vitamin D	212					212

Figure 14: NU-AGE study navigator in LabKey platform. From this panel, authorized users can have a precise idea of the completion status of each dataset composing the study.

habits in response of only having heard about a "healthy diet" once? Because, if one of these assumptions doesn't hold any longer, we may lose all the power the study has or, worst, draw wrong conclusions from the data.

This was the question fiddling in our mind when we first thought about how to verify these hypotheses. However, we were quite sure that the n-dimensional space was the right direction to point at. Which is the best tool to pull down n-dimensional spaces (each one representing a variable) to two or three dimensions? PCA, of course! We did a PCA on T0 nutrients data from all the centers and results were far from being what we expected at all.

What will follow in the next paragraphs is an analysis of the first four components obtained applying PCA our data. A plot of the fraction of explained variance by each component is visible in Figure 17 on page 58, from which you can see that the first four components account for roughly the 60% of the variability. However, the rest of the figures are not exactly what readers are used to expect when talking about PCA, so I will spend a couple of lines explaining them. Let's start with Figure 18 on page 59. The plot on the left shows the loading values for the first component. Loadings are numbers defining the weight each variable has in defining the final value of a given component for each subject. Positive loading will pull the component final value to positive values, while negatives ones will have an opposite effect. How strong this pulling is is defined by the absolute value of each loading. Going back to the plot, on the Y axis we have all the variables, while their loadings are plotted on the X axis. What we can glean from this graph is that the first component seems to be pretty generic: we have a single positive loading – the percentage of energy coming from carbohydrates – while all the others are negative. Probably, the component value depends simply on the total income of food each subject eats. Let's focus now on the right part of the figure. Here you have 5 histograms showing the distribution of the values of first component divided by center (sub-plots) and sex (color). The green bar represents the point were a subject exactly following the NU-AGE diet would lie in the chart. Almost all distributions, independently by sex and center, peak around the ideal diet. The only center that is skewed to the right, thus towards positive values, is UNIBO. This observation nicely fits with the plot on the left: the only positive loading is the amount of energy coming from carbohydrates, and you know, Italians do love pasta!

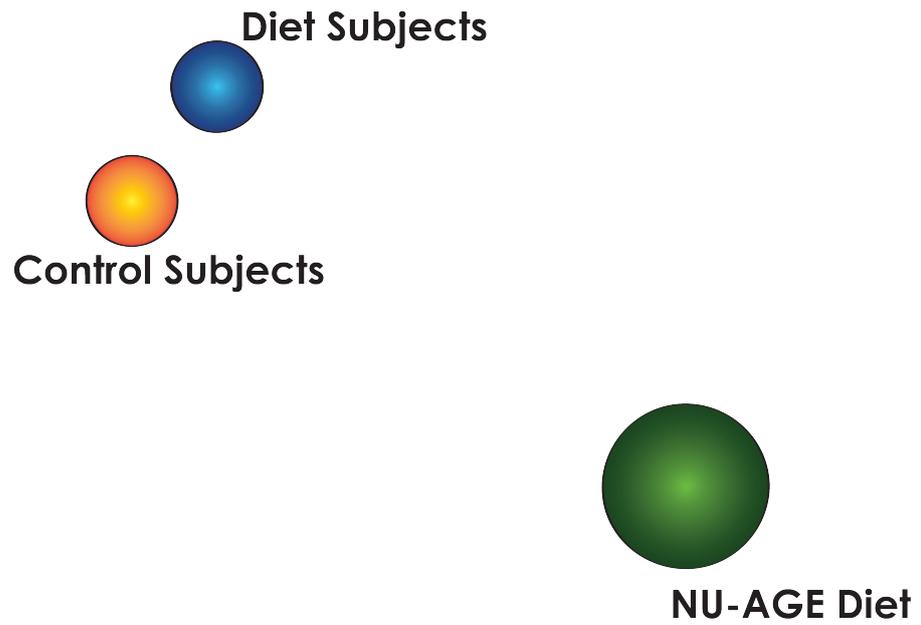


Figure 15: NU-AGE Systems Biology - The ideal enrolled subjects' situation at T0. Future control and diet subject have a similar diet which is far from NU-AGE one.

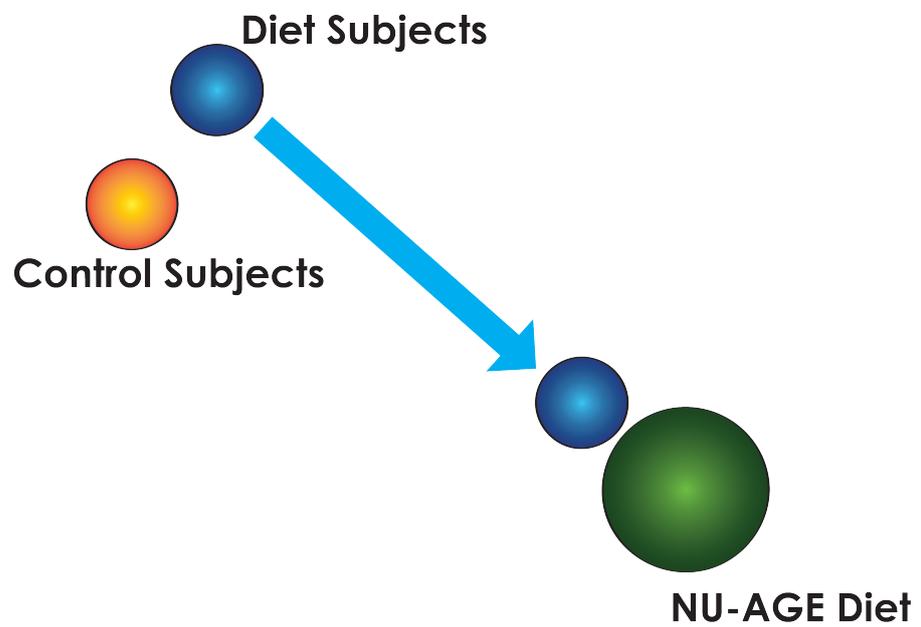


Figure 16: NU-AGE Systems Biology - The ideal enrolled subjects' situation at T1. After the one-year intervention, subjects following the diet branch move their own diet towards NU-AGE one, while controls keep their original positioning.

The second component, visible in Figure 19 on page 60, shows instead a different behavior. We have three important negative loadings, all related to fats (fat, saturated fats, MUFA-PUFA), against a single positive loading, which is again the percentage of energy obtained from carbohydrates. This time the component tries to discriminate between sugars versus fats consumption. Nonetheless, distributions are still pretty similar in all the centers, each peaked around the idea diet.

The third component, visible in Figure 20 on page 61, eventually behaves differently from the previous ones. This time it's protein income (fr-prot-energy variable) against all the other nutrients plus the difference between needed and introduced energy (last variable in the chart, fr-diff-energy). In this case, the charts on the left start to show some differences.

Fourth component, Figure 21 on page 62. Things are getting fuzzier here, and explaining the rationale the PCA is trying to extract is increasingly difficult. We have several nutrients with positive loadings against a single, very important in absolute value, loading: alcohol (fr-alc). We named this component "health conscious versus non caring", but probably a better name would have been "No alcohol vs alcohol". Yet again, most of the centers behave similarly, with the exclusion of WULS (Poland) that seems to have enrolled subjects with a lower preference for alcoholics. Still, distributions peak are around the ideal value.

We indeed plotted up to the 6th component, but after the 4th it was really hard to understand their meaning. In any case, given that these first four plots do account for more than the 60% of the variability, they already give a clear picture of the trends in the data. We now know that NU-AGE subjects lie around NU-AGE diet at T0, but we still miss information on how far and spread they are.

An initial answer to this question could be given by plotting in a three dimensional scatterplot the first three PCA components, that should give us an idea of the relative positioning of each subject with respect to NU-AGE. The result is shown in Figure 22 on page 63, which contains four scatterplots of the same data, rotated at four different angles. The first take home message this representation gives is that subjects form a cloud that, at one side, contains the "ideal diet" itself (represented by axes crossing point). The second message is that the distance between each point and the diet greatly varies, and we thus can't consider all the subjects in the same way. Third message: the cloud is formed by an almost identical number of diet and controls, are they clustered by branch or not? We should have a way of evaluating these distances to better weight the data obtained within the project.

These initial analyses propelled a inter-center effort, mainly driven by UNIBO, to define a compliance index for the NU-AGE diet. Compliance indexes already exist, but are pivoted on guidelines that are different from what defined by the consortia. We thus decided to implement a new index specifically based on the fortified diet that is the key point of the whole project. From a theoretical point of view, this is not complex: dietitians use the same nutrients intakes we used in our PCA to evaluate the progress subjects do towards established daily incomes. A simple euclidean distance between optimal values and each subjects intakes could give us a coarse index to start with. However, there are still open questions that must be addressed:

- not all nutrients have the same importance, and should weight differently in the index;
- optimal incomes are not symmetrical, i.e. it is far better to be 10 units below the threshold value for sodium than 10 units above it;
- People do cheat normally. When talking about diet and health, they cheat even more.

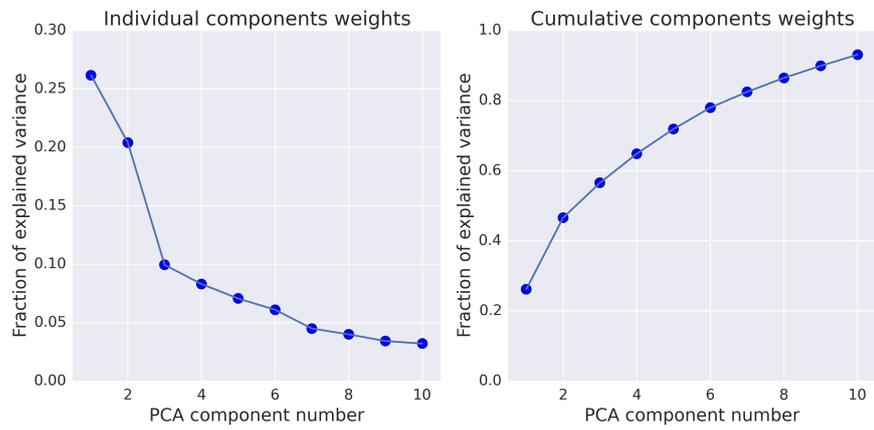


Figure 17: PCA on Nutrients Data - Fraction of explained variance by each PCA component. The first 4 components explain ~60% of the variance.

I'm sorry to say that, at the time of writing of this thesis, the consortia is still discussing the best approach to solve these issues. The last point in the list can be somewhat solved asking dietitians their gut feeling about how trustworthy the data reported by each subject were, but this is far from being a "scientific" approach. On the contrary, to address the other two issues we started a very long work of revision of the literature to come up with a consensus about which nutrients, among the monitored ones, should our index be more focused upon, and how to deal with the lack of symmetry in some of them.

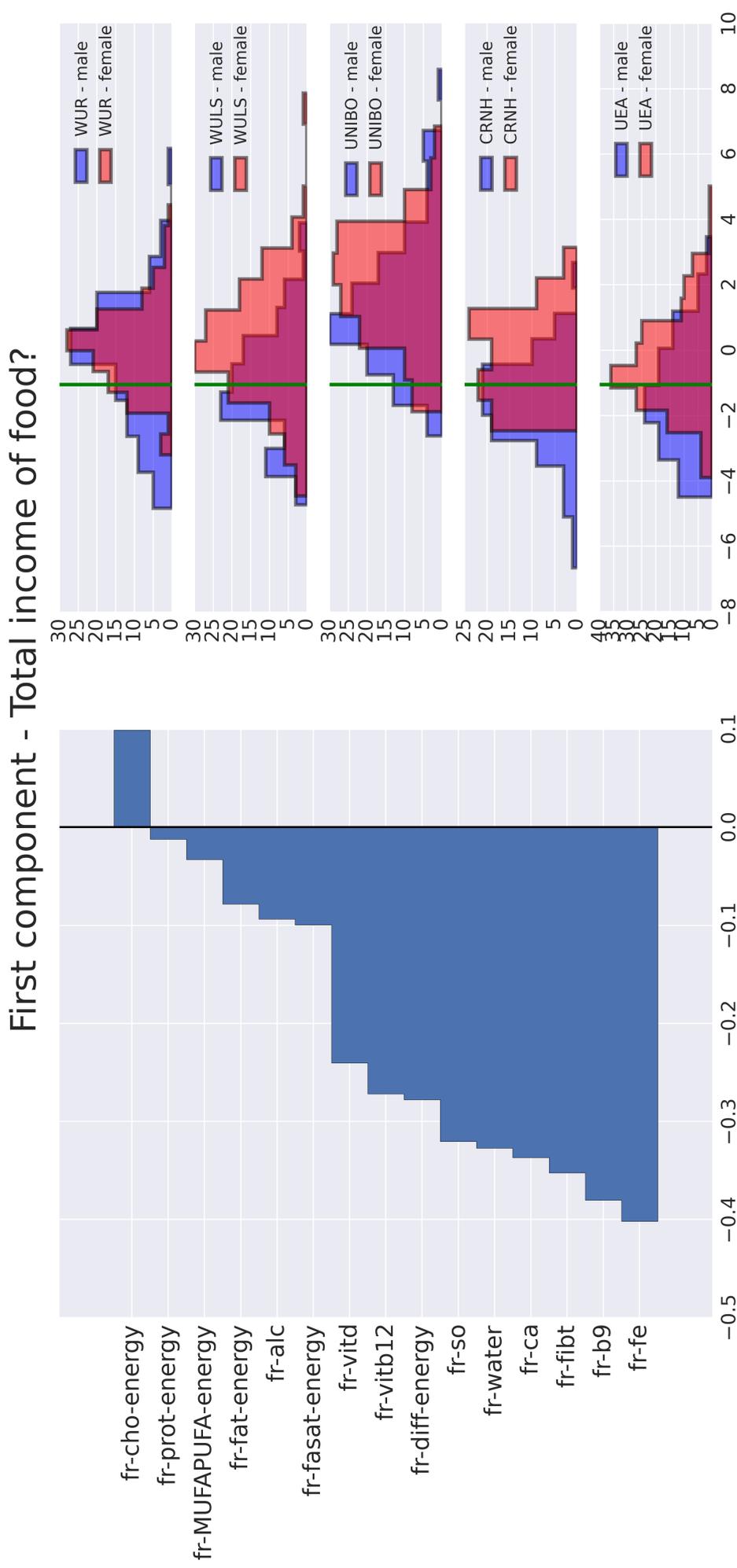


Figure 18: PCA on Nutrients Data - First component. On the left the loadings of each variable in the component are displayed, while the plot on the right shows the component distributions subdivided by center and sex. Acronyms: WUR: Wageningen University; UNIBO: University of Bologna; CRNH: Centre de Recherche en Nutrition Humaine - Clermont-Ferrand; UEA: University of East Anglia.

Second component - Sugars versus Fats?

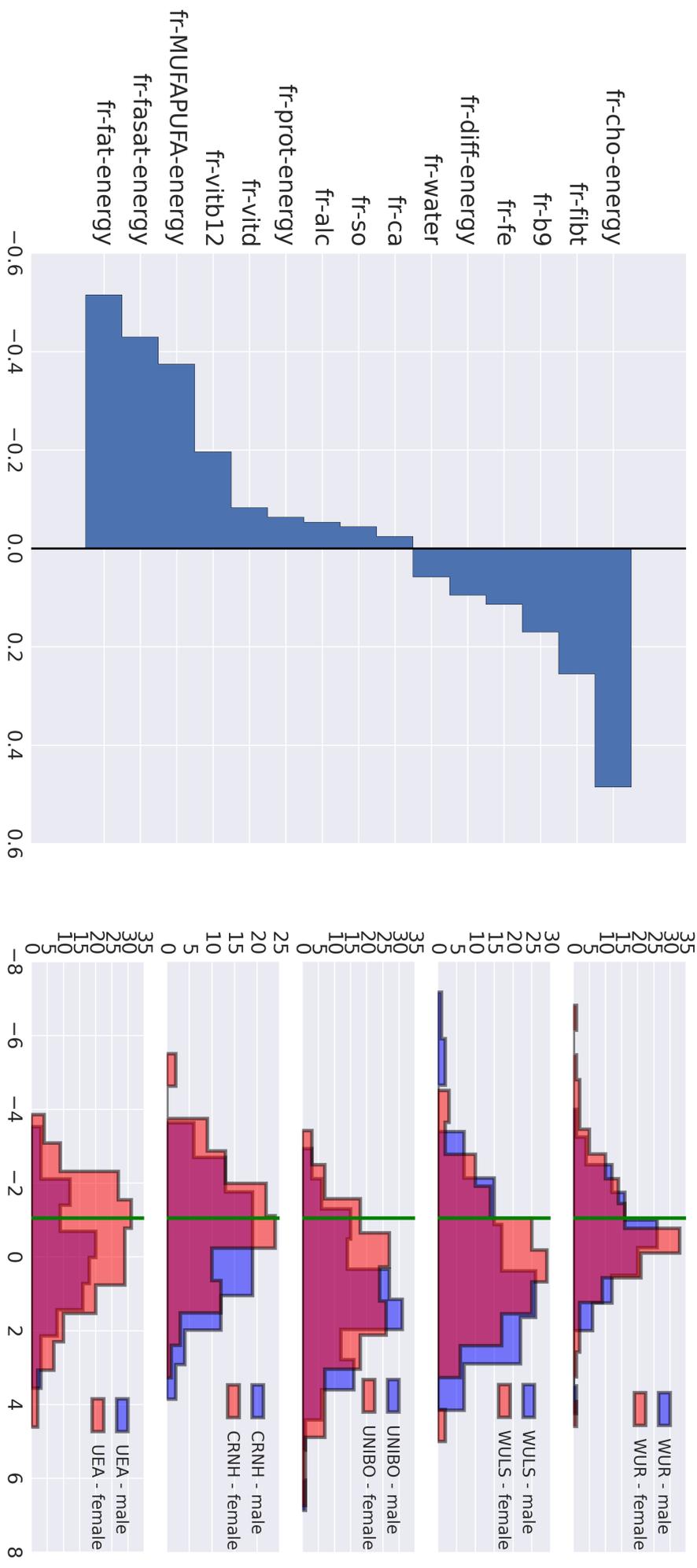


Figure 19: PCA on Nutrients Data - Second component. On the left the loadings of each variable in the component are displayed, while the plot on the right shows the component distributions subdivided by center and sex. Acronyms: WUR: Wageningen University; WULS: Warsaw University of Life Sciences; UNIBO: University of Bologna; CRNH: Centre de Recherche en Nutrition Humaine - Clermont-Ferrand; UEA: University of East Anglia.

Third component - Proteins versus other nutrients?

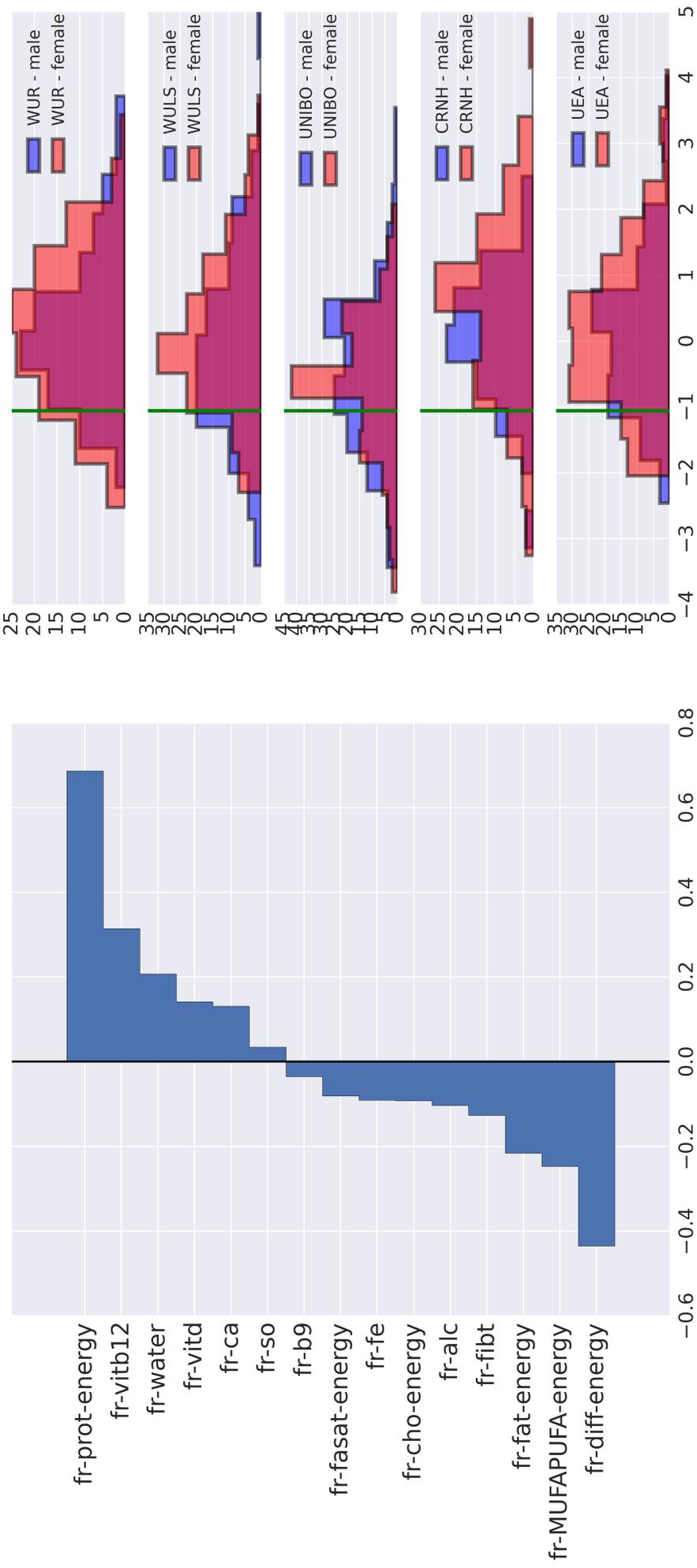
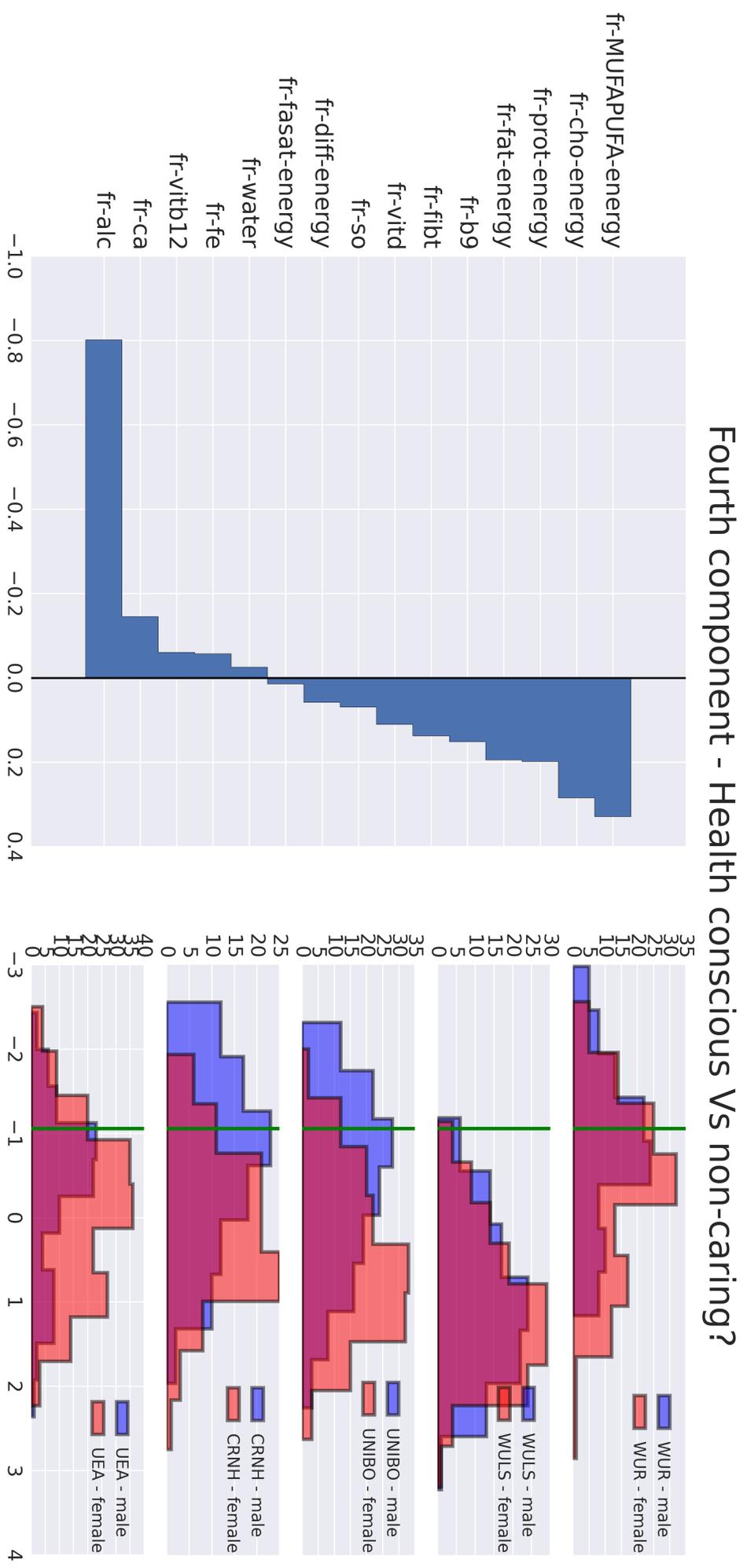


Figure 20: PCA on Nutrients Data - Third component. On the left the loadings of each variable in the component are displayed, while the plot on the right shows the component distributions subdivided by center and sex. Acronyms: WUR: Wageningen University; WULS: Warsaw University of Life Sciences; UNIBO: University of Bologna; CRHN: Centre de Recherche en Nutrition Humaine - Clermont-Ferrand; UEA: University of East Anglia.

Figure 21: PCA on Nutrients Data - Fourth component. On the left the loadings of each variable in the component are displayed, while the plot on the right shows the component distributions subdivided by center and sex. Acronyms: WUR: Wageningen University; WULS: Warsaw University of Life Sciences; UNIBO: University of Bologna; CRHN: Centre de Recherche en Nutrition Humaine - Clermont-Ferrand; UEA: University of East Anglia.



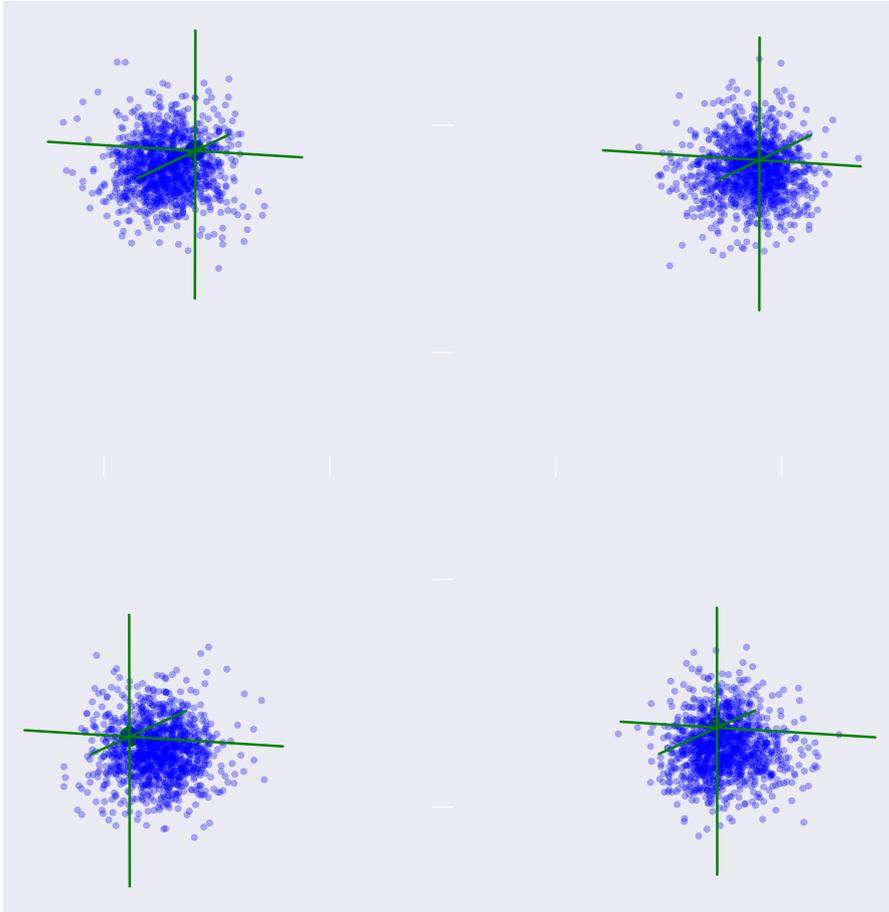


Figure 22: The NU-AGE Cloud. This plot shows a series of 3D scatterplots based on the first three principal components obtained in the described PCA approach, where each plot position is given by its values for the first 3 principal components. The point where the three green axes intersect themselves represents the point where a subject exactly following the NU-AGE diet would lie. The four subplots are different rotations of the same plot.

4.5 GEHA AND THE EPISTASIS

GEHA main aim was to understand the genetics components, if any, of longevity. In particular, our lab was in charge of part of the mitochondrial DNA sequencing and analysis.

4.5.1 Obtaining the sequences

In total, GEHA completely sequenced ~1200 mtDNA samples, 200 of which were done in Bologna, while the rest was sent to the BGI (China) to be processed. However, analyzing the FASTA files obtained by the two laboratories, it was clear to us that samples sequenced in China had a higher number of mutations, most of them recurring at fixed positions, supporting the idea of problems in the sequencing and/or assembling steps. Since raw data from the sequencer were available, we decided to reassemble them from scratch.

The first step was to build a pipeline capable of reading and processing the raw data. From private communications with the BGI, we knew that their pipeline was based on Phred and Phrap, well known command line softwares in this field. To better understand the issues BGI incurred in and possibly solve them we decided to adopt the very same tools.

Phred and Phrap work together to read data coming from the sequencer, the *chromatograms*, call the bases and assemble the contigs. In general, the pipeline successfully assembled most of the sequences, and only small refinements in well-known troublesome positions (i.e. C-stretches) were needed. For a small subset of them, however, the automatic approach proven to be insufficient.

Fixing the errors

Sequences failing automatic assembly mainly came from sub-optimal samples, characterized by quality values lower than the average. In the assembling phase, parameters controlling how quality values are assigned and the minimal thresholds for a read to be included when building contigs have obviously a critical importance. In our initial setup, reads with length below expected due to low quality issues were automatically discarded. This approach represents the golden standard in an ideal situation, but can severely hamper the final result when, for various reason, the sequencing step doesn't perform optimally. Mitochondrial DNA sequencing is well known to be a tricky challenge due to its peculiarities, such as the very high number of homopolymers. To accommodate these lower quality sequences, discarded reads were manually analyzed and pipeline parameters tweaked to allow a less stringent filtering, ultimately leading to complete sequences. Particular care was then used in manually revising the mutations harbored by these samples, since more relaxed parameters in the assembly step can cause the number of errors to surge abruptly. In edge cases, sequences were manually checked base by base by a trained researcher.

How to evaluate the quality of the final product, the sequence? Obtained sequences were continuously aligned with ClustalW (then ClustalOmega) against the mitochondrial reference sequence (rCRS, ANDERSON et al. 1981; ANDREWS et al. 1999. It wasn't possible to adopt RSRS since the whole GEHA project was based on rCRS) to monitor their quality, thus providing a stable guide to tweak the pipeline parameters towards an improvement of the reliability of the process, since detecting much more variability than expected – and described in literature – would immediately switch a red light on. We can call this process a *parameter optimization process*, where the measure leveraged to evaluate the fitness is the sequence quality and reliability.

Comparing the sequences obtained by UNIBO starting from raw data against the initial version allowed to detect ~700 errors, some of them recurring in many samples. Ultimately, the work done allowed to significantly increase the quality of the data, lowering the noise level that could have severely impacted on the statistical analyses.

4.5.2 Analyzing them

FASTA sequences are a step toward data analysis, but they are still a sort of "raw data", albeit reaching them means a significative amount of work. How to proceed further?

A routine analysis that also doubles as quality check is haplogroup assignment (SALAS et al. 2005). In 2013, we published a tool called HaploFind (VIANELLO et al. 2013) that automatically does this task, accepting as input mtDNA sequences in FASTA format and giving back reliable assignments based on the statistical analysis of the sequences available in GenBank, using PhyloTree (VAN OVEN et al. 2009) as a scaffold. Yet, the first step for assigning an haplogroup is to know which mutations the sequence hosts to map them on the tree. HaploFind allows to export, along with the assigned haplogroup, SNP discovery results in various formats ready to be analyzed.

For our analysis we started from the Excel output HaploFind provides, which is basically a long list of rows each one reporting the sequence id of a sample in the first column and its mutations in the second. Other information are also available, but are out of the scope of this thesis. We melted the information contained in the export, ending up with a matrix in which every row represented a subject and every column a mutated position. At the crossing between a subject and a position the nucleotide at that given position in that given sequence is reported.

Where to go from here? In 2013 we published a paper (RAULE et al. 2014) describing the result of a initial set of analyses on these samples, where we failed to identify the association of single SNPs to the longevity phenotype, but we indeed detected significant signals when taking into account complexes as a whole, i.e. OXPPOS complexes. What were we seeing there? One possibility was that these mutations were interacting with each other sprouting in the phenotype of interest. In other words, epistasis may have been at play. For quantitative phenotypes the epistatic effect was defined by Fisher R.A. back in 1918 as a *deviation from additivity in the contribution different loci have on the phenotype*. For qualitative phenotypes, epistasis may be defined as the phenotypic effect two loci have only if both are present.

Epistasis can be detected with different approaches, as thoroughly described in literature (CORDELL 2002; EMILY 2012; MOORE 2015; MOORE et al. 2014; RITCHIE 2015). Nonetheless, we decided to apply new approaches to the problem, starting with a method based on entropy analysis.

Detecting epistasis with entropy

We're not speaking of entropy in chemistry terms here, but in terms of information theory (SHANNON 2001), where it is a measurement of the amount of information a signal carries. In our case, the concept can be rephrased as the amount of information a mutation (or a combination of) contains regarding the fact of being long-lived or not. This value, for the scope of our analysis, can assume values between 0 and 1, since we can only choose between two states (long-lived/control).

Figure 23 on page 67 packs in a single plot all the results we had from this technique. On the Y axis there are the ~1200 GEHA sequences, while the X axis hosts the mutated positions, ordered by descending value of epistasis from left to right, with the blue shadow graphically representing this trend. The crossing between a sequence and a mutation is white colored if the sequence harbors the mutation, black otherwise. Red arcs connect muta-

tions that, as a couple, have an entropy level high enough to enter in the top 500 informative couples, where the opacity of the line is in function of the entropy value. What can we get from this figure? Several interesting things, to be honest:

1. Mutations present in almost all the samples, denoted by white bands in the graph, don't encode any information, since controls and long-lived are equally present in the dataset;
2. We have many short arcs at the left side, ending up in a red glob. There's nothing interesting in these: we're on the left side of the graph, where the entropy level of each mutation *alone* is already high, no surprise that combining two of them we have a strong couple;
3. Long arcs encompassing all the plot, however, begin to be interesting: a strong mutation together with a weak one enters in the top 500 couples;
4. Eventually, the really interesting arcs are those that start and end in weak mutations, which together reach the top 500. Something must be going on there!

With this preliminary analysis we obtained evidences supporting the fact that some deviation from additivity do exist in the longevity phenotype, and we should carefully look into it. This entropy-based approach, albeit promising, do had some severe limitations:

- due to the formalization of the algorithm, it is really difficult to scale up in the number of simultaneously analyzed SNPs (to be read also as: increase the length of the mutations signature) both from a formal and computational point of view. The reader should keep in mind that epistasis analysis is indeed a *NP-hard* problem, a problem that has no solution other than testing all possible cases to choose the best among them.
- it is not possible to compute a significance for entropy values.
- it is not exactly answering our question.

Regarding the last point, this approach guarantees to find the most *informative* couple, while our aim was to find the *probability of a subject being long-lived given a signature of interacting mutations*.

Bayesian approaches for epistasis detection

We can formally rewrite the sentence above, *probability of a subject being long-lived given a signature of interacting mutations*, as

$$P(C|E) \tag{1}$$

where E is the set of mutations in epistasis. Exploiting Bayes' theorem, we can rewrite the formalization as

$$P(C|E) = \frac{P(E|C) * P(C)}{P(E)} \tag{2}$$

where $P(E)$ can be then further divided in its components, thus obtaining

$$P(C|E) = \frac{P(E|C) * P(C)}{P(E|C) * P(C) + P(E|N) * P(N)} \tag{3}$$

where $P(N)$ represents the probability of not being long-lived. This probability should in theory be estimated from GEHA controls, a part of which may, at least from a genetic point of view, be fully fledged future centenarians, thus causing a bias in our evaluation. To reliably estimate the fraction

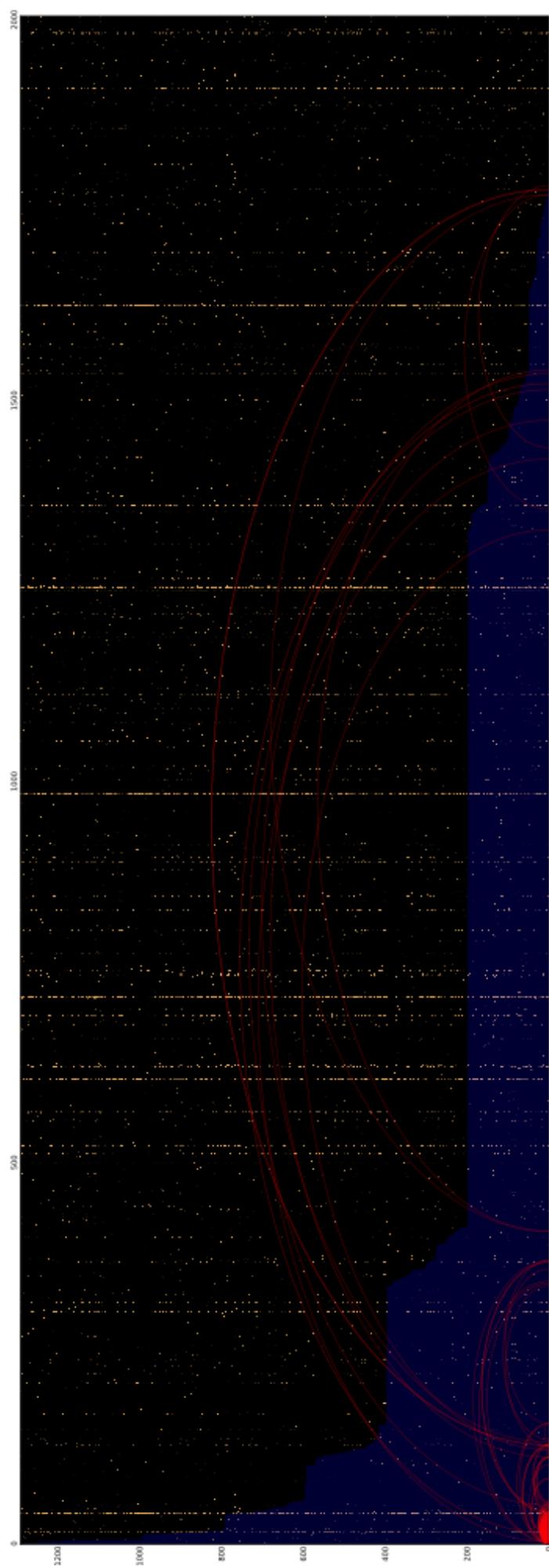


Figure 23: Epistasis analyzed with an entropy-based approach. On the Y axis the ~1200 GEHA sequences displayed one after the other, while the X axis is made of all the mutated positions, order by descending value of epistasis from left to right, as denoted by the blue graphically representing this trend. If a sequence harbors a given mutation, the corresponding crossing point is white-colored, black otherwise. Arcs connect mutations that together have an entropy level high enough to enter in the top 500 couples.

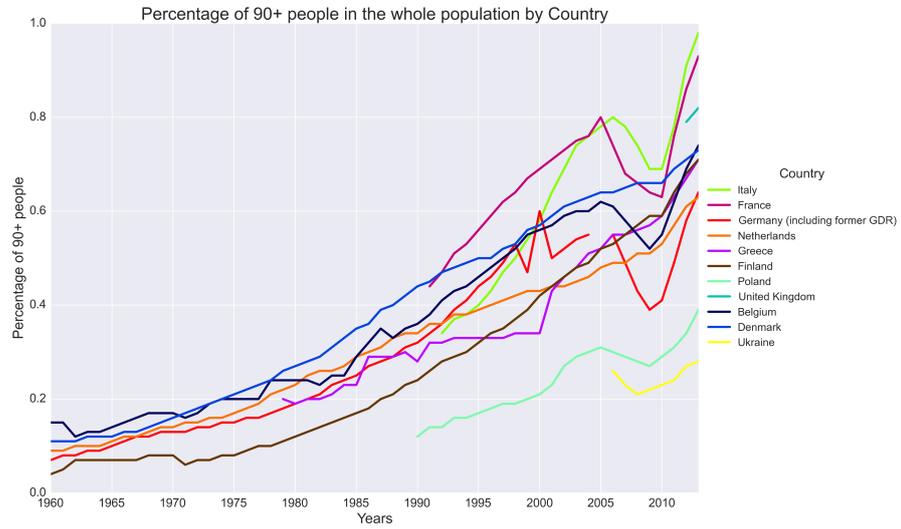


Figure 24: Trend of 90+ people percentage in the population of countries participating in the GEHA project from 1960 to 2013. Data source: Eurostat, December 2014.

of long-lived (in GEHA, "long-lived" subjects are 90+ subjects) in the general population we exploited the demographic data freely provided by Eurostat. As expected, this fraction was very low, but greatly varied among different countries and time periods, as visible in Figure 24. This variance must be taken into account, since it could cause biases when analyzing projects such as GEHA, which collected samples from different countries.

To overcome the issue given by the variability of the fraction of centenarians, we switched our formalization from probability to odds (the two are exactly the same, being $O = \frac{p}{1-p}$) obtaining:

$$O(C : N|E) = \wedge(C : N|E) * O(C : N) \quad (4)$$

where

$$\wedge(C : N|E) = \frac{P(E|C)}{P(E|N)} \quad (5)$$

With this formalization we can separate our knowledge about the general population (term in O) by the information obtained through our observation (term in \wedge), which is our relative risk. For GEHA, since obtaining a solid estimation of the parameter O (also called *prior* in Bayesian statistics) applicable to the whole dataset proved to be impossible given the variability previously described, we decided to ignore it. The price to pay for this is easy to tell: our estimate will only be based on the *relative risk* and will thus be valid only for the GEHA cohort. Nonetheless, this can also be seen as a strong point: anyone knowing the intrinsic risk of his or her population can readily obtain the absolute risk for the population simply multiplying it by the relative risk we estimated.

Last but not least, we wanted a measure to gauge how strong our claims were. Unfortunately, the approach just described doesn't allow to compute p-values for the obtained statistics. We can instead compute the *Bayes factor*, which is formally defined as

$$K = \frac{P(D|H_1)}{P(D|H_0)} \quad (6)$$

and represents the ratio between the probability of the data given the alternative hypothesis divided by the probability of the data given the null hypothesis, or, in our context, between having and not having an epistatic effect between mutations. Higher the Bayes factor value, higher the probability of our model: a bayes factor of 3 means that the model with epistasis is three times more probable that the model without it. Kass gave in 1996

Table 6: Threshold values for K interpretation

K	Strength of evidence
1 to 3	Not worth more than a bare mention
3 to 20	Positive
20 to 150	Strong
>150	Very strong

a scale of interpretation for K values (Kass et al. 1995), visible in Table 6, where values greater than 3 are considered "Positive". Indeed, a Bayes factor of 3 is the threshold we used to flag our results as significant.

When applying the same test multiple times to the same dataset, the probability of obtaining false positive results increases accordingly. For this reason, results must then be corrected for this "multiple testing" procedure. The most famous correction is probably the Bonferroni correction, after the name of the Italian mathematician Carlo Emilio Bonferroni, that requires to test each hypothesis at a significance level of α/n , where α is the desired significance level of the whole set of tests and n the number of performed tests. There are also more relaxed corrections, such as the Benjamini-Hochberg procedure, which aims to control the *false discovery rate* (FRD), usually preserving more true positive results in spite of a relatively small number of "accepted" false positives.

A strong point of our approach is that its results don't need to be corrected in this way, since the correction is *applied within the test itself*. In Bayesian approaches, it is possible to inject in the tests previous knowledge about the data you're going to analyze, with the so-called *priors*. For example, the O term we decided to ignore in Equation 4 on the preceding page was a prior. To avoid the necessity of multiple testing correction, we analyzed our sequences to determine the probably of having in any given position of the mtDNA the wildtype allele, the second most frequent allele, the third and the fourth and used it as a prior in our approach. We are thus telling the test in advance how many mutations it should expect, thus correcting the final result. At the same time, priors allow us to reduce overestimation biases due to small sample size.

4.5.3 Single SNP association

New methods needs to be tested, and we thus tried to identify single SNP associations in our dataset as a test bench. We analyzed two possible representations of the data:

- At aminoacidic level (AA), using non synonymous mutations;
- At nucleotide level (nt), using all types of mutations.

In both cases we had significant results in terms of Bayes Factor, Odds Ratio (a measure of association of a mutation to a given phenotype) and Risk Ratio (a measure of how the risk of occurring in a condition increases having a given mutation).

Figure 25 on the following page shows the result for the analysis at AA level, with the log of the odds ratio on the Y axis and the corresponding Bayes factors on the X axis. We have some solid evidences here, with Bayes factors as high as 14.5 and log odds ratio above 3 for mutations occurring in ATP6 and ND6, both linked to the OXPHOS chain. It has to be noted that the distribution of the point is symmetrical due to the inner working of the odds ratio: an allele (or AA, in this case) is as strong as associated to a phenotype as other alleles (or AA, in this case) occurring at the same position are associated to controls. Figure 26 on page 71 shows the same result but in terms of risk ratio. Here the symmetry in the plot is lost: a

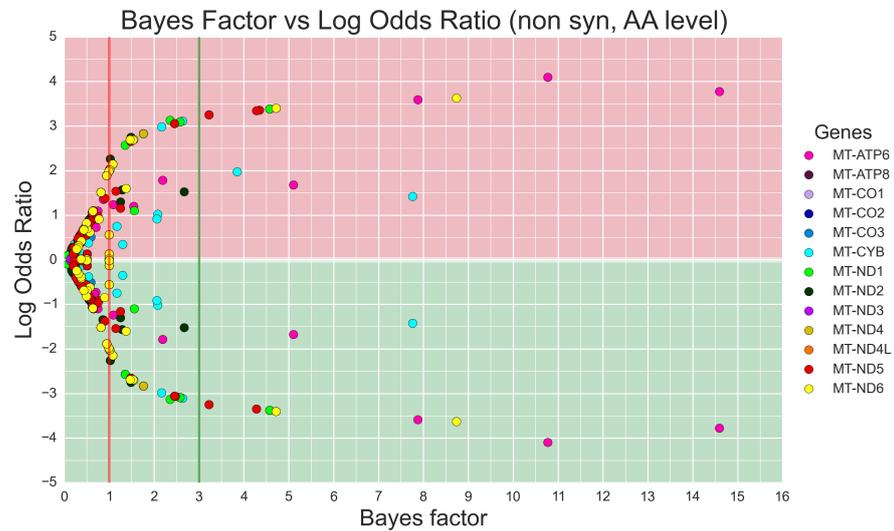


Figure 25: Bayes Factor versus Log Odds Ratio - non syn, AA level. The red and green vertical bars visualize the first and second threshold value for Bayes factor evaluation, where above 3 means positive strength of evidence. Mutations falling in the red and green part of the graph are linked to controls and long-lived, respectively.

mutation can severely increase a subject risk of being a centenarian, but the wild-type will not increase in the same way its risk of being a control. To convince yourself of this last point, think of the following example: we have a random population perfectly splitted in half between healthy and diseased. Healthy people only have the A allele at a given locus, while sick people have both A and B. If I observe B, I'm almost sure that the subject has the disease but, if the subject has A, I'm unable to clearly predict its status.

Switching to nucleotides, Figure 27 on the facing page represent the nucleotide counterpart of the first plot we had for AA, the biggest difference being that here also non coding parts of the mitochondrial DNA, such as the Dloop, are included. And indeed, along with the same good hits we had for ATP6 and ND6 in the AA-based analyses, the strongest hit was this time within the Dloop. The same trend can be seen in Figure 28 on page 72, where mutations in the Dloop showed higher risk ratios that those detected in the coding part of the molecule.

Finally, we designed a sort of "grandchild of the Manhattan Plot" commonly used in GWAS studies, using Bayes factors instead of p-values, which is visible in Figure 29 on page 72. The advantage of this plot is that allows to better understand the abundance of meaningful mutations within each gene, without being hampered by the excess of information you may experience in previous chart.

4.5.4 Multiple SNPs association - Epistasis

After testing the method to detect single SNPs associated with the longevity phenotype, we focused on studying the epistatic interactions between couples of mtDNA variants. When you evaluate single SNP associations, you only need to consider two alternative hypotheses: the null hypothesis, H_0 , that represents the absence of association, and the alternative hypothesis, H_1 , which instead represents the presence of an association. However, when you consider couples, the situation gets more complicated. In the easiest case, you want to know if two variants interact or not, limiting yourself again to two alternative hypotheses. But how these mutations interact? One is dominant on the other? The two work together to give the phenotype?

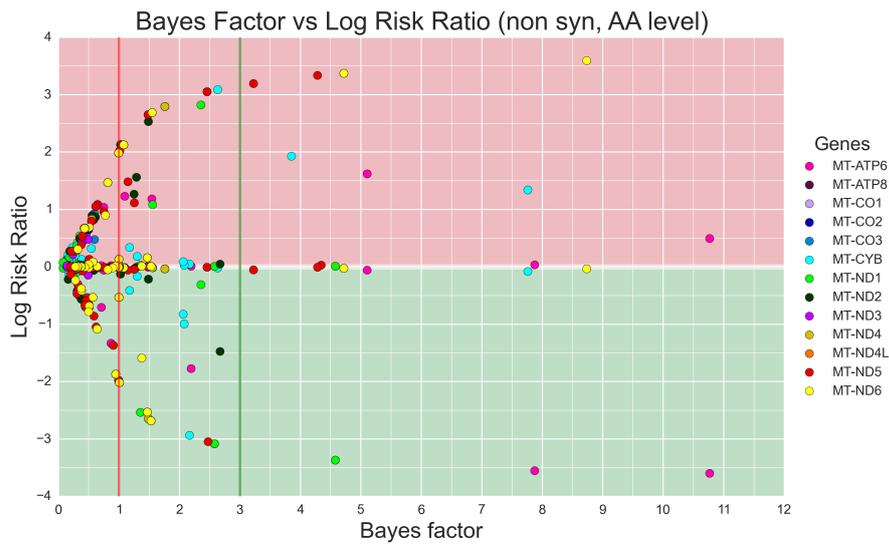


Figure 26: Bayes Factor versus Log Risk Ratio - non syn, AA level. The red and green vertical bars visualize the first and second threshold value for Bayes factor evaluation, where above 3 means positive strength of evidence. Mutations falling in the red and green part of the graph are linked to controls and long-lived, respectively.

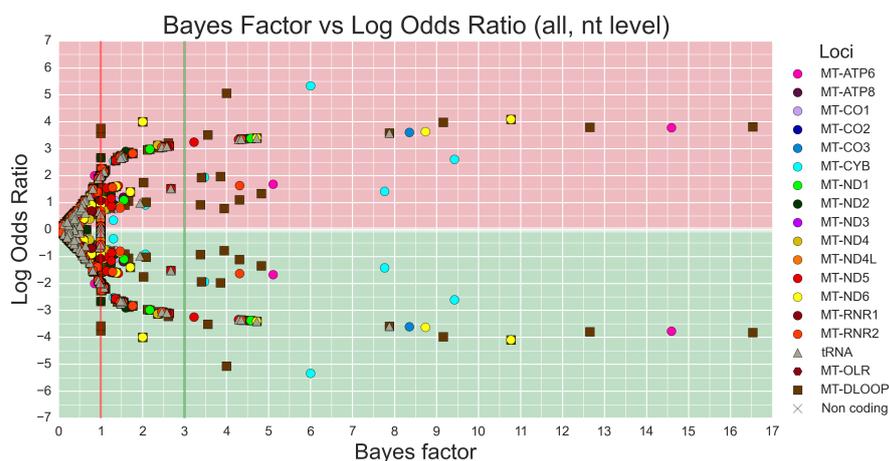


Figure 27: Bayes Factor versus Log Odds Ratio - all, nt level. The red and green vertical bars visualize the first and second threshold value for Bayes factor evaluation, where above 3 means positive strength of evidence. Mutations falling in the red and green part of the graph are linked to controls and long-lived, respectively.

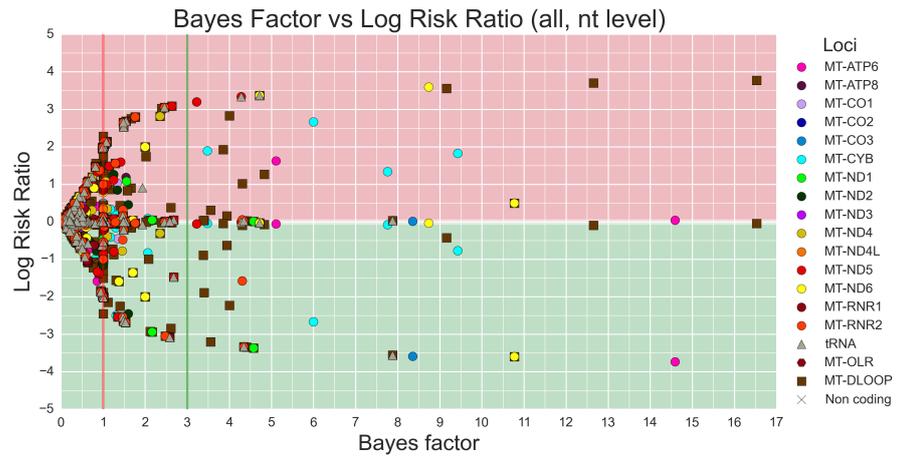


Figure 28: Bayes Factor versus Log Risk Ratio - all, nt level. The red and green vertical bars visualize the first and second threshold value for Bayes factor evaluation, where above 3 means positive strength of evidence. Mutations falling in the red and green part of the graph are linked to controls and long-lived, respectively.

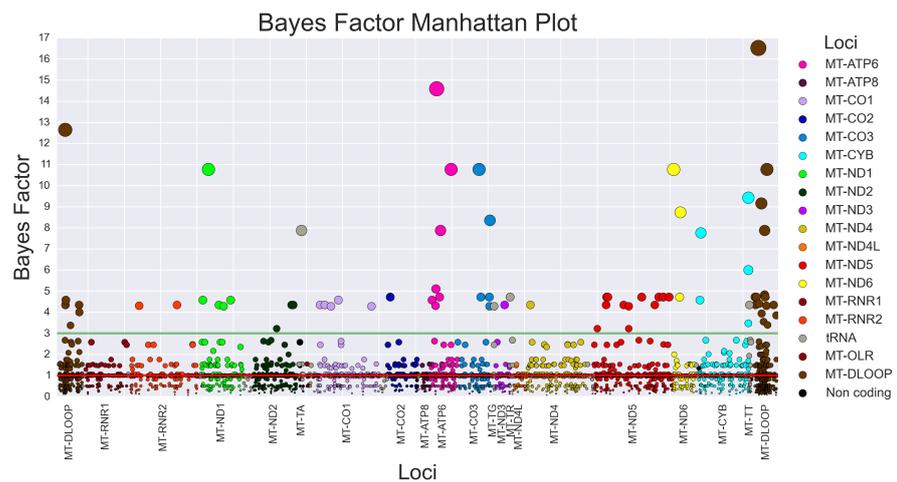


Figure 29: Manhattan plot based on Bayes Factor values - all, nt level. Mutations are colored accordingly to the genes they belong to.

After careful thinking, we built a list of 17 different possible interaction models, all which you can see in Table 7 on the next page. The four central columns describe the four different states in which the two loci can exist, where W stand for wildtype and M for mutated. The coefficients starting with an " f " are representations of the functionality level of the biological system - possibly the pathway - in each situation. In the null model, for example, we always have f_0 , since mutations are not related to the phenotype under analysis. In the fourth model, "Interactions only", the system is affected only when both the mutations are present, hence the " f_1 " in the " M_1M_2 " column. A possible explanation of each model is given in the last column of the table.

To study how variants interacts, we need to test all these models for each possible couple and obtain a probability for each of them. Nonetheless, what we really were looking for are Bayes factors, which are defined as the ratio between two models probabilities. We thus need to choose a "base" model against which all other models will be compared to obtain a set of comparable Bayes factors. Given these premises, the null model, where variants are not related to the phenotype, represents the wisest candidate for obvious reasons. After fitting all the models on a given couple, we pick the one with the highest Bayes factor, which will be the model that better fits the mutations distribution in the recruited samples.

How are we sure that the best model will have the highest Bayes Factor? This boils down to the Occam's razor principle, named after William of Ockham, an English Franciscan friar and theologian (1287 – 1347), which states that "*Plurality must never be posited without necessity*". That is, it is useless (and less powerful) to explain a phenomena with a model with more parameters than required. Let's see how this concept applies to our case with an example. Suppose we're analyzing a couple that is irrelevant for the phenotype of interest: the best match in our list of models would be the "Null model". However, also the "Complete interaction" model may fit the problem: we have four distinct parameters in there, but nothing prevents $f_0 = f_1 = f_2 = f_3$. Both models will thus fit the problem, but the first one will have a much higher likelihood, since it must estimate only a single parameter instead of four based on the same source dataset. Higher likelihood means higher probability, which in turns means higher Bayes factor. Since we grade our models based on this last parameter, we will eventually choose the "Null model" as our best guess of the phenomena we're observing, which is the correct outcome.

Result for the couple-based analysis are reported in Table 8 on page 76, based on the analysis of all the couples composed by AA variants but excluding those positions in which we observed more than two different AA. Indeed, the list of models obtaining a Bayes Factor greater than 3, the threshold we adopted in the previous analysis considering the effect single SNPs have on the phenotype, was much longer. Anyhow, due to the innate greater complexity in this analysis compared to the previous one, we decided to be more restrictive adopting a threshold of 10. The dataset we used to perform this analysis was based on AA variants, encoded in a binary form (0: wildtype, 1: mutated). This helped us in keeping low the computational burden of the process, while providing meaningful results, since mutations causing a change in the AA chain, compared to synonymous variants, are much easier to reconcile to a true biological process.

As you may see from the Table, some of the couples reach very high Bayes factors. The couple ATP6_60 – ND1_17 (where _60 and _17 indicate the mutated position in the AA chain) reaches a Bayes factor as high as 191 where, according to Kass' table for Bayes factor interpretation, a value greater than 150 suggests a "Very strong" evidence. "Strong" evidence is instead what we got for couples ATP6_60 – ND5_159, ATP6_60 – CYB_7, CYB_158 – CYB_16 and ATP6_90 – ND2_119, albeit with progressively lower Bayes factors. For comparison, the higher Bayes factors we obtained in the single variants anal-

Table 7: Epistasis - Possible interaction models

Model name	W1W2	M1W2	W1M2	M1M2	Possible explanation
Null model	fo	fo	fo	fo	Mutations are irrelevant for the phenotype
Mutation 1	fo	f1	fo	f1	Mutation 1 is relevant, Mutation 2 is irrelevant, no interaction
Mutation 2	fo	fo	f1	f1	Mutation 1 is irrelevant, Mutation 2 is relevant, no interaction
Interaction only	fo	fo	fo	f1	Each mutation by itself is irrelevant, they interact
Concordant 1	fo	f1	f1	fo	Both mutations destroy the pathway, the presence of both restores it
Concordant 2	fo	f1	f2	fo	Both mutations destroy the pathway, but in different ways. The presence of both restores it
Concordant 3	fo	f1	f1	f2	Both mutations destroy the pathway, but in different ways. The presence of both restores it, but not in the original form
Pathway	fo	f1	f1	f1	Both mutations destroy the pathway, having both does not restore it
Protective 1	fo	fo	f1	fo	Mutation 1 is protective against Mutation 2
Protective 1	fo	f1	fo	fo	Mutation 2 is protective against Mutation 1
Mutation 1, Interaction	fo	f1	fo	f2	Mutation 1 is relevant, Mutation 2 is irrelevant, they interact
Mutation 2, Interaction	fo	fo	f1	f2	Mutation 1 is irrelevant, Mutation 2 is relevant, they interact
Dominant 1	fo	f1	f2	f1	Both Mutation 1 and Mutation 2 are relevant, Mutation 1 is dominant
Dominant 2	fo	f1	f2	f2	Both Mutation 1 and Mutation 2 are relevant, Mutation 2 is dominant
Complete interaction	fo	f1	f2	f3	Mutation 1 is relevant, Mutation 2 is relevant, they interact in an unpredictable way
Independent 1	fo	f1	f2	f1*f2	The two mutations do independent damage and both are required to cause the phenotype. No interaction
Independent 2	fo	f1	f2	$1-(1-f1)*(1-f2)$	The two mutations do independent damage, the presence of only one is sufficient to cause the phenotype. No interaction

ysis were around 3 to 14. Regarding Risk Factors, we obtained values much lower than those we reached in the single variant analysis, with a maximum of 1.724 and a minimum of - 1.520. However, this may be due to the relative smallness of the dataset, since bayesian approaches tend to underestimate Risk Factors in these cases.

The obvious prosecution of the analysis plan would be to extend the testing to nuclear variants, shining light on the highly debated cross-talk between the two genomes. As detailed in Materials & Methods, we indeed have genotyping data coming from a OmniExpress chip of which we updated the annotation thanks to the Ensembl APIs. However, both the setup of the analysis and the annotation updates took more than expected, and it wasn't possible to complete the needed tests in due time to be included in this thesis. This analysis will thus be the subject of succeeding studies.

4.6 ION TORRENT DATA ANALYSIS, AN EXCURSUS OF

Next Generation sequencing represents the new frontier in biology. Most of the problems that were until some years ago difficult even to think of, are now at hand. The other side of the coin is the amount of data these methods produce and the skills needed to understand, process and analyze them, along with the computational resources required to store and elaborate the information.

Even if my thesis was mainly focused on different topics, the acquisition of a Ion Torrent PGMTM platform (Life Technologies, Thermo Fisher Scientific Inc.) made by a laboratory with which we had long-standing collaborations offered me the occasion to play with the data produced by the sequencer. There was no need to develop new algorithms: the enormous adoption these approaches are experiencing draw the attention of very skilled developers and institution on the field, resulting in many ready-to-use suites and programs. Our goal was thus to glue these resources together to create a new pipeline, possibly based on open source software, able to successfully analyze mitochondrial DNA runs, which were at that time poorly supported by the proprietary suite, ultimately leading to below average variants calling.

Being honest, we didn't started from scratch: we consciously avoided to perform the alignment step afraid of the time and resources needed to do it, preferring to use the contigs assembled by the Life Technologies suite. This fear later proved to be wrong, and we expanded our pipeline to comprise also the alignment step, accomplished with BWA (H. LI et al. 2009a), but this implementation is so young that will not be discussed here. For building our pipeline we draw fully from GATK documentation, in particular from GATK Best Practices (VAN DER AUWERA et al. 2013), and assembled each step in a seamless Python script that, given a set of bam files, process them until the very last step, variant calling (that is not the very last step, as we will see).

Which is the first step, then? **Removing duplicates.** During the sequencing process, the same DNA molecule may be sequenced many times and these duplicate reads shouldn't be used as additional evidence in calling variants. This step can be easily done with the help of Picard, a java library developed by the Broad Institute, which has the handy "MarkDuplicates" function. Most of the NGS data management and processing is exactly like this: a long list of commands to be repeated indefinitely, from here the necessity of an automatic solution. The glue that in our implementation holds everything together is made by Python, leveraging the library Plumbum (described in 3.2.2) to launch commands and interact with the system to monitor their execution.

Table 8: Results of epistasis analysis on mtDNA AA couples
Mutation counts in control, cases

Couple ^a	Mutation counts in control, cases				Winning Model	Bayes Factor	Log Risk Factors ^b			
	W ₁ W ₂	W ₁ M ₂	M ₁ W ₂	M ₁ M ₂			M ₁ W ₂	W ₁ M ₂	M ₁ M ₂	M ₁ M ₂
MT-ATP6_60; MT-ND1_17	641, 630	3, 14	0, 0	1, 3	Complete Interaction	192	1.418	0.017	0.847	
MT-ATP6_60; MT-ND5_159	639, 628	3, 9	2, 1	1, 8	Mutation 1, Interaction	109	1.003	-0.481	1.724	
MT-ATP6_60; MT-CYB_7	353, 310	2, 8	288, 320	2, 9	Mutation 1, Interaction	83	1.327	0.122	1.435	
MT-CYB_158; MT-CYB_16	621, 605	0, 0	4, 15	20, 24	Concordant 2	50	0.026	1.243	0.197	
MT-ATP6_90; MT-ND2_119	644, 642	0, 0	0, 0	1, 5	Complete Interaction	21	0.003	0.003	1.280	
MT-ND1_30; MT-ND4L_62	640, 634	4, 12	0, 0	1, 1	Complete Interaction	16	1.017	0.009	0.009	
MT-CYB_194; MT-ND1_113	624, 640	19, 6	0, 0	2, 0	Dominant 1	14	-1.102	-0.025	-1.522	
MT-ATP6_112; MT-CYB_194	624, 639	0, 0	2, 0	19, 6	Dominant 2	14	-0.024	-1.521	-1.101	
MT-CYB_7; MT-ND2_119	355, 318	289, 324	0, 0	1, 5	Mutation 2, Interaction	10	0.117	0.110	1.380	
MT-ATP6_90; MT-CYB_7	355, 318	0, 0	289, 324	1, 5	Mutation 1, Interaction	10	0.110	0.117	1.380	

^a Variants are reported in the XXX_YY form, where XXX is the mutated protein and YY the mutated position in the AA chain.

^b Risk Ratio must be computed against a base condition. In our analysis, this condition is W₁W₂, that thus doesn't appear in the Risk Ratio results.

Back to our pipeline, then. Next step down the line is **indexing**. GATK requires a index file to lower the time it spends looking for reads inside the bam file, which usually comes with the ".bai" extension. After flagging duplicates, it is mandatory to rebuild the index since the bam is changed.

The next step is probably the most troublesome for mitochondrial DNA: **indel realignment**. When the mapper aligns the reads to the reference, it can produce artifacts in the close proximity of INDELS that may appear to support the existence of several SNPs. Performing a local realignment usually solves the problem, and GATK provides the functionalities to do it. Problem is, GATK realigns indels on the *left* side, while for historical reason some of the mtDNA variants are usually reported in their *right-aligned* form. There's no solution to this: the local realignment step is not mandatory, but greatly improves the performance of the calling around and on INDELS, and must thus be done. The downside is that in this way the operator that will evaluate the called variants down the line should remember this issue and fix the files accordingly, or a custom script must be developed to apply the needed changes automatically.

Next up, **base quality recalibration**. Base qualities are values provided by sequencers that represent how confident they are in the nucleotide they have called (please note, base calling is not variant calling at all!). These values are often affected by systematic errors, and should be adjusted. GATK solves the problem using machine learning techniques to model these errors empirically and fix the quality values accordingly. Also this step has an important impact on the final outcome, and is highly recommended. This single step requires 3 to 5 different commands, still of the idea that the "manual way" is the way to go?

We're almost there: **Variant Calling**. GATK is a fairly advanced piece of software, which can take advantage of the know variability to improve the quality of newly called variants, delivering much better results as you will see in a moment. In our pipeline, mitochondrial DNA variants genotyped in the 1000Genomes project are used to help GATK in doing its job. If available, more than one variability source can be specified. GATK provides two different callers: UnifiedGenotyper and HaplotypeCaller, where the latter is a newly implemented version that fixes many of the glitches the UnifiedGenotyper suffered of, but doesn't support organisms that are not diploid, and mtDNA has a ploidy of 1. This forced us to adopt UnifiedGenotyper in our pipeline, in spite of all the improvements available in HaplotypeCaller.

As said before, there's still a step to do after calling, which is **variant filtering**. Callers are usually really generous in calling variants to achieve a higher sensitivity, and GATK is no exception to this rule. It is thus important to filter variants according to different parameters to reduce the amount of false positives. However, filtering cannot be applied to SNPs and INDELS at the same time: they must be splitted first. Again, GATK helps us providing the needed commands. Once filters are applied, you're free to merge back the variants in a single file. But what these filters are? Here's an example of "standard filters" for mtDNA SNPs:

```

1 "QD < 2.0";"QualityByDepth"
2 "MQ < 40.0";"RMSMapQual"
3 "FS < 60.0";"FisherStrandBias"
4 "HaplotypeScore > 13.0";"HaploScore"
5 "MQRankSum < -12.5";"MQRankSum"
6 "ReadPosRankSum < -8.0";"ReadPosRankSum"

```

GATK, when filtering variants, will check each of these indexes (the first string on the left, i.e. QD) to check if some of the variants match the filter. If yes, the mutation is flagged with the filter name, the string between "", to be later analyzed and possibly removed.

GATK also provides *Variant Quality Score Recalibration* (VSQR), which uses the same machine learning based approach exploited in the base quality recalibration step to recalibrate the quality value of each called variant. How-

ever, this technique requires huge quantities of high quality data to work, a resource that we didn't (and don't) have. As a final remark in this quick discussion about variants calling in NGS, please note that GATK documentation suggests to call more samples simultaneously to improve the result. The rationale is that in this way the caller will use variants coming from all the samples to weight if a given variant do exists or represent a sequencing artifact. Albeit this may prove to be helpful, the aim of our pipeline was to give a mean of rapidly analyze many samples at once, and not perfectly calling each single variant. A batch calling can be possibly done when all sequences are available, but is not a procedure you should do on a daily basis.

How our pipeline behaves compared to the stock one? Before answering this question, we must do a preamble: the test runs we had from the lab were from 4 different chips, with the samples in the first mapped against rCRS, while the others were mapped against RSRS. Additionally, for the rCRS-mapped samples we also had Sanger sequencing data to compare them with, thus basing our consideration on a more solid reference point.

Table 9 on the next page shows the results for the 8 samples in the first chip. Starting from SNPs and taking Sanger as the golden standard, the vast majority of them are detected by all three methods, as visible in row "Ion/Sanger/GATK". Ion Variant caller seems to detect more SNPs than GATK, but manually verifying the reads all those variants doesn't have enough evidence to be reliably identified. GATK, on the contrary, doesn't recognize any additional SNP other than those in common with Sanger, providing a more reliable result. Forcing GATK to call even dubious positions, variants identified by Sanger missing in GATK output do get called, but with a quality so low to be immediately filtered out in normal conditions. From these initial results, it may seem that both NGS pipelines are not up to rival with Sanger, but there are two consideration to be done: first of all, data from this table derive from a first test chip that suffered from severe oscillations in the coverage along the molecule, reaching peaks as low as 1 or 2 and, in fact, this is the reason why both callers refused to call some of the variants. Secondly, we went back to the original Sanger data to double check if variants missing in NGS outputs were well supported by Sanger reads. The results was that even in Sanger a trained operator had to manually evaluate the chromatograms to call those variants. Moving to INDELS calls, and again keeping Sanger as the reference point, we observed quite disappointing results for the Ion caller, which failed to call all Sanger variants with the notable exception of 8281_8289del, which identifies haplogroup B. On top of this poor result, Ion also detected non-existent mutations, as reported in the table. On the contrary, GATK correctly identified all the INDELS detected in Sanger, incurring in a single false positive in Sample 2, probably due to a poly-c stretch occurring a couple of base pairs upstream. Again, our pipeline outperformed Ion stock caller both in sensitivity and specificity, delivering more accurate results.

Table 10 on the facing page reports the results for the same comparison performed on 12 additional samples, of which we unfortunately didn't have Sanger data. Drawing definitive conclusions from this second batch is much more difficult, since we miss a reliable reference point to compare with, but we partially solved this lack manually checking the contigs were the two methods were discordant. Starting from SNPs, we can observe how also in this case the vast majority of them are called by both pipelines. As seen in the previous comparison, Ion seems to call more SNPs than our custom pipeline, but pays this sensitivity in terms of specificity: manually checking the raw data there was no evidence for ~90% of the calls Ion did. In this second comparison, also GATK performances lowered significantly, ending up with a considerably higher number of false positives. INDELS calls done by Ion suffered from the very same issues we saw earlier: it misses most of the real variants (8281_8289del, instead, is correctly detected), and

Table 9: Comparison of variants detected by Sanger sequencing, Ion Torrent variant caller and GATK (against rCRS)

SNPs	Samples							
	1	2	3	4	5	6	7	8
Sanger	42	46	30	31	32	40	33	39
Ion/Sanger/GATK	35	43	27	28	32	40	33	38
Additional in Ion	0	0	4	3	1	0	0	3
Additional in GATK	0	0	0	0	0	0	0	0
INDELs								
Sanger	3	4	1	2	1	2	2	2
Wrong in Ion ^a	3	5	3	1	2	1	0	0
Wrong in GATK ^b	0	1	0	0	0	0	0	0

^a Fails to call Sanger INDELs, with exception of 8281_8289del.

^b Calls all the INDELs identified by Sanger sequencing.

Table 10: Comparison of variants detected by Ion Torrent variant caller and GATK (against RSRS)

SNPs	Samples											
	1	2	3	4	5	6	7	8	9	10	11	12
Ion/GATK	43	39	49	51	45	53	55	52	54	41	40	42
Called by Ion	6	9	6	4	2	3	4	3	7	14	10	3
Called by GATK	0	1	1	1	7	3	2	0	2	!o!	0	0
INDELs												
Called by Ion	0	2	4	3	0	1	5	3	2	3	4	4
Called by GATK	6	5	6	13	9	10	5	10	7	5	8	7

suffers from many false positives. Our pipeline, on the contrary, managed to identify most of the real INDELs, but with a slightly higher level of false positives if compared to the previous test. Summing up, Ion caller seems to struggle quite a lot also in this second batch with both SNPs and INDELs, while GATK performance seems to decrease, in particular for SNPs. There's a reason behind this behavior? In fact, *there is*. Analyzing this second batch we were forced not to use the know variability to help GATK calling the variants. The motivation is quite simple: all the variability detected by the 1000Genomes project is mapped against rCRS, not RSRS. As there is no easy automatic way to remap these data, we disabled this feature. From the effect this had on the results, you can pretty well understand how much this affects the pipeline. We're working to remap the variants against this new reference, but the work is still underway at the moment of writing this thesis. INDELs doesn't seem to be affected by this problem probably because they represent only a very small fraction of the variants identified in the 1000Genomes project, and thus GATK doesn't have enough data to improve their calling anyway.

INDEX

5.1	The Systems Biology challenges	81
5.2	The NU-AGE study	82
5.2.1	Data management	82
5.2.2	Preliminary data analysis	83
5.3	The GEHA Project	84
5.3.1	Background and methods setup	85
5.3.2	Single variants association	86
5.3.3	Multiple variants association - Epistasis	86
5.4	Next Generation Sequencing data analysis	89

5.1 THE SYSTEMS BIOLOGY CHALLENGES

Providing answers to the challenges science is facing in the last years requires an important shift from a "single mechanism" rationale to a systems biology approach, which pervasively analyze biological entities in their entirety, modeling both the inner workings of each biological system and the interactions between them. Phenomena such as inflammaging (CEVENINI et al. 2013; FRANCESCHI et al. 2000, 2007b) must be engaged in this way to obtain meaningful and reproducible results, as postulated in CALÇADA et al. 2014. However, allow these approaches to unleash all their power requires a considerable amount of *clean* data to work with. Thus, the first step in a successful modeling attempt should be to obtain high quality datasets to base the analysis upon.

A possible answer to this requirement are well-managed clinical and non-clinical studies, which serve as an extraordinary source of data proxying the functionality levels of many biological systems. NU-AGE (BERENDSEN et al. 2014; SANTORO et al. 2014), a FP7 European project focused on contrasting inflammaging via a 1-year diet intervention, is a good candidate to act as data source for system biology approaches. In fact, the project aims to collect data about many different fields (i.e. physical and cognitive status, socio-economics indexes) and couple them with cutting-edge omics analyses (i.e. genomics, lipidomics, metabolomics).

A second, somewhat explosive, source of data is represented by *Next Generation Sequencing*. Techniques such as NGS-based epigenetic studies (MEABURN et al. 2012) and RNA-seq, which is able to give a holistic view of the whole transcriptome, have freed sequencing from the long-standing limitations of being restricted to only a part of the life equation, genetics. Research can now obtain gigabytes of data (or tens of genomes, if you prefer) in a matter of weeks, not years, thus making possible to design studies that were not even thinkable before this revolution occurred. Ion Torrent, a NGS approach based on the ion discharge that occurs when a dNTP is incorporated into a growing DNA strand, concurs in this very active and competitive arena with a very high sequencing speed and low upfront and operating costs (MERRIMAN et al. 2012). Recently, this platform was also applied in epigenetic studies (CHENG et al. 2013; CORLEY et al. 2015).

As we said in previous chapters, the very final goal of systems biology is building models, possibly mechanistic, to explain the biological phenomena under analysis. Important efforts have been done in this sense in the inflammaging field, with the modeling of the NF- κ B pathway (BASAK et al. 2012;

TIERI et al. 2012) representing one of the best examples. Inflammaging, as the term says, is related to aging: it represents a low-level chronic inflammation common in elderly people, which is established during the aging process. Aging is also linked to the loss of metabolic flexibility (DIPIETRO 2010; GALGANI et al. 2008), that may ultimately led via multiple pathways to the insurgence of pathological states such as type-2 diabetes (CORPELEIJN et al. 2009; PONUGOTI et al. 2012). Eventually, the last link needed to close this very long chain is made by the fact that immune response itself may be impaired by macronutrients deficiencies (HOTAMISLIGIL et al. 2008), providing the soil on which inflammaging may sprout. Hopefully, the NU-AGE project will help disentangling these very complex interactions with its unmatched high-quality database.

5.2 THE NU-AGE STUDY

NU-AGE enrolled a total of 1250 free living subjects in the age range between 65 and 79 years old in 5 different European countries, equally subdivided in males and females. After an initial characterization, subjects were randomly assigned to one of the two branches constituting the study: diet and controls. Subjects in the first branch followed a 1-year long diet intervention, based on a Mediterranean fortified diet tailored on the elderly's needs, while controls subjects were only recontacted after 12 months for the follow-up without providing them any specific nutritional advice. Each participant was deeply characterized by mean of five standardized questionnaires (encompassing dietary habits, life-style, physical activity, health and functional cognitive status) and, through the collection of blood, urine and faecal samples, with omic and non-omic analyses.

5.2.1 Data management

Given the complexity of the study, is it easy to understand the pressure under which the data collection process was carried out. Errors at the data entry step cause difficult to solve problems at the database level. For this reason, in NU-AGE we aggressively fought errors from the very first moment acting at many different levels, a detailed description of which you can find in the Result chapter of this thesis. Briefly, we designed a data entry interface to provide on-the-fly correction of the data and, where strict rules could not be applied due to the innate variability biological entities have, we proceeded validating the values analyzing their distributions at T0 and reporting suspicious outliers. Eventually, we defined strict boundaries using as limits the 5th and 95th percentiles of the distribution.

To overcome the limitations the software we based our interface upon had (Epidata Entry v3.1), we implemented from scratch a Python-based pipeline to automatically perform data validation and error reporting, with a built-in management of false positives. An additional benefit our solution has is the possibility of leverage Python capabilities to build complex validation rules that are out of the scope of the previously available interfaces. Even if the inner workings of NU-AGE Validator may result too complex to understand for people lacking a computer science background, the whole suite is built with reusability in mind. Questionnaires and checks defined following the EpiData Entry 3.1 version of the format can be seamlessly loaded in our system, whereas new CRFs can easily be defined exploiting the YAML format.

The very last point I would like to discuss here regarding the data entry process, but which also pertains to the best practices for data management, is variables documentation. This is probably the tenth time we run across this concept, but I really believe that it represents a fundamental achievement each project must aim to. The data "half-life" is much longer than

the duration of the project that generated them, but require an additional effort to ensure it reaches the maximum lifespan. This "additional effort" is indeed documenting them. We're not talking of some exoteric who-knows-which-format file to be manually crafted. The documentation for the whole NU-AGE is contained in a single Excel file, which can be easily shared and accessed by people looking for the description of what each dataset contains. In our case, building this file was extremely easy thanks to the parsers available in Validator, but the process should not represent a huge pain in all the projects where data entry was powered by a data entry interface. Drawing from our direct experience, documentation started helping us even before the database official opening for queries: researcher participating in the project are now scrutinizing it to understand the variables they need to answer to their own scientific questions, and will be ready to provide us a list of them as soon as the data will go live, reducing dead times. This approach also eases the workload on the data management team: providing the right data to a group asking for variables A, B and C is much more easier than providing data for a generic "cognitive status".

No official statistics was maintained regarding the number of fixed errors thanks to this multi-levelled approach. Obviously, we can't track the number of alerts each Epidata deployment has issued until now. At the same time, Validator was not thought to provide this kind of feedback, being focused on a single complex task: driving out errors. However, we can state that several systematic errors were detected and solved thanks to Validator, along with a plethora of individual inconsistencies scattered among all questionnaires and centers, most of them caused by erroneous indexes calculations. Also, the approach we described for continuous variables, albeit simplistic, helped us in realizing that a more extensive comparison work had to be carried out to make the nutrients data produced by each centre comparable. This initial consideration allowed us to track inconsistencies that, to be fixed, required to get back to the software houses that originally developed the softwares used by the dietitians to estimate the nutrients intake. Eventually, this gave life to an important comparison work between the different national guidelines that will be surely helpful to forthcoming projects and to better shape possible european-wide efforts to act on diet as a mean of improving health, such as those requested by the EU commission in the framework of Horizon 2020.

5.2.2 Preliminary data analysis

Moving on in our discussion, we can now explore which information we already got from NU-AGE. As previously said, the data entry process is still underway, and the database, especially regarding T1 data, is far from being complete. However, datasets regarding T0 already provided a wealth of data that was worth to explore. The main result we had from these preliminary analyses was that a custom-made compliance index to evaluate subjects was required to better understand the project outcomes. This consideration mainly moved from the PCA analysis we did on the nutrients data gathered at T0, from which we understood that the relative positioning of participants diet in respect to the NU-AGE diet was much more complex than we expected. At a first glance, the analysis of the first four components we saw in the Results section may seem discouraging, given the fact that most participants seems to have a diet that pretty much matches the NU-AGE one already. However, combining the first three components in the 3D scatterplot we presented in the previous chapter, it should be clear that the real picture is more complex than that: a subject presenting a value for the first component very similar to the "optimal value" may at the same time have a much worse value for the second and the third components. I really believe that the title with which we presented the 3D scatterplot, "The NU-AGE Cloud", represents the best definition of the situation we're facing:

a complex yet impalpable net formed by several interplaying parameters. Nonetheless, result from omics and non-omics assays must be weighted by this net of parameters: if a conspicuous number of people enrolled in the diet branch didn't followed the diet we are at risk of drawing the wrong conclusions. This is the reason why UNIBO, in strict collaboration with many of the partners involved in the nutritional trial, started investigating the possibility of creating an index able to condense in a single number the aforementioned net, making weighting NU-AGE results a reality. The rationale we proposed is indeed rather simple: we previously introduced the concept of defining a diet as a point in a n -dimensional space, where n is the number of nutrients dietitians monitor in the subjects. In a similar way, the point representing the NU-AGE diet can be determined in the same space. Then, computing even a simple euclidean distance between these two points would give us a first coarse index to work with. Nonetheless, this is a rather simplistic view of the problem. Different nutrients do have a different importance in determining a person health, and should thus count more in our index. At the same time, most of them are not symmetrical: being over or under the threshold for sodium doesn't affect the system at the same way. NU-AGE consortia is still working on solving these issues, which require a careful revision of both the literature and the already available nutritional indexes to identify clear answers to the aforementioned problems. Nonetheless, once obtained, it will change the rules of the game, allowing us to carefully estimate each subject position in respect to the target diet. This will help us in better determining the NU-AGE cloud, but will also offer an additional advantage in terms of *subjects tracking*. Indeed, the very important information we want to have is not the starting or ending point of each participant, but their *trajectories* in the nutritional space (CALÇADA et al. 2014). Subject enrolled in the intervention branch were requested to fill a 7-days food record at T0 and T1 and a 3-days food record at months 4 and 8, while control subjects filled only the 7-days food record at T0 and T1. These data allow us to have a clear picture of their nutrient intakes and to compute a compliance index for each timepoint, thus following the subjects trajectory in time. It is clear that a diet subject is expected to move towards the diet as time goes by, if even by a little amount, while controls should at maximum circle around a fixed point. Any deviations from these trajectories will suggest that the data we gathered from deviating subjects should be weighted differently from the others. In this way, we elegantly resolve all the "bad news" we had from PCA analysis: we don't care any longer of the single point, being it at the T0 or T1, but we'll evaluate our subjects on their whole "voyage". Moreover, if this index will prove itself reliable, it may even be adopted to guide diet interventions based on the NU-AGE diet aimed to reduce, or even prevent, the inflammaging state, thus providing people with additional healthy years, as Horizon 2020 mandates.

Independently from the amount of work we, together with all NU-AGE partners, already carried out, the data analysis is still at the beginning, with many challenging tasks yet to come. What I personally tried to achieve with my work on this project was laying solid foundations on which analyses, such as those based on multiplex networks (MENICHETTI et al. 2014; SUGIHARA et al. 2012), can be safely designed and carried on.

5.3 THE GEHA PROJECT

NU-AGE wasn't the only topic on which my work was focused on. If for this first project most of my efforts went to planning a solid data management pipeline, I had a chance of touching also the data analysis part, thanks to the data gathered by the GEHA project. GEHA (FRANCESCHI et al. 2007a; SKYTTHE et al. 2011) enrolled 2650 long-lived 90+ sib pairs and 2650 younger, ethnically matched controls from 11 European countries. Out of the more

than 2500 participants, the project successfully sequenced the whole mtDNA molecule in 1292 samples, mainly coming from Finland and Denmark. Sequencing was carried out in Bologna for a small subset of the sequences, while the majority of them were processed by the BGI (China). This splitted source was the main cause of the first problem we had to solve even before starting to analyze the data. In fact, routinely performed quality checks, carried out following guidelines suggested by the literature (SALAS et al. 2005), detected how sequences obtained in China were of a clearly lower quality, harboring an anomalously higher number of mutations than expected. This fact suggested us that something may have gone wrong in the sequencing or in the assembling step. Since raw data were available, we decided to rebuild the sequences from scratch to attempt their recovery. We rapidly built a short but effective pipeline based on Phred, Phrap and Consed (EWING et al. 1998a,b; GORDON et al. 1998) and reanalyzed the data. The result was quite positive for the vast majority of the samples, with the whole sequence obtained right at the first attempt. However, to successfully reconstruct a small set of lower quality samples, we had to lower the pipeline strictness in quality terms. In this way, we managed to retrieve the whole sequences, at the cost of relying on the human intervention to fix errors due to previously excluded low-quality reads. This work allowed to detect, comparing the batch of sequences assembled in China to the one obtained by us, ~700 errors mainly clustered around homopolymer stretches. It is unquestionable that, albeit extremely time-consuming, this effort severely increased the quality of the dataset, providing more consistent data to propel the following analyses.

5.3.1 Background and methods setup

In 2014, RAULE et al. delivered some insight of the initial analyses on the GEHA mtDNA sequences, where the main outcome was represented by the fact that no single mutation was found associated with the longevity phenotype. Nonetheless, when considering the relative abundance of non synonymous variants within subunits forming the OXPHOS complexes, statistical significant associations appeared for subunits composing the OXPHOS complexes I, III and V. This suggested us to further analyze the sequences from an epistasis perspective. Literature was already generous in describing methods to detect epistatic effects (CORDELL 2002; EMILY 2012; MOORE 2015; MOORE et al. 2014; RITCHIE 2015) but, since we realized that none of them was able to fully exploit the datasets we had, we focused on developing new, more powerful approaches. The first candidate we tested was entropy analysis, as defined by SHANNON 2001. Results, as previously shown, did allow us to detect interactions among variants, but the approach somewhat lacked in answering to our original question: *the probability of a subject being long-lived given a signature of interacting mutation*. We thus decided to follow a different approach, based on Bayesian statistics. Initially, we tested this method to detect single SNPs associated to long-lived subjects and, in doing so, we analyzed two possible representations of the data: from an AA point of view, in which only non-synonymous mutations are considered, and from a nucleotide point of view, where all variants, even synonymous ones, are taken into account. Obviously, the former representation do provide results that are more readily understandable: non-syn mutations change an AA in the polypeptide chain, which in turn can hamper the protein functionality. On the contrary, explanations for a change in risk ratios caused by synonymous mutations are harder to give. Still, for variants occurring within gene boundaries, theories about the codon bias and the different processivity of different tRNA may provide an explanation (CAMIOLO et al. 2012; W. JIA et al. 2008; TATS et al. 2008). The biological effect for variants in non-coding regions, instead, should be sought in their regulatory function, i.e. the Dloop involvement in the mtDNA replication (WALLACE 2005).

5.3.2 Single variants association

We first tested the method looking for singularly associated variants, without taking epistasis interactions into consideration. Results for our AA-based analysis were encouraging, since our algorithm was able to grasp associations that were overlooked by previous analyses. In particular, we obtained significant Bayes factors (BF) for mutations harbored in ATP6, ND6, Cytochrome B and various other subunits. This finding, in contrast with the conclusions in RAULE et al. 2014, suggests that single non-syn mutations are indeed involved in shaping the longevity phenotype, with log risk ratio (RR) as relevant as 3.5 and -3.6, meaning a 33-fold increase and a 35-fold decrease in the risk of not being long-lived, respectively. It is important to note that log Risk Ratios are *relative* risks: given the very low probability in the general population, the absolute risk is much lower. Similar results were obtained considering the nucleotide-based representation of our datasets, in which all results previously obtained at the AA level were confirmed at the nucleotide level. Nonetheless, the strongest hit was due to the Dloop, with a Bayes factor of 16.5 and a log risk ratio of 3.77 (43-fold increase). This additional discovery allow us to postulate that also the non-coding parts of the mtDNA molecule do play a role in determining the longevity phenotype. In spite of this promising results, due to the fact that the method was first thought to evaluate only epistatic interactions, we need to consider the effect linkage disequilibrium, which is particularly strong in the mitochondrial DNA due to the lack of recombination, may have in the analyses. We're not saying that these results are inconsistent, but that the relative abundance of the found associated variants may partly be due to mutations that are in linkage: this would cause to detect as associated to the phenotype variants that are only in strong linkage with a mutation associated to the phenotype. Another critical point is represented by the fact that the mutation rate is not uniform on the whole molecule: Dloop has a much higher rate than the coding part, and mutational hotspots such as HV1 and HV2 mutate even more (STONEKING 2000). Given this complex panorama, the prior we used to provide the method with an estimate of the variability expected at each position may not be sufficient. Empirical tests we performed manually changing this prior did not caused severe oscillations in the results, but we're in any case planning to build a more advanced prior based on the analysis of high-quality sequences freely available in GenBank (BENSON et al. 2013), that will solve this problem for good.

5.3.3 Multiple variants association - Epistasis

After testing and tuning the method with single variants association analysis, we moved to epistasis detection. As common sense suggest, the complexity of such an analysis is much greater than the previous one. Indeed, in single associations we are comparing only two hypotheses – the variant is associated or not – while determining the kind of interaction occurring between two mutations requires to test more of them. After careful thinking and investigation, we came up with a list of 17 possible models of interaction, all of which are listed in Table 7 on page 74. These models do encompass all the ways in which two alleles may interact, ranging from pure interaction to more complex cases in which a variant is dominant on the other or in which the two variants singularly cause different damages while their co-presence reconstructs the pathway. The downside of having such a fine analysis is represented by the fact that it do produce a huge quantity of raw data – for each couple we obtain 17 Bayes factors for the 17 models, each of which has 3 separate Risk Factors linked to three possible allele combinations – that must be then filtered to extract the meaningful information.

Final results, contained in Table 8 on page 76, clearly depict a complex panorama where multiple couples do interact and change the probability of being long-lived. The "Complete Interaction" model for couple ATP6_60 – ND1_17 reached a very significant Bayes Factor of 191 (150, following Kass' interpretation table, represents the minimal threshold to claim a Bayes factor as "Very strong") and a log risk ratio of 1.418 (4-fold increase) for the M1W2 allele combination. At the same time, combination W1M2 has little to no effect (RR: 0.017), while log Risk Ratio for the combination M1M2 is 0.847. The arrangement of these values suggests that, while the first mutations is detrimental, the second doesn't have an effect on its own but counterbalances the damaging effect of the first mutation if they both occur. The following couple, ATP6_60 – ND5_159 (BF: 108, RRs: 1.003, -0.481, 1.724), is indeed better explained by a "Mutation 1, Interaction" model, in which the first mutation has an effect on the phenotype (RR: 1.003), the second has a very little protective effect (RRs: -0.481) but they interact together causing a 5-fold increase in the risk of not being long-lived. This couple also offers a second cause of reflection: in these analyses, at least in some cases, we're dealing with rare mutations, which occur in the general population at very low frequencies and it is thus normal to see them in few of our subjects. This, however, makes correctly estimating RRs a difficult task. For example, in the just described couple we had a "control,cases" count of 3,9 for M1W2 and a count of 2,1 for W1M2. While RR estimation for the first is pretty solid, the RR estimation for the second may suffer of overestimation. It is important to note that model selection and RR calculation are two separate processes, the latter being more influenced by extreme cases. In this peculiarity we should look for the reason why our model selection do choose a "Mutation 1, Interaction" instead of a "Complete Interaction", which would better fit observed RRs. For comparison, the third couple, ATP6_60 – CYB_7, which has stronger counts, shows a perfect match between the called model and the RRs ("Mutation 1, Interaction", RRs: 1.327, 0.122, 1.435). The same holds for the fourth couple (CYB_158 – CYB_16, "Concordant 2", RRs: 0.026, 1.243, 0.197). The fifth couple, ATP6_90 – ND2_119 ("Complete Interaction", RRs: 0.003, 0.003, 1.280), gives us the possibility to discuss a second current shortcoming of the method, on which we're already working on. RRs in this case are 0.003, 0.003 and 1.280, with the first and second one identical up to the third decimal digit, pointing to a "Interaction only" model. Nonetheless, the algorithm selects a "Complete Interaction" model as the best representation for this couple behavior, since nothing prohibits $f1 = f2$ (see Table 7). We spent a bit of time trying to understand the reasons behind that and we eventually found two possible explanations: the first is attributable to the fact that for cases M1W2 and W1M2 we don't have any count. However, with priors we are giving to the system a vague idea of what it should expect in terms of mutation counts in each position, and the interplay between these priors and a zero count may in some way affect the selection process. The second reason strictly depends from the first: if the counts are off due to the priors or they are very low on their own, the system struggles in deciding which is the best model, with many related models reaching comparable Bayes factor. At the moment, the system does not take this intrinsic indecision into account, simply choosing the highest scoring model. We're already working on a second iteration of the algorithm that will address this problem acting at two different levels: first, cases with a very low mutation count will be treated better gauging the effect priors have on the models fitness estimation. Second, the routine that picks the best model, which now simply takes the one with the highest Bayes factor, will be augmented to consider also additional indexes, such as mutations counts, Risk Ratios or other biological plausibility measurements, thus empowering a more wise decision that should ultimately lead in all the cases to the selection of the simpler possible model to explain the phenomena.

Independently from the issues just described, caused by pitfalls that are common in newborn algorithms implementations, the method hereby described was able to effectively analyze a massive number of candidate couples (~36000) for their effect on the longevity phenotype, possibly regulating the inflammaging process, in a very short period of time. In fact, thanks to the adoption of analytical methods instead of numerical methods, the time required to analyze a single couple stands around ~0.008 seconds, which means that only 5 minutes were required to complete the just discussed analysis using a single processor. Moreover, since the method computes each couple statistics independently from the others, it is easily parallelizable. Once obtained, this feature will further push the limits of the method, in term of computational scalability, onwards. A so low computational time finally allows to think of completely exploring at least the mitochondrial variability in matter of hours, not years. For example, GEHA detected a total of ~2000 mutated positions at the nucleotide level in its mtDNA samples, which translates in 4 million couples to analyze. Processing them at the actual pace would require ~9 hours ($4,000,000 * 0.008 = 32000$ seconds, 8,8 hours.). After the parallelization of the method, this estimate will be severely reduced, the extent of which depending on the number of CPUs devoted to the task. At that point, including the nuclear variability GEHA detected in the analysis will be totally feasible. A paper describing the strong points and the pitfalls of this statistical analysis method is in preparation.

Before concluding the discussion of GEHA results, we have three additional points to touch. The first is that it is normal that results for the single and couple-based analyses do not overlap, since in the second case we're analyzing a different hypothesis. For example, when we analyze the M1W2 combination, we're not only considering the presence of M1, but also the absence of M2, and the same concept holds for W1M2. After all, having out-of-expected counts for i.e. W1M2 may suggest a interaction between a mutated and a wildtype allele, that our method will successfully recognize, not being at all influenced by labels like "mutated" and "wildtype". Given these premises, it is particularly striking the fact that ATP6 mutations scored as the most longevity-associated in the single analysis and many of the best scouring couples in the epistasis analysis do comprise ATP6. This strongly suggests that ATP6 may play a central role in determining if a subject will be long-lived or not. Additionally, this finding is perfectly in line with the conclusions of RAULE et al. 2014, which points to the OXPHOS complexes as mayor players in shaping longevity.

The second point I would like to discuss is the fact that log Risk Ratios, independently from being computed on single variants or mutation couples, can be summed to obtain the log Risk Ratio of having, for example, two independent mutations. This consideration paves the way to the estimation of the RR for a subject starting from its own mutational signature, simply adding the single RR due to each mutation or combination of. However, given the data we have at the moment, this must be still considered an incomplete dream: how to deal with the fact that the same mutation may have a different RR if considered alone or in combination with another variant, let alone if involved in higher grades interactions? Unfortunately, our knowledge is far from being advanced enough to answer to this last question.

The third, and last, point is about flexibility. In the context of this thesis, we've always declined this method as a way to evaluate epistatic effects occurring between two or more mutations. But this was just for convenience, due to the problem we were discussing. The algorithm expects as input a binary matrix but doesn't make any assumption on the kind of data the matrix itself contains. At the same time, all the models of interaction we defined are applicable in most of the biological contexts. In practice, this means that the very same method may be used to test more heterogeneous

scientific questions, such as: "Do this mutation interact with high blood pressure to give rise to cardiac problems?" or, more generally, "Do being overweight interacts with a lower cognitive function to cause accelerated aging?". For sure, NU-AGE will put this technique at good use: "Do this mutation in combination with an impaired nutrients intake shape inflammation and inflammaging phenomenas?" is just one of the questions that may find an answer with this method.

5.4 NEXT GENERATION SEQUENCING DATA ANALYSIS

Advanced approaches as the one just described usually require massive quantities of raw data to successfully identify variants that have only small effects on the phenotype, which indeed is the working hypothesis when dealing with complex phenotypes. Next generation sequencing represents the most successful attempt to provide to these methods enough fuel to run consistently. It is indisputable that NGS completely revolutionized the way in which researcher think about sequencing, delivering at the same time an extremely high throughput and a consistent fall of the upfront and operative costs. During my PhD, I had the opportunity to work with one of these machines, an Ion Torrent PGM™ platform, with the aim of implementing a new pipeline able to autonomously perform all the steps needed to transform the raw data produced by the sequencer in the final result: variant calls. In doing so, we adopted two open source softwares, Picard and GATK (DEPRISTO et al. 2011; MCKENNA et al. 2010; Picard Website), which together concur in providing all the instruments needed to complete all the stages of which the variant calling process is made of. GATK developers' team, drawing fully from their own expertise in sequencing at the Broad Institute, prepared a set of Best Practices (VAN DER AUWERA et al. 2013) to follow in most use cases, on which our pipeline is heavily inspired. As previously explained, the "life" of variant calling is mainly constituted by repeating the same, highly predictable, commands invoking already written algorithms. What we thus did was making this process automatic, thanks to a program implemented in Python that leveraged the excellent Plumbum library to interact with the system and monitor the tasks.

A direct speed comparison between our custom pipeline and the instrument manufacturer's one is not possible, since our implementation doesn't include the read alignment task. There are many reason for that, the main being the high computational requirements needed to accomplish this step. Furthermore, successful contigs assembly is critical for obtain high quality results, and must be double-checked by an operator before proceeding to the next stages. This assignment is usually achieved with softwares like FastQC (FastQC Website), which automatically compute quality indexes useful to rapidly evaluate the contigs quality. It is thus contradictory to add a step requiring human intervention to an automatic pipeline. However, given the fact that most of the bad runs can be identified right from the sequencer, and that the computational power for aligning mitochondrial reads is much lower than we expected, we eventually added the reads alignment step in our pipeline, but this inclusion is so recent that its benefits are still to be assessed in the long term and thus can't be described here.

On the contrary, a comparison of the two pipelines in terms of sensitivity and sensibility in variant calling is indeed possible. As reported in the Results section, our custom pipeline outpaced the manufacturer's system in calling both SNPs and INDELS. However, this difference partially faded out when we were forced to disable the GATK feature that includes information given by known variability in the calling of variants hosted by new samples. This clearly demonstrates the importance broad project such

as 1000Genomes and the *open data revolution*, which is infecting more and more researchers everyday, both have in improving the daily routine of labs all around the world. Personally, I deeply believe in both of them.

6 | CONCLUSIONS

The reader, reaching this point, may ask himself, or herself, why all the work hereby described matters with a PhD in experimental pathology. Indeed, the whole thesis is focused on providing and analyzing data to better understand how to counteract inflammaging and explain the longevity phenotype, two deeply related phenomena.

Crunching everything in a bullet list, in this work we achieved the:

- Design and implementation of a data entry interface in Epidata for the NU-AGE study;
- Design and implementation of the database models for NU-AGE;
- Deployment of a LabKey instance for the data management needs of NU-AGE;
- Design and implementation of a framework for data validation in clinical and non-clinical projects;
- Preliminary analyses on NU-AGE nutrients data;
- Drafting of a compliance index to evaluate NU-AGE subjects nutritional trajectories;
- Development of a Bayesian-based method to evaluate genetic variants associations and their epistatic interactions;
- Development of a pipeline for Next Generation Sequencing variant calling starting from Ion Torrent data;

Going into the details, in NU-AGE my work was to ensure that all the data captured in the context of the project were collected with the highest possible level of standardization, hence providing a solid base on which analysis may have been planned. Striving to reach this goal we produced a set of high-quality tools that have the potential of being adopted in forthcoming studies, Validator among the others for its potential to be used in other studies. Also our Epidata-based interface, albeit suffering of the old-age problems we already discussed, may be put at good use in certain context, i.e. where a single center is involved in the study.

After analyzing the *T0* data about the nutrients intake, we realized that clearly understanding how to weight each subject data in function of his/her adherence to the diet, or in function of being a good control, was a key requisite to achieve the project goals. We thus started the definition of an index to precisely estimate the diet compliance of each participant. This particular work, unfortunately, is still in development. However, when established, it could be used well outside the boundaries of the NU-AGE study, for example to evaluate how a person diet relates with the diet the project defined to help counteract inflammaging. Eventually, should NU-AGE confirms the diet as an effective way to contrast the chronic inflammation that characterize the aging process, this index may acquire a significant meaning also in pathology, given that most age-related diseases have an inflammatory base.

GEHA aim was somewhat more limited, focused solely on the genetic components of longevity. Since previous analyses suggested that single mutations were not enough to explain the aging process and the answer was instead probably lying in the interaction of multiple mutations, we designed a new method based on Bayesian statistics to explore the possibility

of epistatic effects among different variants. The developed algorithm has proved itself as a good and efficient tool to explore the association of both single and multiple interacting mutations to the longevity phenotype. Nonetheless, it suffers of all the usual youth issues every implementation has at its first iteration. We'll work in the future to iron these wrinkles out. As far as the result we obtained applying this method to GEHA data, we were able to detect single variants mutation associated with the aging process that were overlooked by previous analyses. At the same time, we identified several AA mutations forming couples that seems to impact on the longevity phenotype. While additional checks are required before fully accepting these results as true, i.e. checking if the two AA changes affect interacting protein domains, from a statistical point of view they are already enough strong by themselves.

Speed is the driving problem that dictates which analysis can or cannot be done in reasonable time in the world of interactions evaluation, being it a *NP-hard* problem. Our method takes around 7 milliseconds to evaluate all the possible ways two mutations have to interact. This performance make exploring the whole mitochondrial variability, at the nucleotide level, doable in a matter of hours. Furthermore, we're already planning to extend the analysis to nuclear variants, finally shedding a bit of light on the nuclear-mitochondrial crosstalk. Finally, the fact that the method does not make any assumption on the input data allows to analyze scientific questions outside pure genetics: in theory, evaluating if a certain mutation interacts with some pathological state to give rise to a particular phenotype should not be a problem anymore.

Finally, our work to define a new pipeline to process Ion Torrent raw data eventually ended in a system that, at least for what concerns the mitochondrial DNA, outperforms in the variant calling task the stock pipeline provided by the manufacturer both in terms of specificity and sensitivity. As for the Bayesian method cited above, also in this case we're committed in further improving the pipeline performances by mean of providing it high-quality known variability data and including the contings assembly step.

This thesis, in front of several closed questions, leaves some open ones behind. In extreme cases, it even contributed in opening some. However, for most of them a proper solution was at least drafted. After all, that's how science does work. Whenever possible, all the code developed was designed with reusability in mind, building tools that may be in future used by different projects and individuals to tackle the same problems we faced ourself, leaving them more time to enjoy the much more funny task of breaking their own mind on understanding results.

Stealing from the words of Dr. Bang Wong, creative director of the Broad Institute of MIT, a small wish for all the people that spent their time in reading this 3-years extract of my life:

I hope you'll find good things in your data!

AUTHOR'S PUBLICATIONS

- CALÇADA, DULCE, DARIO VIANELLO, ENRICO GIAMPIERI, CLAUDIA SALA, GASTONE CASTELLANI, ALBERT DE GRAAF, BAS KREMER, BEN VAN OMMEN, EDITH FESKENS, AURELIA SANTORO, et al. (2014). "The role of low-grade inflammation and metabolic flexibility in aging and nutritional modulation thereof: A systems biology approach". In: *Mechanisms of Ageing and Development* 136-137, pp. 138-147. ISSN: 18726216. DOI: 10.1016/j.mad.2014.01.004.
- RAULE, NICOLA, FEDERICA SEVINI, SHENGTING LI, ANNALaura BARBIERI, FEDERICA TALLARO, LAURA LOMARTIRE, DARIO VIANELLO, ALBERTO MONTESANTO, JUKKA S. MOILANEN, VLADYSLAV BEZRUKOV, et al. (2014). "The co-occurrence of mtDNA mutations on different oxidative phosphorylation subunits, not detected by haplogroup analysis, affects human longevity and is population specific". In: *Aging Cell* 13.3, pp. 401-407. ISSN: 14749726. DOI: 10.1111/ace1.12186.
- SALVIOLI, STEFANO, DANIELA MONTI, CATIA LANZARINI, MARIA CONTE, CHIARA PIRAZZINI, MARIA GIULIA BACALINI, PAOLO GARAGNANI, CRISTINA GIULIANI, ELISA FONTANESI, RITA OSTAN, et al. (2013-03-01T00:00:00). "Immune System, Cell Senescence, Aging and Longevity - Inflamm-Aging Reappraised". In: *Current Pharmaceutical Design* 19.9, pp. 1675-1679.
- SEVINI, FEDERICA, CRISTINA GIULIANI, DARIO VIANELLO, ENRICO GIAMPIERI, AURELIA SANTORO, FIAMMETTA BIONDI, PAOLO GARAGNANI, GIUSEPPE PASSARINO, DONATA LUISELLI, MIRIAM CAPRI, et al. (2014). "mtDNA mutations in human aging and longevity: Controversies and new perspectives opened by high-throughput technologies". In: *Experimental Gerontology* 56. Mitochondria, Metabolic Regulation and the Biology of Aging, pp. 234-244. ISSN: 0531-5565. DOI: <http://dx.doi.org/10.1016/j.exger.2014.03.022>.
- SEVINI, FEDERICA, DANIELE YANG YAO, LAURA LOMARTIRE, ANNALaura BARBIERI, DARIO VIANELLO, GIANMARCO FERRI, EDGARDO MORETTI, MARIA CRISTINA DASSO, PAOLO GARAGNANI, DAVIDE PETTENER, et al. (2013). "Analysis of Population Substructure in Two Sympatric Populations of Gran Chaco, Argentina". In: *PLoS ONE* 8.5, e64054. ISSN: 19326203. DOI: 10.1371/journal.pone.0064054.
- VIANELLO, DARIO, FEDERICA SEVINI, GASTONE CASTELLANI, LAURA LOMARTIRE, MIRIAM CAPRI, and CLAUDIO FRANCESCHI (2013). "HAPLOFIND: A new method for high-throughput mtDNA haplogroup assignment". In: *Human Mutation* 34, pp. 1189-1194. ISSN: 10597794. DOI: 10.1002/humu.22356.

REFERENCES

BIBLIOGRAPHY

- ANDERSON, S, A T BANKIER, B G BARRELL, M H DE BRUIJN, A R COULSON, J DROUIN, I C EPERON, D P NIERLICH, B A ROE, F SANGER, et al. (1981). "Sequence and organization of the human mitochondrial genome." In: *Nature* 290, pp. 457–465. ISSN: 00280836. DOI: 10.1038/290457a0.
- ANDREWS, R M, I KUBACKA, P F CHINNERY, R N LIGHTOWLERS, D M TURNBULL, and N HOWELL (1999). "Reanalysis and revision of the Cambridge reference sequence for human mitochondrial DNA." In: *Nature genetics* 23, p. 147. ISSN: 1061-4036. DOI: 10.1038/13779.
- BANDELT, HANS-JÜRGEN, ANITA KLOSS-BRANDSTÄTTER, MARTIN B RICHARDS, YONG-GANG YAO, and IAN LOGAN (2014). "The case for the continuing use of the revised Cambridge Reference Sequence (rCRS) and the standardization of notation in human mitochondrial DNA studies." In: *Journal of human genetics* 59.October, pp. 66–77. ISSN: 1435-232X. DOI: 10.1038/jhg.2013.120.
- BARRETT, TANYA, STEPHEN E. WILHITE, PIERRE LEDOUX, CARLOS EVANGELISTA, IRENE F. KIM, MAXIM TOMASHEVSKY, KIMBERLY A. MARSHALL, KATHERINE H. PHILLIPPY, PATTI M. SHERMAN, MICHELLE HOLKO, et al. (2013). "NCBI GEO: Archive for functional genomics data sets - Update". In: *Nucleic Acids Research* 41.D1, pp. D991–D995. ISSN: 03051048. DOI: 10.1093/nar/gks1193.
- BAR-YAACOV, DAN, AMIT BLUMBERG, and DAN MISHMAR (2012). "Mitochondrial-nuclear co-evolution and its effects on OXPHOS activity and regulation". In: *Biochimica et Biophysica Acta - Gene Regulatory Mechanisms* 1819.9–10, pp. 1107–1111. ISSN: 18749399. DOI: 10.1016/j.bbagr.2011.10.008.
- BASAK, SOUMEN, MARCELO BEHAR, and ALEXANDER HOFFMANN (2012). "Lessons from mathematically modeling the NF- κ B pathway". In: *Immunological Reviews* 246.1, pp. 221–238. ISSN: 01052896. DOI: 10.1111/j.1600-065X.2011.01092.x.
- BEHAR, DORON M., MANNIS VAN OVEN, SAHARON ROSSET, MAIT METSPALU, EVA LIIS LOOGVÄLI, NUNO M. SILVA, TOOMAS KIVISILD, ANTONIO TORRONI, and RICHARD VILLEMS (2012). "A "copernican" reassessment of the human mitochondrial DNA tree from its root". In: *American Journal of Human Genetics* 90.4, pp. 675–684. ISSN: 00029297. DOI: 10.1016/j.ajhg.2012.03.002.
- BENSON, DENNIS A., MARK CAVANAUGH, KAREN CLARK, ILENE KARSCH-MIZRACHI, DAVID J. LIPMAN, JAMES OSTELL, and ERIC W. SAYERS (2013). "GenBank". In: *Nucleic Acids Research* 41.D1, pp. D36–D42. ISSN: 03051048. DOI: 10.1093/nar/gks1195.
- BERENDSEN, AGNES, AURELIA SANTORO, ELISA PINI, ELISA CEVENINI, RITA OSTAN, BARBARA PIETRUSZKA, KATARZYNA ROLF, NOËL CANO, AURÉLIE CAILLE, NOËLLE LYON-BELGY, et al. (2014). "Reprint of: A parallel randomized trial on the effect of a healthful diet on inflammaging and its consequences in European elderly people: Design of the NU-AGE dietary intervention study". In: *Mechanisms of Ageing and Development* 136-137, pp. 14–21. ISSN: 18726216. DOI: 10.1016/j.mad.2014.03.001.
- BIAGI, ELENA, MARCO CANDELA, SUSAN FAIRWEATHER-TAIT, CLAUDIO FRANCESCHI, and PATRIZIA BRIGIDI (2012). "Ageing of the human metaorganism: The microbial counterpart". In: *Age* 34.1, pp. 247–267. ISSN: 01619152. DOI: 10.1007/s11357-011-9217-5.
- BLAGOSKLONNY, MIKHAIL V. and MICHAEL N. HALL (2009). "Growth and aging: a common molecular mechanism." In: *Ageing* 1, pp. 357–362. ISSN: 19454589.
- BLANKENBERG, DANIEL, GREGORY VON KUSTER, NATHANIEL CORAOR, GURUPRASAD ANANDA, ROSS LAZARUS, MARY MANGAN, ANTON NEKRUTENKO, and JAMES TAYLOR (2010). "Galaxy: A web-based genome analysis tool for

experimentalists". In: *Current Protocols in Molecular Biology*. John Wiley & Sons, Inc. ISBN: 0471142727. DOI: 10.1002/0471142727.mb1910s89.

- BOUWENS, MARK, ONDINE VAN DE REST, NEELE DELLSCHAFT, MECHELD GROOTTE BROMHAAR, LISETTE C P G M DE GROOT, JOHANNA M. GELEIJNSE, MICHAEL MÜLLER, and LYDIA A. AFMAN (2009). "Fish-oil supplementation induces antiinflammatory gene expression profiles in human blood mononuclear cells". In: *American Journal of Clinical Nutrition* 90.2, pp. 415–424. ISSN: 00029165. DOI: 10.3945/ajcn.2009.27680.
- CALÇADA, DULCE, DARIO VIANELLO, ENRICO GIAMPIERI, CLAUDIA SALA, GASTONE CASTELLANI, ALBERT DE GRAAF, BAS KREMER, BEN VAN OMMEN, EDITH FESKENS, AURELIA SANTORO, et al. (2014). "The role of low-grade inflammation and metabolic flexibility in aging and nutritional modulation thereof: A systems biology approach". In: *Mechanisms of Ageing and Development* 136-137, pp. 138–147. ISSN: 18726216. DOI: 10.1016/j.mad.2014.01.004.
- CAMIOLO, SALVATORE, LORENZO FARINA, and ANDREA PORCEDDU (2012). "The Relation of Codon Bias to Tissue-Specific Gene Expression in Arabidopsis thaliana". In: *Genetics* 192.2, pp. 641–649. DOI: 10.1534/genetics.112.143677.
- CANNINO, G, C M DI LIEGRO, and A M RINALDI (2007). "Nuclear-mitochondrial interaction". In: *Mitochondrion* 7, pp. 359–366. ISSN: 15677249. DOI: 10.1016/j.mito.2007.07.001.
- CAPRI, MIRIAM, AURELIA SANTORO, PAOLO GARAGNANI, MARIA GIULIA BACALINI, CHIARA PIRAZZINI, FABIOLA OLIVIERI, ANTONIO PROCOPIO, STEFANO SALVIOLI, and CLAUDIO FRANCESCHI (2013). "Genes Of Human Longevity: An Endless Quest?" In: *Current vascular pharmacology* 12.5, pp. 707–717. ISSN: 1875-6212. DOI: 10.2174/1570161111666131219110301.
- CEVENINI, ELISA, ELENA BELLAVISTA, PAOLO TIERI, GASTONE CASTELLANI, FRANCESCO LESCAI, MIRKO FRANCESCONI, MICHELE MISHTO, AURELIA SANTORO, SILVANA VALENSIN, STEFANO SALVIOLI, et al. (2010). "Systems biology and longevity: an emerging approach to identify innovative anti-aging targets and strategies". In: *Current pharmaceutical design* 16.7, pp. 802–813. DOI: 10.2174/138161210790883660.
- CEVENINI, ELISA, LAURA INVIDIA, FRANCESCO LESCAI, STEFANO SALVIOLI, PAOLO TIERI, GASTONE CASTELLANI, and CLAUDIO FRANCESCHI (2008). "Human models of aging and longevity". In: *Expert Opinion on Biological Therapy* 8.9, pp. 1393–1405. DOI: 10.1517/14712598.8.9.1393.
- CEVENINI, ELISA, DANIELA MONTI, and CLAUDIO FRANCESCHI (2013). "Inflammageing." In: *Current opinion in clinical nutrition and metabolic care* 16.1, pp. 14–20. ISSN: 1473-6519. DOI: 10.1097/MCO.0b013e32835ada13.
- CHENG, CHRISTINE S, KUNAL RAI, MANUEL GARBER, ANDREW HOLLINGER, DANA ROBBINS, SCOTT ANDERSON, ALYSSA MACBETH, AUSTIN TZOU, MAURICIO O CARNEIRO, RAKTIMA RAYCHOWDHURY, et al. (2013). "Semiconductor-based DNA sequencing of histone modification states." In: *Nature communications* 4, p. 2672. ISSN: 2041-1723. DOI: 10.1038/ncomms3672.
- CLAESSON, MARCUS J., IAN B. JEFFERY, SUSANA CONDE, SUSAN E. POWER, EIBHLÍS M. O'CONNOR, SIOBHÁN CUSACK, HUGH M. B. HARRIS, MAIREAD COAKLEY, BHUVANESWARI LAKSHMINARAYANAN, ORLA O'SULLIVAN, et al. (2012). "Gut microbiota composition correlates with diet and health in the elderly". In: *Nature* 488.7410, pp. 178–184. ISSN: 0028-0836. DOI: 10.1038/nature11319.
- COCK, PETER J A, TIAGO ANTAO, JEFFREY T. CHANG, BRAD A. CHAPMAN, CYMON J. COX, ANDREW DALKE, IDDO FRIEDBERG, THOMAS HAMELRYCK, FRANK KAUFF, BARTEK WILCZYNSKI, et al. (2009). "Biopython: Freely available Python tools for computational molecular biology and bioinformatics". In: *Bioinformatics* 25.11, pp. 1422–1423. ISSN: 13674803. DOI: 10.1093/bioinformatics/btp163.

- CORDELL, HEATHER J (2002). "Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans." In: *Human molecular genetics* 11.20, pp. 2463–2468. ISSN: 0964-6906. DOI: 10.1093/hmg/11.20.2463.
- CORLEY, M J, W ZHANG, X ZHENG, and A K LUM-JONES A. MAUNAKEA (2015). "Semiconductor-based sequencing of genome-wide DNA methylation states". In: *Epigenetics* 20. DOI: 10.1080/15592294.2014.1003747.
- CORPELEIJN, E., W. H M SARIS, and E. E. BLAAK (2009). "Metabolic flexibility in the development of insulin resistance and type 2 diabetes: Effects of lifestyle: Etiology and Pathophysiology". In: *Obesity Reviews* 10.2, pp. 178–193. ISSN: 14677881. DOI: 10.1111/j.1467-789X.2008.00544.x.
- CUNNINGHAM, FIONA, M RIDWAN AMODE, DANIEL BARRELL, KATHRYN BEAL, KONSTANTINOS BILLIS, SIMON BRENT, DENISE CARVALHO-SILVA, PETER CLAPHAM, GUY COATES, STEPHEN FITZGERALD, et al. (2014). "Ensembl 2015". In: *Nucleic Acids Research*. DOI: 10.1093/nar/gku1010.
- DE MARTINIS, MASSIMO, CLAUDIO FRANCESCHI, DANIELA MONTI, and LIA GINALDI (2006). "Inflammation markers predicting frailty and mortality in the elderly". In: *Experimental and Molecular Pathology* 80.3, pp. 219–227. ISSN: 00144800. DOI: 10.1016/j.yexmp.2005.11.004.
- DEPRISTO, MARK A, ERIC BANKS, RYAN POPLIN, KIRAN V GARIMELLA, JARED R MAGUIRE, CHRISTOPHER HARTL, ANTHONY A PHILIPPAKIS, GUILLERMO DEL ANGEL, MANUEL A RIVAS, MATT HANNA, et al. (2011). "A framework for variation discovery and genotyping using next-generation DNA sequencing data." In: *Nature genetics* 43.5, pp. 491–498. ISSN: 1061-4036. DOI: 10.1038/ng.806.
- DIPETRO, LORETTA (2010). "Exercise training and fat metabolism after menopause: implications for improved metabolic flexibility in aging." In: *Journal of applied physiology (Bethesda, Md. : 1985)* 109.6, pp. 1569–1570. ISSN: 8750-7587. DOI: 10.1152/jappphysiol.01152.2010.
- DOMÍNGUEZ-GARRIDO, ELENA, DIANA MARTÍNEZ-REDONDO, CARMEN MARTÍN-RUIZ, AURORA GÓMEZ-DURÁN, EDUARDO RUIZ-PESINI, PILAR MADERO, MANUEL TAMPARILLAS, JULIO MONTOYA, THOMAS VON ZGLINICKI, CARMEN DíEZ-SÁNCHEZ, et al. (2009). "Association of mitochondrial haplogroup J and mtDNA oxidative damage in two different North Spain elderly populations". In: *Biogerontology* 10.4, pp. 435–442. ISSN: 13895729. DOI: 10.1007/s10522-008-9186-y.
- EDGAR, ROBERT C. (2004). "MUSCLE: Multiple sequence alignment with high accuracy and high throughput". In: *Nucleic Acids Research* 32.5, pp. 1792–1797. ISSN: 03051048. DOI: 10.1093/nar/gkh340.
- EMILY, M. (2012). "IndOR: A new statistical procedure to test for SNP-SNP epistasis in genome-wide association studies". In: *Statistics in Medicine* 31.21, pp. 2359–2373. ISSN: 02776715. DOI: 10.1002/sim.5364.
- EWING, BRENT and PHIL GREEN (1998a). "Base-Calling of Automated Sequencer Traces Using Phred. II. Error Probabilities". In: *Genome Research* 8.3, pp. 186–194.
- EWING, BRENT, LADEANA HILLIER, MICHAEL C. WENDL, and PHIL GREEN (1998b). "Base-Calling of Automated Sequencer Traces Using Phred. I. Accuracy Assessment". In: *Genome Research* 8.3, pp. 175–185. DOI: 10.1101/gr.8.3.175.
- FRANCESCHI, CLAUDIO (2007). "Inflammaging as a Major Characteristic of Old People: Can It Be Prevented or Cured?" In: *Nutrition Reviews* 65.SUPPL.3, S173–S176. ISSN: 00296643. DOI: 10.1111/j.1753-4887.2007.tb00358.x.
- FRANCESCHI, CLAUDIO, VLADYSLAV BEZRUKOV, HÉLÈNE BLANCHÉ, LARS BOLUND, KAARE CHRISTENSEN, GIOVANNA DE BENEDICTIS, LUCA DEIANA, EFSTATHIOS GONOS, ANTTI HERVONEN, HUANNING YANG, et al. (2007a). "Genetics of healthy aging in Europe: The EU-integrated project GEHA (Genetics of Healthy Aging)". In: *Annals of the New York Academy of Sciences* 1100.1, pp. 21–45. ISSN: 00778923. DOI: 10.1196/annals.1395.003.
- FRANCESCHI, CLAUDIO, MASSIMILIANO BONAFÈ, SILVANA VALENSIN, FABIOLA OLIVIERI, MARIA DE LUCA, ENZO OTTAVIANI, and GIOVANNA DE BENEDIC-

- TIS (2000). "Inflamm-aging. An evolutionary perspective on immunosenescence." In: *Annals of the New York Academy of Sciences* 908, pp. 244–254. ISSN: 0077-8923. DOI: 10.1111/j.1749-6632.2000.tb06651.x.
- FRANCESCHI, CLAUDIO, MIRIAM CAPRI, DANIELA MONTI, SERGIO GIUNTA, FABIOLA OLIVIERI, FEDERICA SEVINI, MARIA PANAGIOTA PANOURGIA, LAURA INVIDIA, LAURA CELANI, MARIA SCURTI, et al. (2007b). "Inflammaging and anti-inflammaging: A systemic perspective on aging and longevity emerged from studies in humans". In: *Mechanisms of Ageing and Development* 128.1, pp. 92–105. ISSN: 00476374. DOI: 10.1016/j.mad.2006.11.016.
- GALGANI, JOSE E, CEDRIC MORO, and ERIC RAVUSSIN (2008). "Metabolic flexibility and insulin resistance." In: *American journal of physiology. Endocrinology and metabolism* 295.5, E1009–E1017. ISSN: 0193-1849. DOI: 10.1152/ajpendo.90558.2008.
- GARAGNANI, PAOLO, CHIARA PIRAZZINI, CRISTINA GIULIANI, MARCO CANDOLA, PATRIZIA BRIGIDI, FEDERICA SEVINI, DONATA LUISELLI, MARIA GIULIA BACALINI, STEFANO SALVIOLI, MIRIAM CAPRI, et al. (2014). "The three genetics (Nuclear DNA, mitochondrial DNA, and Gut microbiome) of longevity in humans considered as metaorganisms". In: *BioMed Research International* 2014. ISSN: 23146141. DOI: 10.1155/2014/560340.
- GIARDINE, BELINDA, CATHY RIEMER, ROSS C. HARDISON, RICHARD BURHANS, LAURA ELNITSKI, PRACHI SHAH, YI ZHANG, DANIEL BLANKENBERG, ISTVAN ALBERT, JAMES TAYLOR, et al. (2005). "Galaxy: A platform for interactive large-scale genome analysis". In: *Genome Research* 15.10, pp. 1451–1455. ISSN: 10889051. DOI: 10.1101/gr.4086505.
- GOECKS, JEREMY, ANTON NEKRUTENKO, and JAMES TAYLOR (2010). "Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences." In: *Genome biology* 11.8, R86. ISSN: 1465-6906. DOI: 10.1186/gb-2010-11-8-r86.
- GORDON, DAVID, CHRIS ABAJIAN, and PHIL GREEN (1998). "Consed: A graphical tool for sequence finishing". In: *Genome Research* 8.3, pp. 195–202. ISSN: 10889051. DOI: 10.1101/gr.8.3.195.
- GOUDEY, BENJAMIN, DAVID RAWLINSON, QIAO WANG, FAN SHI, HERMAN FERRA, RICHARD M CAMPBELL, LINDA STERN, MICHAEL T INOUE, CHENG SOON ONG, and ADAM KOWALCZYK (2013). "GWIS—model-free, fast and exhaustive search for epistatic interactions in case-control GWAS." In: *BMC genomics* 14 Suppl 3, S10. ISSN: 1471-2164. DOI: 10.1186/1471-2164-14-S3-S10.
- GUO, WEIMIN, EUNHEE KONG, and MOHSEN MEYDANI (2009). "Dietary polyphenols, inflammation, and cancer." In: *Nutrition and cancer* 61.6, pp. 807–810. ISSN: 0163-5581. DOI: 10.1080/01635580903285098.
- HAZKANI-COVO, EINAT, RAYMOND M. ZELLER, and WILLIAM MARTIN (2010). "Molecular poltergeists: Mitochondrial DNA copies (numts) in sequenced nuclear genomes". In: *PLoS Genetics* 6. ISSN: 15537390. DOI: 10.1371/journal.pgen.1000834.
- HENDRICKSON, SHER L, HOLLI B HUTCHESON, EDUARDO RUIZ-PESINI, JASON C POOLE, JAMES LAUTENBERGER, EFE SEZGIN, LAWRENCE KINGSLEY, JAMES J GOEDERT, DAVID VLAHOV, SHARYNE DONFIELD, et al. (2008). "Mitochondrial DNA haplogroups influence AIDS progression." In: *AIDS (London, England)* 22, pp. 2429–2439. ISSN: 0269-9370. DOI: 10.1097/QAD.0b013e32831940bb.
- HOTAMISLIGIL, GÖKHAN S and EBRU ERBAY (2008). "Nutrient sensing and inflammation in metabolic diseases." In: *Nature reviews. Immunology* 8.12, pp. 923–934. ISSN: 1474-1733. DOI: 10.1038/nri2449.
- HUNTER, JOHN D. (2007). "Matplotlib: A 2D graphics environment". In: *Computing in Science and Engineering* 9.3, pp. 99–104. ISSN: 15219615. DOI: 10.1109/MCSE.2007.55.
- JEFFERY, IAN B. and PAUL W. O'TOOLE (2013). "Diet-microbiota interactions and their implications for healthy living". In: 5.1, pp. 234–252. ISSN: 20726643. DOI: 10.3390/nu5010234.

- JIA, JUN JING, YUN BO TIAN, ZHEN HUI CAO, LIN LI TAO, XI ZHANG, SI ZHEN GAO, CHANG RONG GE, QIU YE LIN, and M. JOIS (2010). "The polymorphisms of UCP1 genes associated with fat metabolism, obesity and diabetes". In: *Molecular Biology Reports* 37.3, pp. 1513–1522. ISSN: 03014851. DOI: 10.1007/s11033-009-9550-2.
- JIA, WENLI and PAUL G. HIGGS (2008). "Codon Usage in Mitochondrial Genomes: Distinguishing Context-Dependent Mutation from Translational Selection". In: *Molecular Biology and Evolution* 25.2, pp. 339–351. DOI: 10.1093/molbev/msm259.
- KASS, RE ROBERT E. and ADRIAN E. AE RAFTERY (1995). "Bayes factors". In: *Journal of the American Statistical Association* ... 90.430, pp. 773–795.
- KENNEDY, SCOTT R., JESSE J. SALK, MICHAEL W. SCHMITT, and LAWRENCE A. LOEB (2013). "Ultra-Sensitive Sequencing Reveals an Age-Related Increase in Somatic Mitochondrial Mutations That Are Inconsistent with Oxidative Damage". In: *PLoS Genetics* 9.9, e1003794. ISSN: 15537390. DOI: 10.1371/journal.pgen.1003794.
- KENYON, L and C T MORAES (1997). "Expanding the functional human mitochondrial DNA database by the establishment of primate xenomito-chondrial cybrids." In: *Proceedings of the National Academy of Sciences of the United States of America* 94.17, pp. 9131–9135. ISSN: 0027-8424. DOI: 10.1073/pnas.94.17.9131.
- KLUGER, JEFFREY (2008). "A brief history of: cloning." In: *Time* 172.2, p. 22. ISSN: 0040781X.
- KUJOTH, GREGORY C., CHRISTIAAN LEEUWENBURGH, and TOMAS A. PROLLA (2006). "Mitochondrial DNA mutations and apoptosis in mammalian aging". In: *Cancer Research* 66, pp. 7386–7389. ISSN: 00085472. DOI: 10.1158/0008-5472.CAN-05-4670.
- LAGOUGE, M and N-G LARSSON (2013). "The role of mitochondrial DNA mutations and free radicals in disease and ageing." In: *Journal of internal medicine* 273.6, pp. 529–43. ISSN: 1365-2796. DOI: 10.1111/joim.12055.
- LARKIN, M. A., G. BLACKSHIELDS, N. P. BROWN, R. CHENNA, P. A. MCGETTIGAN, H. MCWILLIAM, F. VALENTIN, I. M. WALLACE, A. WILM, R. LOPEZ, et al. (2007). "Clustal W and Clustal X version 2.0". In: *Bioinformatics* 23.21, pp. 2947–2948. ISSN: 13674803. DOI: 10.1093/bioinformatics/btm404.
- LI, HENG and RICHARD DURBIN (2009a). "Fast and accurate short read alignment with Burrows-Wheeler transform". In: *Bioinformatics* 25.14, pp. 1754–1760. ISSN: 13674803. DOI: 10.1093/bioinformatics/btp324.
- LI, HENG, BOB HANDSAKER, ALEC WYSOKER, TIM FENNEL, JUE RUAN, NILS HOMER, GABOR MARTH, GONCALO ABECASIS, and RICHARD DURBIN (2009b). "The Sequence Alignment/Map format and SAMtools". In: *Bioinformatics* 25.16, pp. 2078–2079. ISSN: 13674803. DOI: 10.1093/bioinformatics/btp352.
- LI, MINGKUN, ROLAND SCHROEDER, ALBERT KO, and MARK STONEKING (2012). "Fidelity of capture-enrichment for mtDNA genome sequencing: Influence of NUMTs". In: *Nucleic Acids Research* 40.18, e137. ISSN: 03051048. DOI: 10.1093/nar/gks499.
- LUNG, BIRGIT, ANJA ZEMANN, MONIKA J. MADEJ, MARKUS SCHUELKE, SANDRA TECHRITZ, STEPHANIE RUF, RALPH BOCK, and ALEXANDER HÜTTENHOFER (2006). "Identification of small non-coding RNAs from mitochondria and chloroplasts". In: *Nucleic Acids Research* 34.14, pp. 3842–3852. ISSN: 03051048. DOI: 10.1093/nar/gkl448.
- MCKENNA, AARON, MATTHEW HANNA, ERIC BANKS, ANDREY SIVACHENKO, KRISTIAN CIBULSKIS, ANDREW KERNYTSKY, KIRAN GARIMELLA, DAVID ALTSHULER, STACEY GABRIEL, MARK DALY, et al. (2010). "The genome analysis toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data". In: *Genome Research* 20.9, pp. 1297–1303. ISSN: 10889051. DOI: 10.1101/gr.107524.110.

- McKINNEY, WES (2010). "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by STÉFAN VAN DER WALT and JARROD MILLMAN, pp. 51–56.
- MEABURN, EMMA and REINER SCHULZ (2012). "Next generation sequencing in epigenetics: Insights and challenges". In: *Seminars in Cell and Developmental Biology* 23.2, pp. 192–199. ISSN: 10849521. DOI: 10.1016/j.semcdb.2011.10.010.
- MENICHETTI, GIULIA, DANIEL REMONDINI, PIETRO PANZARASA, RAÚL J. MONDRAGÓN, and GINESTRA BIANCONI (2014). "Weighted multiplex networks". In: *PLoS ONE* 9.6, e97857. ISSN: 19326203. DOI: 10.1371/journal.pone.0097857.
- MERRIMAN, BARRY, ION TORRENT, and JONATHAN M. ROTHBERG (2012). "Progress in Ion Torrent semiconductor chip based sequencing". In: *Electrophoresis* 33.23, pp. 3397–3417. ISSN: 01730835. DOI: 10.1002/elps.201200424.
- MOORE, JASON H (2015). "Epistasis Analysis Using ReliefF". In: *Epistasis*. Ed. by JASON H MOORE and SCOTT M WILLIAMS. Vol. 1253. *Methods in Molecular Biology*. Springer New York, pp. 315–325. ISBN: 978-1-4939-2154-6. DOI: 10.1007/978-1-4939-2155-3_17.
- MOORE, JASON H, RYAN AMOS, JEFF KIRALIS, and PETER C ANDREWS (2014). "Heuristic Identification of Biological Architectures for Simulating Complex Hierarchical Genetic Interactions Genetic Epidemiology". In: *Genetic Epidemiology* 39.1, pp. 25–34. ISSN: 1098-2272. DOI: 10.1002/gepi.21865.
- NELSON, ELIZABETH K, BRITT PIEHLER, JOSH ECKELS, ADAM RAUCH, MATTHEW BELLEW, PETER HUSSEY, SARAH RAMSAY, CORY NATHE, KARL LUM, KEVIN KROUSE, et al. (2011). "LabKey Server: an open source platform for scientific data integration, analysis and collaboration." In: *BMC bioinformatics* 12.1, p. 71. ISSN: 1471-2105. DOI: 10.1186/1471-2105-12-71.
- NIEMI, ANNA KISA, ANTTI HERVONEN, MIKKO HURME, PEKKA J. KARHUNEN, MARJA JYLHÄ, and KARI MAJAMAA (2003). "Mitochondrial DNA polymorphisms associated with longevity in a Finnish population". In: *Human Genetics* 112.1, pp. 29–33. ISSN: 03406717. DOI: 10.1007/s00439-002-0843-y.
- NUNN, ALISTAIR VW, JIMMY D BELL, and GEOFFREY W GUY (2009). "Lifestyle-induced metabolic inflexibility and accelerated ageing syndrome: insulin resistance, friend or foe?" In: *Nutrition & metabolism* 6.1, p. 16. ISSN: 1743-7075. DOI: 10.1186/1743-7075-6-16.
- O'DEA, ELLEN L, DERREN BARKEN, RAECHEL Q PERALTA, KIM T TRAN, SHANNON L WERNER, JEFFREY D KEARNS, ANDRE LEVCHENKO, and ALEXANDER HOFFMANN (2007). "A homeostatic model of IkappaB metabolism to control constitutive NF-kappaB activity." In: *Molecular systems biology* 3.1, p. 111. ISSN: 1744-4292. DOI: 10.1038/msb4100148.
- PÉREZ, FERNANDO and BRIAN E. GRANGER (2007). "IPython: A system for interactive scientific computing". In: *Computing in Science and Engineering* 9.3, pp. 21–29. ISSN: 15219615. DOI: 10.1109/MCSE.2007.53.
- PICARDI, ERNESTO and GRAZIANO PESOLE (2012). "Mitochondrial genomes gleaned from human whole-exome sequencing". In: *Nature Methods* 9.6, pp. 523–524. ISSN: 1548-7091. DOI: 10.1038/nmeth.2029.
- PONUGOTI, BHASKAR, GUANGYU DONG, and DANA T. GRAVES (2012). "Role of forkhead transcription factors in diabetes-induced oxidative stress". In: *Experimental Diabetes Research* 2012, p. 939751. ISSN: 16875214. DOI: 10.1155/2012/939751.
- PURCELL, SHAUN, BENJAMIN NEALE, KATHE TODD-BROWN, LORI THOMAS, MANUEL A R FERREIRA, DAVID BENDER, JULIAN MALLER, PAMELA SKLAR, PAUL I W DE BAKKER, MARK J DALY, et al. (2007). "PLINK: a tool set for whole-genome association and population-based linkage analyses." In: *American journal of human genetics* 81.3, pp. 559–575. ISSN: 00029297. DOI: 10.1086/519795.
- PÜTZ, B., T. KAM-THONG, N. KARBALAI, A. ALTMANN, and B. MÜLLER-MYHSOK (2013). "Cost-effective GPU-grid for genome-wide epistasis calculations".

- In: *Methods of Information in Medicine* 52.1, pp. 91–95. ISSN: 00261270. DOI: 10.3414/ME11-02-0049.
- QUINLAN, AARON R. and IRA M. HALL (2010). “BEDTools: A flexible suite of utilities for comparing genomic features”. In: *Bioinformatics* 26.6, pp. 841–842. ISSN: 13674803. DOI: 10.1093/bioinformatics/btq033.
- R FOUNDATION FOR STATISTICAL COMPUTING, VIENNA, AUSTRIA (2008). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria.
- RACKHAM, O., A.-M. J. SHEARWOOD, T. R. MERCER, S. M. K. DAVIES, J. S. MATTICK, and A. FILIPOVSKA (2011). “Long noncoding RNAs are generated from the mitochondrial genome and regulated by nuclear-encoded proteins”. In: *Rna* 17.12, pp. 2085–2093. ISSN: 1355-8382. DOI: 10.1261/rna.029405.111.
- RAND, D M, R A HANEY, and A J FRY (2004). “Cytonuclear cooperation: the genomics of cooperation”. In: *Trends in Ecology & Evolution* 19.12, pp. 645–653. ISSN: 0169-5347. DOI: <http://dx.doi.org/10.1016/j.tree.2004.10.003>.
- RATTAN, SURESH IS (2014). “Aging is not a disease: implications for intervention”. In: *Aging and Disease* 5.3, p. 196. DOI: 10.14336/AD.2014.0500196.
- RAULE, NICOLA, FEDERICA SEVINI, SHENGTING LI, ANNALaura BARBIERI, FEDERICA TALLARO, LAURA LOMARTIRE, DARIO VIANELLO, ALBERTO MONTE-SANTO, JUKKA S. MOILANEN, VLADYSLAV BEZRUKOV, et al. (2014). “The co-occurrence of mtDNA mutations on different oxidative phosphorylation subunits, not detected by haplogroup analysis, affects human longevity and is population specific”. In: *Aging Cell* 13.3, pp. 401–407. ISSN: 14749726. DOI: 10.1111/ace1.12186.
- REARDON, W. (2002). “Emery and Rimoin’s Principles and Practice of Medical Genetics”. In: *Journal of medical genetics* 39, p. 454. DOI: 10.1136/jmg.39.6.454-a.
- RIBARIČ, SAMO (2012). “Diet and aging”. In: *Oxidative Medicine and Cellular Longevity* 2012, p. 741468. ISSN: 19420900. DOI: 10.1155/2012/741468.
- RITCHIE, MARYLYN D (2015). “Finding the Epistasis Needles in the Genome-Wide Haystack”. In: *Epistasis*. Ed. by JASON H MOORE and SCOTT M WILLIAMS. Vol. 1253. Methods in Molecular Biology. Springer New York, pp. 19–33. ISBN: 978-1-4939-2154-6. DOI: 10.1007/978-1-4939-2155-3_2.
- SALAS, ANTONIO, ÁNGEL CARRACEDO, VINCENT MACAULAY, MARTIN RICHARDS, and HANS JÜRGEN BANDELT (2005). “A practical guide to mitochondrial DNA error prevention in clinical, forensic, and population genetics”. In: *Biochemical and Biophysical Research Communications* 335.3, pp. 891–899. ISSN: 0006291X. DOI: 10.1016/j.bbrc.2005.07.161.
- SALVIOLI, STEFANO, MIRIAM CAPRI, AURELIA SANTORO, NICOLA RAULE, FEDERICA SEVINI, STELLA LUKAS, CATIA LANZARINI, DANIELA MONTI, GIUSEPPE PASSARINO, GIUSEPPINA ROSE, et al. (2008). “The impact of mitochondrial DNA on human lifespan: A view from studies on centenarians”. In: *Biotechnology Journal* 3, pp. 740–749. ISSN: 18606768. DOI: 10.1002/biot.200800046.
- SALVIOLI, STEFANO, DANIELA MONTI, CATIA LANZARINI, MARIA CONTE, CHIARA PIRAZZINI, MARIA GIULIA BACALINI, PAOLO GARAGNANI, CRISTINA GIULIANI, ELISA FONTANESI, RITA OSTAN, et al. (2013). “Immune system, cell senescence, aging and longevity - inflamm-aging reappraised.” In: *Current pharmaceutical design* 19.9, pp. 1675–9. ISSN: 1873-4286. DOI: 10.2174/1381612811319090015.
- SANTORO, AURELIA, VALENTINA BALBI, ELISA BALDUCCI, CHIARA PIRAZZINI, FRANCESCA ROSINI, FRANCESCA TAVANO, ALESSANDRO ACHILLI, PAOLA SIVIERO, NADIA MINICUCI, ELENA BELLAVISTA, et al. (2010). “Evidence for sub-haplogroup H5 of mitochondrial DNA as a risk factor for late onset alzheimer’s disease”. In: *PLoS ONE* 5. ISSN: 19326203. DOI: 10.1371/journal.pone.0012037.
- SANTORO, AURELIA, BRIGIDI PATRIZIA, S GONOS EFSTATHIOS, A BOHR VILHELM, and FRANCESCHI CLAUDIO (2014). “Mediterranean Diet and Inflamm-

- maging in the elderly — The European project NU-AGE". In: *Mechanisms of Ageing and Development* 136–137, pp. 1–2. ISSN: 0047-6374. DOI: <http://dx.doi.org/10.1016/j.mad.2014.01.006>.
- SHANNON, C. E. (2001). "A mathematical theory of communication". In: *ACM SIGMOBILE Mobile Computing and Communications Review* 5, p. 3. ISSN: 15591662. DOI: 10.1145/584091.584093.
- SHERRY, S T, M H WARD, M KHOLODOV, J BAKER, L PHAN, E M SMIGIELSKI, and K SIROTKIN (2001). "dbSNP: the NCBI database of genetic variation." In: *Nucleic acids research* 29.1, pp. 308–311. ISSN: 1362-4962. DOI: 10.1093/nar/29.1.308.
- SHOKOLENKO, INNA N, GLENN L WILSON, and MIKHAIL F ALEXEYEV (2014). "Aging: A mitochondrial DNA perspective, critical analysis and an update". In: *World journal of experimental medicine* 4.4, p. 46.
- SIEVERS, FABIAN, ANDREAS WILM, DAVID DINEEN, TOBY J GIBSON, KEVIN KARPLUS, WEIZHONG LI, RODRIGO LOPEZ, HAMISH MCWILLIAM, MICHAEL REMMERT, JOHANNES SÖDING, et al. (2011). "Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega". In: *Molecular Systems Biology* 7.1. ISSN: 1744-4292. DOI: 10.1038/msb.2011.75.
- SKYTTHE, A., S. VALENSIN, B. JEUNE, E. CEVENINI, F. BALARD, M. BEEKMAN, V. BEZRUKOV, H. BLANCHE, L. BOLUND, K. BROZCEK, et al. (2011). "Design, recruitment, logistics, and data management of the GEHA (Genetics of Healthy Ageing) project". In: *Experimental Gerontology* 46.11, pp. 934–945. ISSN: 05315565. DOI: 10.1016/j.exger.2011.08.005.
- STONEKING, MARK (2000). "Hypervariable Sites in the mtDNA Control Region Are Mutational Hotspots". In: *American Journal of Human Genetics* 67.4, pp. 1029–1032.
- SUGIHARA, GEORGE, ROBERT MAY, HAO YE, CHIH-HAO HSIEH, ETHAN DEYLE, MICHAEL FOGARTY, and STEPHAN MUNCH (2012). "Detecting Causality in Complex Ecosystems". In: *Science* 338.6106, pp. 496–500. DOI: 10.1126/science.1227079.
- TATS, AGE, TANEL TENSON, and MAIDO REMM (2008). "Preferred and avoided codon pairs in three domains of life". In: *BMC Genomics* 9.1, p. 463. ISSN: 1471-2164. DOI: 10.1186/1471-2164-9-463.
- TIERI, PAOLO, ALBERTO TERMANINI, ELENA BELLAVISTA, STEFANO SALVIOLI, MIRIAM CAPRI, and CLAUDIO FRANCESCHI (2012). "Charting the NF- κ B pathway interactome map". In: *PLoS ONE* 7.3, e32678. ISSN: 19326203. DOI: 10.1371/journal.pone.0032678.
- TORNATORE, L, A K THOTAKURA, J BENNETT, M MORETTI, and G FRANZOSO (2012). "The nuclear factor κ B signaling pathway: integrating metabolism with inflammation". In: *Trends Cell Biol* 22.11, pp. 557–566. DOI: S0962-8924(12)00140-7 [pii] \$backslash\$r10.1016/j.tcb.2012.08.001.
- TRANA, GREGORY J (2011). "Mitochondrial – nuclear epistasis : Implications for human aging and longevity". In: *Ageing Research Reviews* 10.2, pp. 238–252. ISSN: 1568-1637. DOI: 10.1016/j.arr.2010.06.003.
- TRIFUNOVIC, ALEKSANDRA, ANNA WREDENBERG, MARIA FALKENBERG, JOHANNES N. SPELBRINK, ANJA T. ROVIO, CARL E. BRUDER, MOHAMMAD BOHLOOLY-Y, SEBASTIAN GIDLOF, ANDERS OLDFORS, ROLF WIBOM, et al. (2004). "Premature ageing in mice expressing defective mitochondrial DNA polymerase". In: *Nature* 429.6990, pp. 417–423. DOI: 10.1038/nature02517.
- VAN DER AUWERA, GERALDINE A., MAURICIO O. CARNEIRO, CHRISTOPHER HARTL, RYAN POPLIN, GUILLERMO DEL ANGEL, AMI LEVY-MOONSHINE, TADEUSZ JORDAN, KHALID SHAKIR, DAVID ROAZEN, JOEL THIBAUT, et al. (2013). "From fastQ data to high-confidence variant calls: The genome analysis toolkit best practices pipeline". In: *Current Protocols in Bioinformatics*. John Wiley & Sons, Inc. ISBN: 0471250953. DOI: 10.1002/0471250953.bi1110s43.
- VAN DER WALT, STÉFAN, S. CHRIS COLBERT, and GAËL VAROQUAUX (2011). "The NumPy array: A structure for efficient numerical computation". In:

- Computing in Science and Engineering* 13.2, pp. 22–30. ISSN: 15219615. DOI: 10.1109/MCSE.2011.37.
- VAN OMMEN, BEN, JILDAU BOUWMAN, LARS O. DRAGSTED, CHRISTIAN A. DREVON, RUAN ELLIOTT, PHILIP DE GROOT, JIM KAPUT, JOHN C. MATHERS, MICHAEL MÜLLER, FRE PEPPING, et al. (2010). “Challenges of molecular nutrition research 6: The nutritional phenotype database to store, share and evaluate nutritional systems biology studies”. In: *Genes and Nutrition* 5, pp. 189–203. ISSN: 15558932. DOI: 10.1007/s12263-010-0167-9.
- VAN OVEN, MANNIS and MANFRED KAYSER (2009). “Updated comprehensive phylogenetic tree of global human mitochondrial DNA variation.” In: *Human mutation* 30, pp. 386–394. ISSN: 10981004. DOI: 10.1002/humu.20921.
- VIANELLO, DARIO, FEDERICA SEVINI, GASTONE CASTELLANI, LAURA LOMARTIRE, MIRIAM CAPRI, and CLAUDIO FRANCESCHI (2013). “HAPLOFIND: A new method for high-throughput mtDNA haplogroup assignment”. In: *Human Mutation* 34, pp. 1189–1194. ISSN: 10597794. DOI: 10.1002/humu.22356.
- WALLACE, DOUGLAS C (2005). “A mitochondrial paradigm of metabolic and degenerative diseases, aging, and cancer: a dawn for evolutionary medicine.” In: *Annual review of genetics* 39, pp. 359–407. ISSN: 0066-4197. DOI: 10.1146/annurev.genet.39.110304.095751.
- (2007). “Why do we still have a maternally inherited mitochondrial DNA? Insights from evolutionary medicine.” In: *Annual review of biochemistry* 76, pp. 781–821. ISSN: 0066-4154. DOI: 10.1146/annurev.biochem.76.081205.150955.
- (2013). “Bioenergetics in human evolution and disease: implications for the origins of biological complexity and the missing genetic variation of common diseases.” In: *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 368.1622, p. 20120267. ISSN: 1471-2970. DOI: 10.1098/rstb.2012.0267.
- WASKOM, MICHAEL, OLGA BOTVINNIK, PAUL HOBSON, JOHN B COLE, YAROSLAV HALCHENKO, STEPHAN HOYER, ALISTAIR MILES, TOM AUGSPURGER, TAL YARKONI, TOBIAS MEGIES, et al. (2014). “seaborn: v0.5.0 (November 2014)”. In: DOI: 10.5281/zenodo.12710.
- WU, MICHAEL C., SEUNGGEUN LEE, TIANXI CAI, YUN LI, MICHAEL BOEHNKE, and XIHONG LIN (2011). “Rare-variant association testing for sequencing data with the sequence kernel association test”. In: *American Journal of Human Genetics* 89.1, pp. 82–93. ISSN: 00029297. DOI: 10.1016/j.ajhg.2011.05.029.
- YANG, JENQ-LIN, LIOR WEISSMAN, VILHELM A BOHR, and MARK P MATTSON (2008). “Mitochondrial DNA damage and repair in neurodegenerative disorders.” In: *DNA repair* 7, pp. 1110–1120. ISSN: 15687864. DOI: 10.1016/j.dnarep.2008.03.012.
- YATES, ANDREW, KATHRYN BEAL, STEPHEN KEENAN, WILLIAM McLAREN, MIGUEL PIGNATELLI, GRAHAM R S RITCHIE, MAGALI RUFFIER, KIERON TAYLOR, ALESSANDRO VULLO, and PAUL FLICEK (2014). “The Ensembl REST API: Ensembl Data for Any Language”. In: *Bioinformatics*. DOI: 10.1093/bioinformatics/btu613.

WEB BIBLIOGRAPHY

- Python Website. *Python programming language - Official Website*. [Online; accessed 2014-12-04]. URL: <http://www.python.org>.
- pypy Website. *PyPy - Official Website*. [Online; accessed 2014-11-25]. URL: <http://pypy.org/>.
- Django Website. *Django Project - Official Website*. [Online; accessed 2014-12-4]. URL: <http://www.djangoproject.com>.
- SciPy Website. *SciPy - Official Website*. [Online; accessed 2014-11-25]. URL: <http://www.scipy.org/>.

Basemap Website. *Basemap Website*. [Online; accessed 2014-12-17]. URL: <https://github.com/matplotlib/basemap>.

scrapy Website. *Scrapy - Official Website*. [Online; accessed 2014-12-04]. URL: <http://scrapy.org/>.

Plumbum Website. *Plumbum Website*. [Online; accessed 2015-01-17]. URL: <https://plumbum.readthedocs.org/>.

SQLAlchemy Website. *SQLAlchemy Website*. [Online; accessed 2014-12-8]. URL: <http://www.sqlalchemy.org/>.

CRAN Website. *CRAN Website*. [Online; accessed 2014-12-18]. URL: <http://cran.r-project.org/>.

Bioconductor Website. *Bioconductor Website*. [Online; accessed 2014-12-18]. URL: <http://master.bioconductor.org/>.

VCF Format Specification website. *VCF Specification*. [Online; accessed 2014-12-6]. URL: <http://samtools.github.io/hts-specs/VCFv4.2.pdf>.

PostgreSQL Website. *PostgreSQL Website*. [Online; accessed 2014-12-11]. URL: <http://www.postgresql.org/>.

FastQC Website. *FastQC Website*. [Online; accessed 2014-12-13]. URL: <http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>.

Picard Website. *Picard Website*. [Online; accessed 2014-12-18]. URL: <http://broadinstitute.github.io/picard/>.

EpiData Entry Website. *EpiData Entry Website*. [Online; accessed 2014-12-28]. URL: <http://cran.r-project.org/>.

Git Website. *Git Website*. [Online; accessed 2014-12-31]. URL: <http://git-scm.com/>.

YAML Website. *YAML Website*. [Online; accessed 2015-01-04]. URL: <http://www.yaml.org/>.

ReportLab Website. *ReportLab Website*. [Online; accessed 2015-01-04]. URL: <http://www.reportlab.com/>.

openpyxl Website. *openpyxl Website*. [Online; accessed 2015-01-04]. URL: <https://openpyxl.readthedocs.org/>.

ACKNOWLEDGEMENTS

Sorry for any english reader, but I'm moving to Italian for the acknowledgements, also because I don't really think I will have english readers in this section, after all!

Com'è giusto che sia, questo condensato di tre anni di vita va pensato come l'estratto del lavoro di molte persone, non di una sola mente. Può essere che "lavoro" significhi solo una pacca sulla spalla, o una risata, nel momento giusto, ma nessuno sa come sarebbe andata senza quel "lavoro". Insieme, abbiamo superato montagne che mai avrei pensato anche solo di vedere.

Per ormai la terza volta – e si spera l'ultima, altre tesi, se Dio vuole, non ci saranno – il primo posto va al Professor Franceschi, che ormai non so più nemmeno io dove tragga tutta la fiducia che mi dimostra, giorno dopo giorno. Il Prof. Salvioli, che si è prodigato a leggere questa tesi e a piazzare lì una serie di suggerimenti uno più tagliente dell'altro che sicuramente ne hanno aumentato la qualità. Di certo, ricorderò le nostre sghignazzate di qualità.

Aurelia, con cui abbiamo condiviso parte delle magne sfortune di NU-AGE – compreso attraversare di notte uno poco accogliente mare greco – e con cui siamo riusciti, spero, a guidare il progetto lontano dalle acque piene di scogli di un data management mal pensato e mal implementato. Ma questo, solo il tempo lo dirà. Federica S., che si è dimostrata ancora una volta una più che efficace guida nel mondo del mitocondrio e del suo piccolo, ma estremamente complesso, genoma. Ad entrambe va anche un grazie per la voglia con cui hanno emendato da questa tesi tutte le cretinaggini che il mio neurone solitario ha partorito. A proposito di partorire! Aure, Numa come sta? :-)

Enrico, il "nostro" caro fisico-statistico-programmatore-problemsolver, a cui ho fatto sputare un bel po' di sangue per tradurre la teoria dell'epistasi in qualcosa che potesse anche essere lontanamente matematicamente rigoroso. Per questo, e molto altro, grazie. Il viaggio non sarebbe stato lo stesso senza di lui, e probabilmente sarei finito in qualche gorgo molto prima della fine. Il Prof. Castellani – che nonostante io non sia un fisico e quindi per definizione non capisca nulla – riesce sempre a far sentire la sua benevola presenza.

Cristina, Claudia, Nicoletta, Barbara, Grazia, Vincenzo, e tutti quelli/e che sono stati a contatto con me in questi lunghi 3 anni per più di 10 minuti al giorno, beh, grazie per la pazienza, e per esservi subiti i miei più che frequenti borbottii (e non solo borbottii) rabbiosi. Laura, che ormai da un po' non fa più parte della partita, perché è stata un'ottima guida nei primi mesi. Elisa, a.k.a. "Fontanella", "Fountain" e via discorrendo, per tutta la carica di simpatia e vitalità che ogni volta porta con sé dagli Appennini. Ci manchi, spero tu lo sappia. Chiara, e tutte le risate che ci siamo fatti, con la speranza di scavarci presto con sangue, unghie e mente un futuro in questo complesso mondo, o almeno di riuscire a provarci, giusto LP? Sì LP, un posticino c'è anche per te, ovviamente!

Menzione d'onore ai miei genitori – per i quali mi sono già fregato le due migliori immagini nei ringraziamenti della tesi magistrale, e di nuove onestamente non ne ho – che sempre e comunque hanno continuato a sopportarmi e supportarmi. Dopo 3 anni, però, devo ancora capire quale sia l'intrinseca opportunità che mia madre si ostina a farmi cogliere a son di urla in una maglietta buttata sulla sedia in camera. Tenta ogni volta di spiegarmelo, ma ormai temo non ci arriverò mai. Mio padre, che ogni volta che può, corre in mio aiuto. Spesso, corre per cercare di distogliere mia madre dallo spiegarmi la faccenda della maglietta! Chi li capisce, dopotutto! Mah!

C'è una stella che brilla ormai da molti anni nella mia vita, e che ancora oggi illumina il mio cammino per non farmi perdere la strada. Federica, grazie di tutto quello che hai fatto e fai ogni giorno.

Zio Tullio e Zia Adelma, nonostante quanto promesso, temo di essere di nuovo colpevole di una cronica mancanza di più che dovute attenzioni nei vostri confronti. Mi dispiace. Mi trovo spesso a ripensare a quel gioioso e giocoso passato, e ogni volta una gran malinconia mi attanaglia per ore.

Valter, who actually didn't invent any new definition in the last 3 years on the "strange things" I'm doing – pretty strange honestly, he already invented all of them, I suppose –, but still tries to lock me in all the hardest manual works I've seen until now. To be honest, I must say that my ability in avoiding them greatly improved, so the vast majority of the times he must do them on his own. Shame on me, I suppose. Donatella, che nonostante io faccia parte della serie B, trova sempre il tempo e il modo di farmi sentire a casa.

Un posto, importante, va dato anche agli amici storici: Grande Orso in testa, nonostante sia emigrato in terre molto più fredde. Walter, Matteo e Alberto, grazie per esserci stati anche questa volta, e grazie per quei favolosi kebab! Un grazie anche a Vinicio, Totò e Stefano, che a loro modo sono stati miei compagni in quest'avventura.

La mia seconda famiglia – Massimiliano, Michela, la Luci, Diego, il Nonno Olimpico ("Are Vianeo", come sempre), Gabriella, Michela, Claudio, e molti altri – è ancora lì. Ci vediamo meno, ma li porto nella mente e nel cuore ogni giorno, assieme a tutte le sfide che abbiamo affrontato e vinto insieme. Gianluca, finalmente sono riuscito a farti fare qualcosa nella vita, e tecnico sei diventato! Promessa mantenuta. M&M, un grosso in bocca al lupo per il fantastico viaggio in cui state per lanciarmi.

Mia nonna, la nonna che sempre mi suggeriva le tabelline di nascosto, ci ha nel frattempo lasciati. Sono sicuro, però, che assieme a Mansueto, Adele e Bepi e Giorgia è lassù che se la ride, fra una partita a carte e una bottiglia per festeggiare quest'ennesimo traguardo.

Ci sarebbe ancora tanta gente da nominare. Probabilmente troppa. Mi spiace ragazzi e ragazze, qui non c'è proprio più posto, ma statene certi, non è sulla carta che si ricordano certe cose, ma altrove.

Grazie a tutti!
Vianello D.

