

Analysis of Genetic Polymorphisms for Statistical Genomics: Tools and Applications

Carlos Morcillo Suárez

TESI DOCTORAL UPF / 2011

DIRECTOR DE LA TESI

Dr. Arcadi Navarro i Cuartiellas

DEPARTAMENT DE CIÈNCIES EXPERIMENTALS I DE LA
SALUT

a un señor pequeñito

Acknowledgments

Siempre me ha parecido muy raro que las tesis tengan una sección titulada "*agradecimientos*" y no tengan otra titulada "*reproches*". Con el tiempo he aprendido (y las canas son testigo de que tiempo he tenido) que las capacidades sociales del común de los mortales no se acaban donde se acaban las mías y ellos saben leer reproches en matices y detalles de los agradecimientos. Yo no.

A Arcadi le reprocho que me sedujera con patrañas y falsas promesas para que dejara IBM y me viniera a la UPF. Que se aprovechara de nuestra amistad para hacerme creer que hacer un doctorado podría ser una experiencia interesante y que me metiera en el lío de la ciencia que me ha hecho tener que aprender cosas nuevas cada día como si fuera un crío pequeño. ¿No hemos quedado ya que estoy un poco madurito para estas cosas?

Sin Arcadi esta tesis no se hubiera podido perpetrar. Pero no es el único culpable, no. Y ya que me he puesto a tirar de la manta, pues vamos a contar las verdades.

Si el equipo de gente que participó en la creación de SNPator hubiera colaborado un poco, podríamos haber reventado la aplicación y yo hubiera vuelto a IBM a tiempo para recuperar mi carrera de técnico de sistemas. Pero no, va y los tíos se dedican a trabajar mucho y bien! A Ángel Carreño, Txema Heredia y Ricard Sangrós les ha importado un pito mi futuro y se han dedicado con esfuerzo y talento a crear y mantener la aplicación. No se lo perdono.

El peor de todos es Pep Alegre. ¿Pero qué se ha creído ese tío? ¿Qué programar es un arte? ¿Que la perfección existe? En vez de programar con el chapuza++ como hacemos todos, este desalmado programó SNPator de tal manera que a uno le entra el síndrome de Stendhal cada vez que mira el código.

Ahora ya puedo pasar a los agradecimientos.

A Arcadi por la libertad que me ha dado.

A los compañeros, en la Pompeu y en IBM, por todas las veces que han dejado de hacer algo para echarme una mano.

A mi Kasia por seguir creyendo mis promesas en contra de la evidencia.

A mis padres por todo lo demás.

Abstract

New approaches are needed to manage and analyze the enormous quantity of biological data generated by modern technologies. Existing solutions are often fragmented and uncoordinated and, thus, they require considerable bioinformatics skills from users. Three applications have been developed illustrating different strategies to help users without extensive IT knowledge to take maximum profit from their data.

SNPator is an easy-to-use suite that integrates all the usual tools for genetic association studies: from initial quality control procedures to final statistical analysis. CHAVA is an interactive visual application for CNV calling from aCGH data. It presents data in a visual way that helps assessing the quality of the calling and assists in the process of optimization. Haplotype Association Pattern Analysis visually presents data from exhaustive genomic haplotype associations, so that users can recognize patterns of possible associations that cannot be detected by single-SNP tests.

Resum

Calen noves aproximacions per la gestió i anàlisi de les enormes quantitats de dades biològiques generades per les tecnologies modernes. Les solucions existents, sovint fragmentaries i descoordinades, requereixen elevats nivells de formació bioinformàtica. Hem desenvolupat tres aplicacions que il·lustren diferents estratègies per ajudar als usuaris no experts en informàtica a aprofitar al màxim les seves dades.

SNPator és un paquet de fàcil us que integra les eines usades habitualment en estudis de associació genètica: des del control de qualitat fins les anàlisi estadístiques. CHAVA és una aplicació visual interactiva per a la determinació de CNVs a partir de dades aCGH. Presenta les dades visualment per ajudar a valorar la qualitat de les CNV predites i ajudar a optimitzar-la. Haplotype Pattern Analysis presenta dades d'anàlisi d'associació haplotípica de forma visual per tal que els usuaris puguin reconèixer patrons de associacions que no es possible detectar amb tests de SNPs individuals.

Preface

The scientific world is living times of enormous changes due to the continuous development of new technologies with capabilities not imaginable a few years ago. Biomedical sciences are not an exception to that trend and new and more powerful technologies are increasing dramatically the amount of biological data generated and ready to be "*read and interpreted*" by the research community.

But tools and strategies of old do not work anymore. They quickly become obsolete as they are congested by the mere size of the bulk of data that scientists need to process. Innovative ways of handling, processing, reviewing and analyzing data are essential to take advantage of the technological progress that we are witnessing.

In this context, this thesis tries to contribute to this need proposing and developing some new approaches that are designed to make the life of scientists easier. Some of them are based on the integration in a single environment of data processing modules that are otherwise sparsely distributed and, thus, are difficult to coordinate. Others are based in taking advantage of the intuitive abilities of human brains to process visually huge amounts of data.

Index

Acknowledgments

Abstract

Preface

Index

1. Introduction.....	1
1.1 The Quest for the Roots of Human Variation.....	3
1.1.1 Genetic Polymorphisms	3
1.1.2 Genetic Mapping	4
1.1.3 Linkage Analysis	5
1.1.4 SNPs	8
1.1.5 Genetic Association	9
1.1.6 Genome-Wide Association Studies	11
1.1.7 CNVs	14
1.1.8 Epigenetic Polymorphisms.....	16
1.1.9 Ultrasequencing	17
1.2 The Technological Revolution.....	19
1.2.1 Moore's Law	19
1.2.2 Data Handling	21
1.2.3 Processing Data.....	22
1.2.4 Interpreting Data	23
1.2.4.1 Multiple Testing.....	23
1.2.4.2 Looking at the Data.....	24
1.2.4.3 New Approaches.....	28
1.3 Bioinformatics.....	30
1.4 Objectives.....	32
2. SNPator	35
2.1 Conception and Objectives	37
2.2 Web Application. Easy Access Everywhere.	39
2.3 Privacy Issues. Who can Access the Data.	43
2.4 Data Structure. Transparency.	44
2.5 Uploading Data.....	49
2.6 Quality Control. The Data Caring Module.....	50
2.7 Validation. Deciding the Ploidy.	55
2.8 The User Results Section.	57
2.9 Filters.....	58
2.9.1 The Boolean Interface.....	60
2.9.2 Creating the Filter.....	60
2.9.3 Marking the Data.....	62
2.9.4 Using the Filters	63

2.10 Batch Mode	63
2.11 Retrieving Data.....	65
2.12 Analyzing Data	66
2.13 The Calculating Machine.....	67
2.14 Computational Servers. Web Services	72
2.15 Massive Data Transfers	76
2.16 Pipeline Oriented Tasks. Connecting Functions	79
2.17 Development - Test - Production. Organizing the Work	80
2.18 System Management	82
2.19 SNPator Management.....	84
2.20 Implementation Effort.....	86
2.21 SNPator Use. Publication.....	86
3. CHAVA.....	89
3.1 The Problem.....	91
3.1.1 CGH	91
3.1.2 HMM.....	92
3.2. The Application.....	93
3.2.1 The Basic structure of the HMM	94
3.2.2 The Visual Element.....	96
3.2.3 Statistical Information.....	101
3.2.4 Array Structure	101
3.2.5 Command Line Mode.....	102
3.3 Use of CHAVA.....	103
3.3.1 Simulations by Evolutionary Algorithms.....	104
4. Haplotype Association Pattern Analysis	109
4.1 Introduction.....	111
4.2 Materials and Methods	113
4.2.1 Data preparation	113
4.2.2 Analysis and Visualization	114
4.3 Results	118
4.3.1 Comparison between Real and Random Datasets.....	123
4.3.2 Simulations.....	123
4.3.3 Classification of relevant regions	127
4.3.4 Literature search	130
4.3.5 LD patterns.....	130
4.4 Discussion	133
4.4.1 LD ,haplotype patterns and recombination	135
5. Discussion	139
Bibliography	147
Annex	

1. Introduction

1.1 The Quest for the Roots of Human Variation

1.1.1 Genetic Polymorphisms

If two haploid human genomes are selected at random, they will be almost identical. They will present differences in only approximately 1% of their span. A genetic polymorphism is defined as an element of the genetic material of a species which is not identical in all its individuals¹ but can present different variants with an abundance greater than a certain threshold (generally 1-5%). These variants are called alleles.

Genetic polymorphisms, to which I will refer in what follows simply as “polymorphisms”, are of great interest in several fields. In the case of evolution and population genetic studies, the types and distributions of polymorphisms among different populations and among different members of a population can give us insights about what regions of the genome have been experiencing selection and the characteristics of this selection. Events about the demographic history of a population can also be inferred studying patterns of polymorphism within a population, and comparing them between two or among many of them.

Polymorphisms are, also, partially responsible for the inheritance of differential phenotypic traits, in some cases explaining most of the population's variance in the trait. That makes them interesting because by mapping phenotypic variance to genomic polymorphisms we can get insights about the functions of the genes where these polymorphisms are located and about the general mechanisms of genetics. Also, certain polymorphisms can be responsible for (or influence) the development of pathologies in particular individuals. Identifying the polymorphisms associated with a disease may foster our understanding of the disease and the development of methods to prevent, alleviate or cure it. In theory, knowledge of the phenotypic correlates of genetic variability could allow to personalize treatments to every individual according to

¹ In diploid species we should take individual chromosomes instead of individuals for this definition.

their genotype. This has been labeled as "personalized medicine" (Donnelly 2008).

In order to achieve a full understanding of the mechanisms by which different alleles generate diverse phenotypes there are two previous tasks to do. On one hand, it is important to understand the nature of polymorphisms. That means defining how many types there are, their characteristics, their distribution in the genome and among different individuals. The other basic task is to try to localize, at least approximately, where in the genome are the elements responsible for the inheritance of different phenotypes. This second task is Genetic Mapping. Both tasks are strongly interrelated and have advanced in parallel. Some genetic mapping techniques, as we will see later, have had to wait for years until the knowledge about some kinds of genetic polymorphisms reached a level that was high enough to allow the techniques to be used in practice.

1.1.2 Genetic Mapping

Genetic mapping is the localization of genes underlying phenotypes on the basis of correlation with DNA variation, without the need for prior hypothesis about biological function (Altshuler et al. 2008).

Although widespread use of cheap and efficient whole genome sequencing will change this, until now, researchers could only test a fraction of the genome polymorphisms. Any genetic mapping technique is designed to take advantage of situations where those polymorphisms that can be tested, called genetic markers, harbor information about their genomic context. A statistical association between a marker and a phenotype indicates a genetic cause for the phenotype in some region around the marker.

There are two basic techniques of genetic mapping: linkage and association. Linkage analysis takes advantage of the fact that alleles of two loci segregate together with a probability that decreases with the distance between them. Association analysis takes advantage of the demographic history of our species that has left strong local correlations among close polymorphisms. For

instance, when Copy Number Variation (CNV) analysis finds a correlation of a CNV with a disease, it hints that the causal element is either inside or close to that CNV.

When the phenotypes of interest are related to health issues, genetic mapping is the basic tool of genetic epidemiology, which studies the genetic factors underlying health and disease and how they interact among them and with environmental factors.

1.1.3 Linkage Analysis

In a strict sense, linkage analysis is the study of the patterns of segregation of alleles of different loci. It is the tool that Sturtevant and Morgan used at the beginnings of the 20th century to create the first genetic maps (Griffiths et al. 1993). It is very powerful when working with model organisms that can be crossed at will.

In the context of modern human genetic epidemiology, linkage analysis takes the form of the study of one or several families which present some familiar disease. The health status of every member of the family is determined and a set of markers across the whole genome is genotyped for each individual. The rationale of the technique is to check whether, looking at the set of meiosis that took place in the pedigree, some marker seems to segregate together with the disease more often than should be expected by chance.

The likelihood of obtaining the pattern of meiotic recombination observed in the pedigree under a model with a certain amount of linkage between a marker and the disease status is calculated. The likelihood of obtaining it under a model of no linkage, that is, free recombination, is calculated too. The ratio of the first to the second, expressed as base 10 logarithm is the LOD score (Fig. 1.a) and is calculated for the whole genome at discrete intervals (Fig. 1.b).

To consider a region candidate to harboring the element responsible for the phenotype, a threshold value of $LOD \geq 3$ is generally required. This means that the likelihood under the hypothesis of linkage should be at least 1000 times greater than under the null hypothesis of no linkage between the marker and the disease-causing locus. This is to correct for the effect of the

recurrent testing of many markers. Since markers are not independent, a general effective number of tests is computed considering the average recombination rate of the genome.

a)

$$LOD = \log \left(\frac{P(X|c = \theta)}{P(X|c = 0.5)} \right)$$

b)

Parametric Analysis for Dominant_0.8

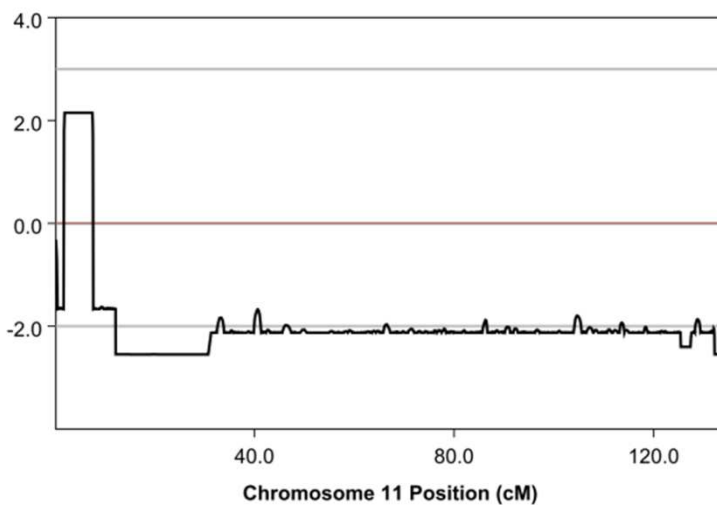


Fig. 1 a) Under the hypothesis of independence between marker and disease, the recombination fraction (θ) equals 0.5. Several discrete θ values smaller than 0.5 are checked to calculate LOD score in points around the marker. **b)** LOD score values obtained using MERLIN (Abecasis et al. 2002) for chromosome 11 considering a dominant model with penetrance = 0.8.

LOD Score calculations are simple when working with small, fully informative pedigrees. This is not the usual case. Actual pedigrees present missing genotypes, missing individuals or meiosis with unknown phase of marker and the disease. For every missing

piece of data, the likelihood calculation has to create a branch in the tree of probabilities. And every possible value of that missing data point has to be assigned prior probabilities. To avoid losing too much power, population allele frequencies and linkage disequilibrium values have to be taken into account when assigning those prior probabilities.

Several software applications have been developed to run linkage analysis calculations: GENEHUNTER (Kruglyak et al. 1996), LINKAGE (Lathrop et al. 1984), MERLIN (Abecasis et al. 2002) among many.

However, when pedigree complexity and incompleteness crosses a certain threshold, the evaluation of the complete probability tree becomes unfeasible and finding an exact solution has to be ruled out. Here modern heuristic approaches² (such as genetic algorithms, simulated annealing, etc) can help finding approximate solutions. As an example, SIMWALK2 (Sobel et al. 2001) uses Markov chain Monte Carlo and simulated annealing to work with high complex solution spaces in linkage analysis.

Although the origins of Linkage Mapping can be traced to almost a century ago, it could not be applied in a systematic way to the study of human variation and disease until the appearance of the first genetic maps in the end of the eighties (Donis-Keller et al. 1987). Armed with these sets of markers, easy to genotype, covering by linkage most of the human genome, research groups have mapped thousands of diseases over the last two decades. At July 2011, the public database Online Mendelian Inheritance in Man³ (OMIM,) records 3,208 Phenotype descriptions with known molecular basis.

Linkage analysis, as a genetic mapping technique, only points at a region of the genome (2 to 10 Mb long) and has to be complemented by other approaches (fine mapping, resequencing, etc) in order to ascertain the molecular mechanisms under the studied phenotype.

² Michalewicz, Z., et al. (2002). How to solve it : modern heuristics. Berlin ; New York, Springer.

³ [http:// www.ncbi.nlm.nih.gov/omim](http://www.ncbi.nlm.nih.gov/omim)

The high level of success of linkage analysis in the mapping of diseases with a clear Mendelian pattern of heredity was not replicated when trying to map complex diseases. These diseases, that have no clear modes of inheritance (*i.e.*, in contrast to Mendelian diseases they are not clearly dominant, or recessive or autosomal) and present strong environmental influence, are responsible for main health issues in developed societies, in terms of people affected and resources committed. This has fostered the development of association studies as a genetic mapping approach that, in principle, could be successfully applied to complex diseases.

In the same way that the practical development of linkage analysis depended on gaining basic knowledge on some highly polymorphic markers, especially Short Tandem Repeats (STRs), in the case of association the polymorphism of reference has been Single Nucleotide Polymorphisms (SNPs).

1.1.4 SNPs

SNPs are single base pair positions in genomic DNA at which different sequence alternatives (alleles) exist in normal individuals in some population(s), wherein the least frequent allele has an abundance of 1% or greater (Brookes 1999).

SNP's properties are very convenient for association studies. Although their strict definition allows for SNPs with 3 or 4 different alleles, in practice they are very rare (Brookes 1999) and the majority of them are binary. This has allowed the development of mechanized genotyping systems that have increased the productivity of the process and have lowered the price per genotype by several orders of magnitude (from 1\$ to 0.001\$) (Altshuler et al. 2008).

The number of SNPs is very high, there are over 13 million registered in the dbSNP public database⁴ and the number has been growing, until almost peaking recently with the publication of

⁴ <http://www.ncbi.nlm.nih.gov/SNP>

the results of the 1000 Genomes Project. They are distributed along the genome in a quite homogeneous way. It is almost always possible to find SNPs in any proposed target region.

The international consortium HapMap⁵ was launched to obtain a detailed description of SNP frequencies and their correlation across the genome in 270 samples from 3 continents. Around 4 million SNPs were genotyped in the phases I (International HapMap Consortium 2005) and II (Frazer et al. 2007) of the project and new populations were added in a third phase.

HapMap results confirmed that most common SNPs are correlated to SNPs around them and described and quantified those correlations. This non random distribution of alleles in nearby loci is known as linkage disequilibrium (LD). LD is not evenly distributed and appears to be segmented in zones called haplotype blocks. SNPs inside these blocks show strong LD among them but low LD with SNPs in neighboring blocks. Since LD structure is probably a result of the history of recombination events, the limits between blocks are considered to be recombination hotspots.

By selecting a limited number of SNPs each one representative of its correlated genomic environment, it should be possible to capture most of common SNP variation in a population. Those SNPs are called tag SNPs. In European and Asian populations, which have a reduced diversity compared to Africans due to demographic historical reasons, 500,000 tag SNPs seem to capture most of the population diversity (International HapMap Consortium 2005).

1.1.5 Genetic Association

A genetic association analysis doesn't work with families but with non related individuals. Its most basic form would comprise:

- Selecting two groups of unrelated individuals which differ in some characteristic of our interest: people with a disease against healthy controls, acute patients against insidious

⁵ <http://www.hapmap.org>

patients, patients with adverse reactions to medications against patients without, etcetera.

- Genotype a set of SNPs in all individuals
- Check if the genotypes obtained for each SNP are distributed between both groups in a way compatible with chance. We can use simple contingency tables of allelic or genotypic frequencies against individual status (Fig. 2).

If some SNP shows a significant deviation from what we would expect by chance it does not mean necessarily that this very SNP is the causal element of the phenotype that we are studying.

In fact it is much more probable that the causal element is some other polymorphism, not necessarily a SNP, located around the polymorphism that we have tested and in LD with it.

SNP rs14576	Allele A	Allele C	
Disease	1045	1013	2058
Controls	987	1045	2032
	2032	2058	4090

$$\chi^2 = 1,9879$$

$$P \text{ value} = 0,1586$$

Allele distribution compatible with chance

Fig. 2 Allele frequencies for SNP rs14576 in cases and controls are compared using an ordinary 2x2 contingency table. Significance is calculated using a Pearson's chi-square test.

In “Candidate Region Association Studies”, SNPs are selected following a previous hypothesis that points at a certain region. This hypothesis may arise from theoretical reasons or it may be that the

current study aims at replicating some findings in previous studies. When selecting the actual SNPs of the regions to be genotyped, diverse criteria can be used as, for example, maximizing minimum allele frequency (MAF) of the SNPs to increase power, avoid SNPs with incompatibilities with the experimental techniques to be used, select non synonymous SNPs in coding regions or use SNPs that maximize the LD coverage of the region.

In the last decade of the 20th century numerous association studies were performed and around 600 positive associations were reported. However of 166 associations replicated 2 or more times, only 6 presented consistent replications (Hirschhorn et al. 2002).

1.1.6 Genome-Wide Association Studies

A new paradigm that defended a systematic approach of Genome-Wide Association Studies (GWAs) with high number of SNPs began to take form in the late nineties (Lander 1996; Collins et al. 1997). It was the result of several observations and assumptions.

Linkage analysis had failed to map complex diseases. One possible reason was that these diseases respond to polygenic factors that familiar approaches have no power to detect. Association studies, with their potential of recruiting big numbers, could reach enough power to detect those factors.

Candidate Region Association Studies had not yielded very good results. Current biological knowledge to select appropriate candidate genomic locations seemed to be too scarce to work. In fact, the experience of linkage analysis when zooming into a selected region with several genes until detecting the actual causing mutation had shown very often how difficult is to predict in advance which gene will be linked to a phenotype. So a hypothesis free whole genome approach was needed.

Technology was evolving very quickly and dense arrays of SNPs capable of genotyping hundreds of thousands of SNPs at an affordable price would be a reality soon.

The "Common Disease - Common Variant" hypothesis was proposed as a conceptual basis for Genome-Wide Association

Studies. This hypothesis holds that, in contrast to Mendelian diseases, which are caused by rare mutations, complex diseases are probably caused by common variants, that is, present at least in 1% of the population. The theory holds that, after recent human population expansions and its corresponding allele expansions, common diseases, which usually have late onsets and sometimes depend on modern environmental conditions for their development, have partially escaped the action of natural selection. That would explain that complex disease causing alleles would have extraordinary high frequencies compared with Mendelian diseases (Reich et al. 2001).

High SNP frequencies confer high statistical power and SNP allele frequencies higher than 1% are a condition for the success of the GWAs. This created the doubt of whether the "Common Disease-Common Variant" hypothesis was not more an operational approach than a well founded hypothesis. In many cases, doubts were in fact open skepticism. (Weiss et al. 2000).

Whatever the status of the "Common Disease-Common Variant" hypothesis the basic tools required to carry out GWASs at a massive scale were available around 2005: a detailed description of SNP variability and LD structure provided by The HapMap Project and high throughput SNP genotyping techniques with array capabilities in the order of hundreds of thousands. A flood of GWAS ensued.

Since then, a big number of GWASs has been performed on diverse phenotypes, particularly on diseases. Considering the numbers of samples involved in some of the studies and the complexities associated with handling them, the effort and resources of the biomedical research community involved in the process have been enormous.

According to the "Catalog of Published Genome-Wide Association Studies"⁶ (Hindorff et al. 2009), there were 951 published GWAS

⁶ <http://www.genome.gov/gwastudies/>

reports at June 2011⁷, showing an almost exponential increase since 2006 as can be seen in Fig. 3.

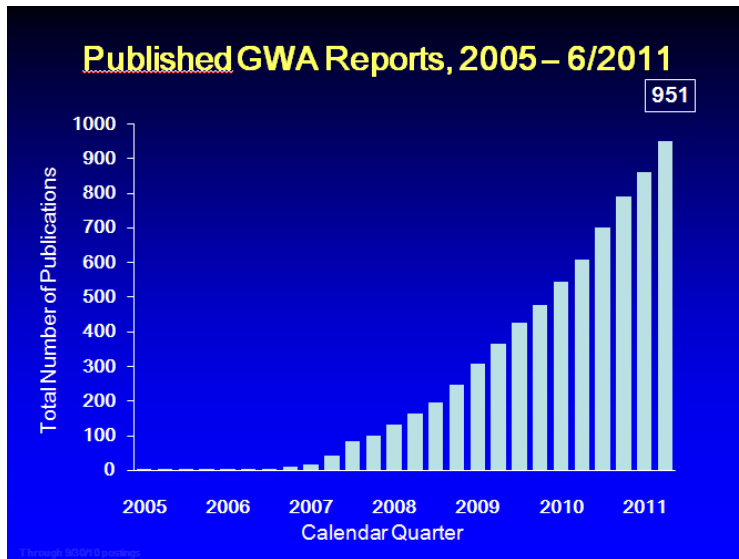


Fig. 3 Evolution of GWASs registered in the "Catalog of Published Genome Wide Associations". Courtesy: National Human Genome Research Institute.

The results of this effort are mixed. On one hand, many significant associations have been found across the genome for multiple phenotypes, some of them with strong p-values and consistently replicated. (See Fig. 4).

The associations found, on the other hand, tend to show small effect sizes, most of them showing an increase in risk between 1.1 to 1.5 fold (Manolio et al. 2009). In fact, the number of samples used in many studies, although in the order of thousands, is not enough to generate the statistical power needed to discover associations with so small effect sizes.

This inability of the associations found with GWASs to predict the risk of presenting a certain phenotype has ignited the so called "Missing Heritability" debate. There are different ideas proposed to

⁷ Only studies attempting to assay at least 100,000 SNPs in their initial stage are recollected in this catalog.

explain the lack of predictability of GWAS results obtained so far. These include additive polygenic effects, which would be individually too low to give significant signals; effects of rare variants, which the most widely used arrays do not allow to test in the context of GWASs; epigenetic effects, etc. A series of recent works, however, showed that common SNPs genotyped in GWAS can explain a large proportion of human heritability in human height (Yang et al. 2010) and in some diseases (Lee et al. 2011). This seems to point in the direction of the additive polygenic hypothesis.

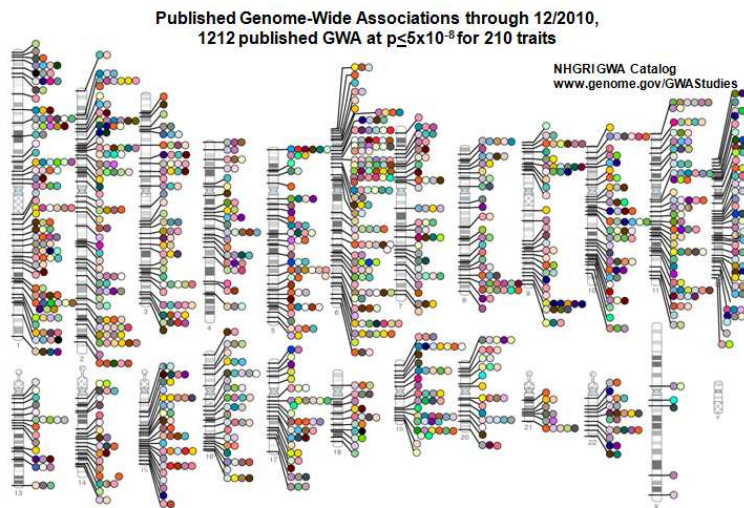


Fig. 4 SNP associations from GWAS mapped to their chromosome locations. Courtesy: National Human Genome Research Institute. (www.genome.gov/gwastudies/)

1.1.7 CNVs

Copy Number Variation (CNV) refers to fragments of the genome that can appear in a different number of copies in different chromosomes (See Fig. 5). By convention, only fragments longer than 1Kb with a similarity higher than 90% are considered CNVs. If an individual lacks any copy of a CNV we call it a deletion.

The influence of CNVs in some disorders has been known for quite some time (Fellermann et al. 2006; Weiss et al. 2008) but only in recent years they have been thoroughly described and cataloged (Kidd et al. 2008). They have turned out to be a more general kind of segregating polymorphism than previously thought. Most of them, however, can be mapped by LD from nearby SNPs (Tuzun et al. 2005; Eichler 2006) so there is debate about what the actual contribution to the risk of most diseases is.

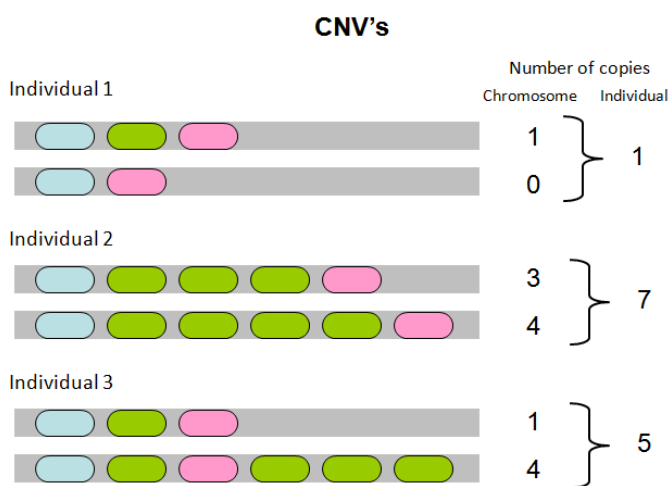


Fig. 5 A CNV region (in green) appears copied different times in different chromosomes and even is completely absent in one of them.

CNVs can contain genes and affect phenotypes by altering its genetic dose. They can also cause pairing problems during meiosis and bring about diverse structural alterations that can be pathological, such as deletions or inversions. CNV implication has been shown for multiple diseases (Tuzun et al. 2005; Eichler 2006).

Genotyping an individual for a certain CNV will mean, not only to establish the number of copies that it harbors, but to determine the individual distribution of each copy. Mere information on number of copies can be sometimes ambiguous because different

combinations of CNVs in each chromosome of a diploid organism will generate the same global figure.

Reference lab methods for discovering and genotyping CNVs like the creation of Fosmid Libraries (Donahue et al. 2007), are too complex for routine epidemiological studies. For such studies, two approaches are generally used.

Array-based Comparative Genomic Hybridization (aCGH) is based in mixing the exact same quantities of fragmented DNA from two individuals after labeling them with two different dyes and then hybridizing the mix against an array of DNA probes. For each probe, the intensity of each dye is measured. Differences in the dye intensity should indicate differences in the amount of DNA of that region present in each individual. This is an indication of copy number differences between them. The relative intensity of the dyes can help us also to quantify more precisely the nature of the copy number variation.

Data coming from aCGH experiments has to be processed to segregate true signals from noise and to call the putative CNV regions. There are multiple algorithms for CNV calling. The sequential nature of the data generated makes it very appropriate for Hidden Markov Models (HMM) based approaches (Day et al. 2007).

CNVs can also be studied using SNP genotyping arrays. The raw intensity values of the 2 alleles of each SNP that are processed in the SNP calling procedure can be processed in different ways in order to call CNVs. There are also several algorithms for this purpose (Cooper et al. 2008; Korn et al. 2008).

1.1.8 Epigenetic Polymorphisms

When cells differentiate in the development of an organism, they acquire characteristics that are passed on to posterior cell divisions without modification of the DNA sequence. Cells "remember" their type cell and this "memory" that can be transferred to descendant cells has to be somehow molecularly codified without changes in the series of nucleotides that constitute the DNA sequence. These kind of inheritable traits are called epigenetic traits.

The most studied molecular mechanism underlying epigenetics is DNA methylation, although there are several others described as for example histone methylation and acetylation. Methylation is involved in the differentiation of tissues and plays a role in the development of some diseases (Jiang et al. 2004).

What has turned epigenetics into a field of growing interest is, first, the discovery that it can be polymorphic and so be responsible for phenotypic differences among individuals, and, second, that epigenetic traits can be inherited from one generation to the next (Rakyan et al. 2006). There are even hints of possible phenotypic variation of individuals dependent of environmental conditions of previous generations and transmitted epigenetically (Pembrey et al. 2006).

As in the case of other polymorphisms, new technological advances are allowing to genotype methylations in a massive, parallel way, with ever higher throughput and diminishing costs. The Human Epigenome Project⁸ is an international consortium in the image of Genome and HapMap projects which

“...aims to identify, catalogue and interpret genome-wide DNA methylation patterns of all human genes in all major tissues. Methylation is the only flexible genomic parameter that can change genome function under exogenous influence. Hence it constitutes the main and so far missing link between genetics, disease and the environment that is widely thought to play a decisive role in the aetiology of virtually all human pathologies.”⁹

1.1.9 Ultrasequencing

The whole intellectual building of genetic mapping, as said before, was founded on the fact that it is not possible to genotype the whole set of variation that a set of individuals may carry and therefore methods have to be found to explore it from a subset of polymorphisms, the “genetic markers”. If cheap whole genome sequencing is available, then the former restriction does not hold

⁸ <http://www.epigenome.org>

⁹ <http://www.epigenome.org/index.php?page=project>

any more because the whole of human genetic variation (at least the variation of the genome sequence) can be accessed and statistically tested against phenotypic variance.

We are not yet there. Sequencing costs have fallen several orders of magnitude but have not yet reached the price level of a SNP array. Sequencing with current technologies generates short reads (a few tens or hundreds of base pairs) without distinguishing chromosome phase. These reads have to be compared against public known sequences to be assembled. This creates a bias about what kind of new mutations can be detected (Bonetta 2010). The bioinformatics resources that have to be committed to the processing and interpretation of the huge amounts of data generated by second-generation sequencing are enormous. New developments in algorithms and procedures are needed.

There are, however, several approaches to this technique that are already yielding results. The sequence of members of a family carrying a Mendelian disease can be compared to the Human Reference Sequence to try to detect putative disease causing mutations (Roach et al. 2010). Candidate genes from a diseased person whole genome sequence can be scanned searching for rare mutations (Lupski et al. 2010).

Whole genome exon sequencing is becoming a cost-effective alternative for studying familiar diseases and there are already some commercial specialized products available. The whole exon sequence is compared against public sequences looking for deleterious mutations (Ng et al. 2010).

Taking advantage of the new sequencing techniques, the 1000 Genomes Project¹⁰ is trying to sequence a large number of people, beyond the 1000 proposed initially, to create a comprehensive catalog on human variation. The project has already released to the public the full genomes of almost 1000 individuals, albeit at a relatively low coverage. It is expected that the project will be almost finished over the near months.

¹⁰ <http://www.1000genomes.org>

1.2 The Technological Revolution

1.2.1 Moore's Law

In 1965, Intel co-founder Gordon E. Moore predicted, in a paper that would become extremely famous, that the computing power of an integrated circuit would double approximately every two years (Moore 1965). This is an astonishing rate of growth that promised to change our world:

Integrated electronics will make electronic techniques more generally available throughout all of society, performing many functions that presently are done inadequately by other techniques or not done at all. (Moore 1965)

Integrated circuits will lead to such wonders as home computers -or at least terminals connected to a central computer—automatic controls for automobiles, and personal portable communications equipment. The electronic wristwatch needs only a display to be feasible today. (Moore 1965)

Moore's law has hold. The promise was delivered and computer technology has doubled capacities every two years. As a consequence, our world is not the same in many respects that it was in 1965.

But not only computers improved in capacity at exponential rate. Lots of other technologies have grown at a similar pace in the last decades. Biology and biomedical technologies are among them.

As far as genetics is concerned, sequencing and genotyping techniques have improved their throughput rates and decreased their unit prices by several orders of magnitude. It took around 3,000 million dollars and almost a decade to The Human Genome Project to sequence the first human genome. A decade later, new generation sequencing technologies are approaching the price of sequencing an individual to the barrier of 10,000\$. The National Human Genome Research Institute publishes detailed data about

the evolution of sequencing costs (Wetterstrand). Fig. 6, obtained from the Institute's web page, shows a reduction in the cost of genome sequencing in a 10 year span of four orders of magnitude. Notice also that, from 2007 on, the reduction progress has even outstripped Moore's Law prediction equivalent.

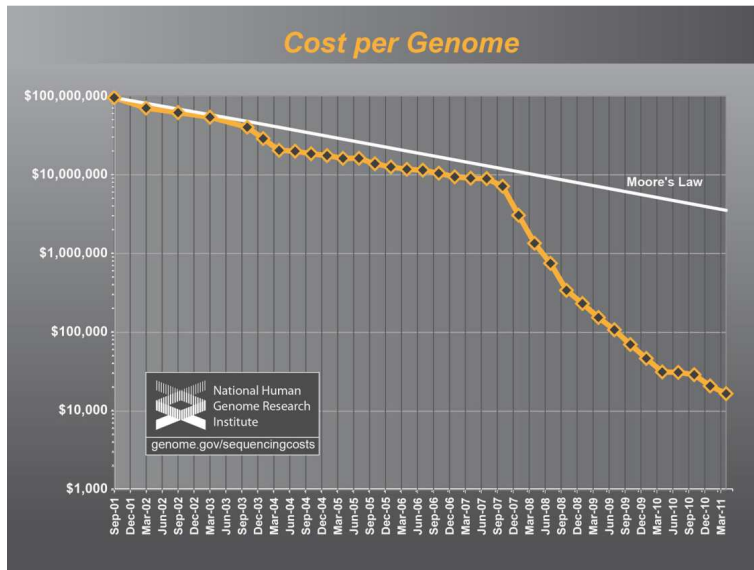


Fig. 6 Cost per Genome evolution since 2001. Wetterstrand KA. DNA Sequencing Costs: Data from the NHGRI Large-Scale Genome Sequencing Program Available at: www.genome.gov/sequencingcosts. Accessed [2011/08/03].

SNP genotyping presents a similar process. Beginning as a manual and expensive procedure, current array-based parallel genotyping platforms allow to genotype around one million SNPs of an individual with high accuracy for a few hundred dollars.

Everywhere we see a similar pattern. Technologies allow for massive and cheaper production of already existing data and bring to reality new methods to obtain new types of data. We have entered an era of massive data production.

Of course this is very good news for genetics in general and for genetic epidemiology in particular, but it is changing important

aspects of the way science needs to be done. Data generation has stopped being the bottleneck of the research activity. Very often now, data management and analysis, formerly relatively straightforward activities, stand as the real bottleneck. Science stops being the kind of craftwork activity that once was and is becoming an organized endeavor with a division of highly specialized labor, non unlike many modern industries.

In this changing environment there are important methodological, organizational and technical changes that have to be implemented in order to adapt to the new situation.

1.2.2 Data Handling

The most immediate challenge has to do with the practicalities of handling enormously large amounts of data. One complete human genome sequence at 30-fold coverage generated with a new ultrasequencing device takes ~90Gb of disk memory (Bonetta 2010). Not all technologies are as resource hungry as sequencing but, whatever the kind of data used, the need soon appears of some kind of informatics expertise and dedicated infrastructure. Raw data is processed generating at each step new sets of files. Original, working and final data has to be stored in an easily recoverable way. A backup system has to be installed to guarantee survival of data in case of a major incidence, etc.

The simple handling of such big amounts of data is troublesome. Sometimes it is faster to send a hard drive by ordinary mail than sending its information through an Internet connection. Big files are difficult to edit or manipulate. Trivial operations with smaller files become a nightmare that needs special software or techniques.

The parallelization and automation of lab work implies that specialized software has to be used to manage it. Lots of commercially available products have been designed for this purpose under the generic name of Laboratory Information Management System (LIMS).

The global effect of this situation has been to improve the level of Information Technologies (IT) expertise in biological research

groups, either by adding IT professionals to the staff or by learning new skills. Often both.

1.2.3 Processing Data

When working manually, there are a lot of steps that are done naturally using our human abilities without considering the complexity they require. For instance, a simple look at a gel electrophoresis allows knowing, by the position of the bands, which microsatellite alleles are present in the sample. However, when working with a genotyping array of, say, 500K SNPs, intensity values for each SNP are produced. The calling of the alleles has to be somehow automated. The algorithm that such process requires is not necessarily simple.

The Wellcome Trust Case Control Consortium (WTCCC) faced such a challenge when developing its pioneering study on 7 seven complex diseases (WTCCC 2007). The array technologies, the samples and the management resources were all in place. They had, however, to devote important resources to the development of a calling algorithm called CHIAMO to interpret the intensity values given by the arrays and decide which alleles presented each genotype.

As another example, when SNP genotyping companies began to include SNPs in putative CNV regions in their arrays, research groups had to develop computer algorithms to call the presence of actual CNVs from the SNP intensity data obtained from the arrays e. g. (Cooper et al. 2008).

Although at first sight it may look strange that companies release research products without the appropriate algorithms already developed to interpret the results, this happens because the research labs want to access the technologies as soon as possible. Furthermore, this is a two-way cooperation since the research groups will benefit from early product release but can help with their expertise in the development of good processing tools.

Also, basic quality control methods checking that the genotyping process is running correctly have to be automated. Replicated tests, reference samples, Hardy-Weinberg equilibrium tests, family

inheritance consistency, etc have to be controlled in a novel way in a massive production environment and, thus, require specialized software and procedures.

1.2.4 Interpreting Data

Analyzing results for a case/control association study of a million SNPs, to use an example, involves a set of challenges not present in a study of a handful of SNPs.

1.2.4.1 Multiple Testing

There is a massive multiple testing problem. In a million tests, under the null hypothesis, there will be approximately 10,000 p-values smaller than 0.01 (i.e. the 1% of the tests).

Standard methods for multiple test correction, such as Bonferroni or False Discovery Rate (FDR) (Benjamini et al. 1995) can be used. However, after Bonferroni correction for one million tests, the standard 0.05 significance threshold becomes a nominal p-value of 5×10^{-8} . This is a very stringent value that will diminish the power of the study to detect weak associations.

A further point that makes standard methods inadequate in the context of GWAS, is that most of these corrections assume independence of all the test performed in a GWAS. But SNPs are not independent because, among other reasons, the variants they present are correlated due to linkage disequilibrium. More realistic p-values can be calculated using permutation and resampling methods but with the disadvantage that these are computationally costly and have to be repeated for every set of data.

In another approach, since the human genome LD structure is known after the HapMap Project (International HapMap Consortium 2005), effective numbers of tests can be calculated for commercially available arrays (Duggal et al. 2008). A global effective test number can also be calculated from all available HapMap SNP information of the genome and results fall in the range of 10^{-7} to 10^{-8} . However, the standard practice, in order to simplify, is to use the 5×10^{-7} significance threshold value proposed by WTCCC (WTCCC 2007).

Anyway, significance thresholds have a limited importance since there are a lot of possible methodological artifacts that can generate false positives. Replication of results with independent samples is the key criterion for the community to assign credibility to an association (Chanock et al. 2007).

1.2.4.2 Looking at the Data

Human brains are brilliant machines capable of amazing computational feats, but only if they are fed with data in the proper formats and structures. A long melody can be effortlessly remembered but if notes are codified as figures almost nobody will be able to remember the resulting list of numbers.

A researcher presented with a list of a million association test results, each with genomic position information, p-value, odds ratio, etc, will not be able to make any sense of it. That is why descriptive statistics were invented and data are ordered, summarized and means and variances are calculated. Also, a very powerful approach consists in structuring data in some adequate graphical layout and taking advantage of human's innate visual computational capabilities to interpret them. This approach has been intuitively known for centuries and is behind the idea of the graphical representation of mathematical functions. The question

$$\text{is } f(4) > f(3)? \text{ for } f(x) = x^3 - 9x$$

becomes trivial if, instead of the numerical representation of the function, the equivalent graphical representation of it is provided as can be seen in Fig. 7.

In the case of GWAS, two graphical techniques have become fundamental tools of the field: Q-Q plot and Manhattan plot.

A Q-Q plot is a general graphical method for comparing two sets of data and assessing if they belong to the same distribution. In GWAs it has taken a particular form. Under the null hypothesis, p-values obtained in a genomic association test should present a uniform distribution between 0 and 1. An ordered list of empirical p-values is plotted against an ordered list of expected p-values under uniform distribution. All values are previously converted to -

log10 before plotting to improve the resolution on the very small values. If the null hypothesis holds true at the whole-genome level, the points will fall on the identity line as shown in Fig. 8.

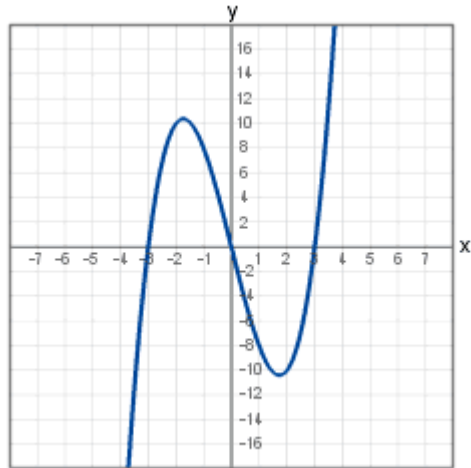


Fig. 7 Partial graphical representation of the function $f(x) = x^3 - 9x$.

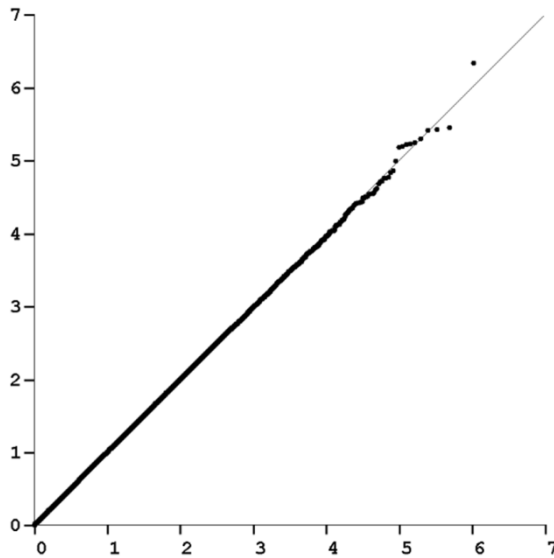


Fig. 8 Q-Q Plot fitting to expected values under null hypothesis

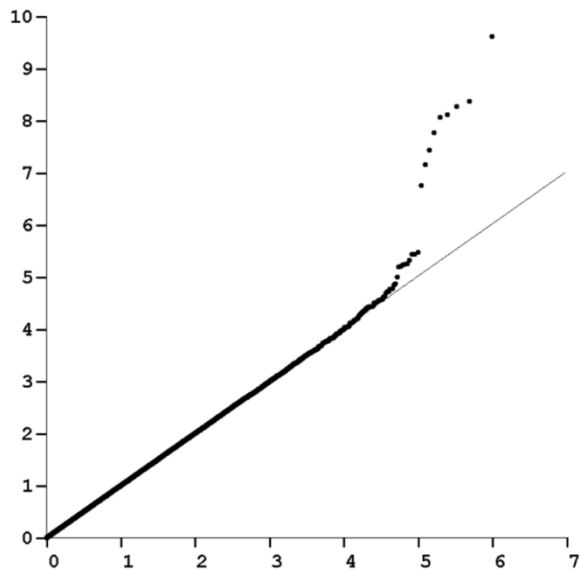


Fig. 9 Q-Q plot showing some markers associated with the phenotype.

In contrast, when Q-Q plot produces an image deviating from the identity line, the characteristics of this deviation convey information about the whole experiment. If some markers are associated with the studied phenotype, a deviation in the upper part of the graph will appear as shown in Fig. 9, indicating that several small p-values have been detected that go beyond the expectation of random p-values under a true null-hypothesis.

Sometimes, however, as shown in Fig. 10, a general deviation appears in an important part of the span of the graph. When this happens, it raises important suspicions that some kind of artifact has generated false positives. Population structure between cases and controls is the first one that comes to mind.

Q-Q plots allow us to make an instant visual assessment of the global behavior of a multiplicity of tests in a similar but more detailed way of what a FDR correction achieves analytically.

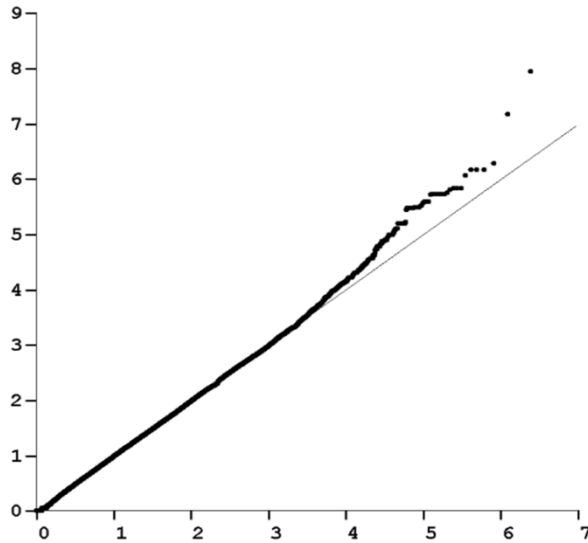


Fig. 10 Suspicion of population structure after the long deviation of the Q-Q plot from the identity line.

The second form of visual device typical of GWAS, Manhattan plots, help us to distinguish true positives from experimental or data artifacts. It is a plot of all $-\log_{10}(\text{p-values})$ of SNP tests against its genomic position. The rationale of this plot is that true positives, since they are in LD with nearby markers, will cluster in certain regions, while highly significant values produced by artifacts will be isolated. The clustering of associated markers will create a typical tower-like structure from where the Manhattan plot takes its name. See Fig. 11.

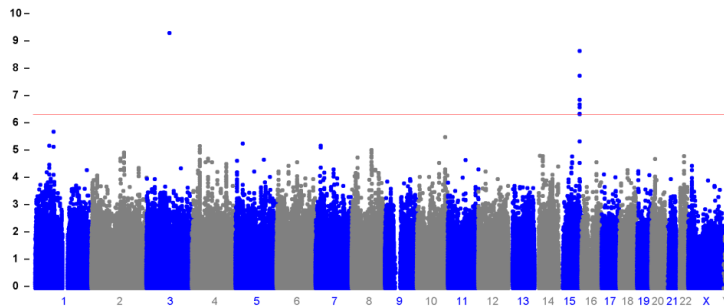


Fig. 11 Example of Manhattan plot. A tower in chromosome 15 seems to hint at a true association while the isolated highly significant point in chromosome 3 is probably an artifact.

LD plots are another example of successful visual representation of complex data (Fig. 12). All the SNP to SNP LD values are plotted into a matrix showing, for example, D' values and different colors depending on the level of significance.

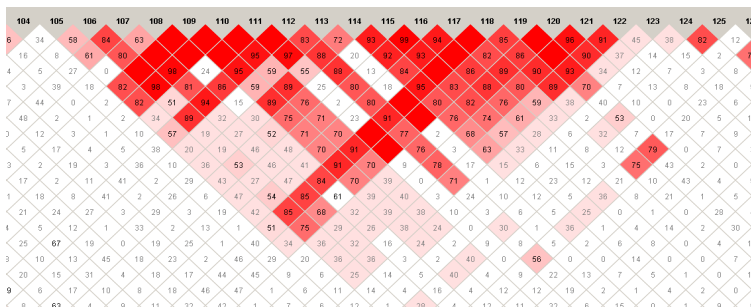


Fig. 12 A haplotype block emerges as an inverted pyramid in a LD plot. Surrounding the block appear two putative recombination hotspots. Image generated with the program Haploview (Barrett et al. 2005).

Hundreds or thousands of individual statistics values are handled in a parallel way in the image. The emergent colored inverted pyramids allow an immediate visual identification of the haplotype block and recombination hotspots structure.

1.2.4.3 New Approaches

Old methods and procedures, as seen before, can be adapted to the abundance and types of data currently generated. However, these new data make possible also the development of new statistical methods that can extract richer information about the relationships between genomes and phenotypes.

Dense genotype data together with extensive SNP population information allow for whole genome individual haplotypes to be estimated. Those estimated haplotypes can be subsequently screened for associations with a given phenotype. Haplotype estimation can be performed with different algorithms. The most commonly used are Expectation-Maximization (Excoffier et al. 1995) and PHASE (Stephens et al. 2003). The last one has shown

in simulations greater accuracy in imputing individual haplotypes but is much slower. In theory, under certain conditions, haplotypes should be able to map diseases with more power than individual SNPs.

Detailed population diversity descriptions such as the ones coming from HapMap or the 1000 Genomes Project make it possible to estimate, for any given genotyped individual coming from a population for which ample genotype information is available, polymorphisms that have not actually been genotyped. This is called “Imputation” and is based on the fact that, within a certain region, unknown polymorphisms are in LD with known ones (Marchini et al. 2007; Browning et al. 2009; Li et al. 2010). Since imputation allows obtaining data from polymorphisms that have not actually been genotyped, it does reduce costs if it turns out to be reliable. Simulations and parallel genotyping of imputed data show that imputation predicts acceptably well, although some studies suggest that it could increase the number of false positives (Almeida et al. 2011).

There is a lot of functional knowledge stored in public databases helped by the development of a conceptual systematization as in The Gene Ontology¹¹ (Ashburner et al. 2000). Several GWAS results analysis strategies have been developed that look for enrichment in association in regions related to some biological function or biochemical path (Elbers et al. 2009; Eleftherohorinou et al. 2009; Peng et al. 2010). So, in studies where no individual marker has shown association as a result of the stringent significant thresholds imposed by multiple test corrections, functional associations can be found that deviate from what could be expected by chance.

Another approach made possible for the availability of data is to search for genome wide epistatic interactions. Markers who independently do not present association may present joint combinations of alleles that present increased risk for the phenotype of interest. Among many examples, Multifactor Dimension Reduction (MDR) is a widely used strategy designed

¹¹ <http://www.geneontology.org>

with the goal of detection of epistasis (Moore et al. 2006). The problem with the study of epistasis is the explosion of the combinatorial space when interactions of more than two markers are considered. An exact computation of all probabilities quickly becomes impossible, which makes it difficult to evaluate the significance of any finding. This is, as seen before in the case of complex pedigrees, an appropriate field to try alternative heuristic approaches that offer approximate solutions.

1.3 Bioinformatics

Biomedical sciences have reacted to the new highly technological environment and the abundance of data with an increased use of computer resources and skills. The high degree of specialization has even suggested the possibility of defining an independent scientific area: bioinformatics.

The solutions to cope with the challenges posed for the technological advances described in previous sections took generally the form of isolated software developments, each one designed to solve a certain need. There has not been, in principle, a joint effort to coordinate a consistent body of solutions. Instead, every research group or institution has developed its own approach to handle a particular issue. The result is an enormous set of software tools (see for example "The Rockefeller List of Genetic Analysis Software"¹²) with complementary but also overlapping functions, different input/output data formats, different operative system and hardware requirements, different theoretical approaches, etc.

Data processing work usually takes the form of a pipeline of connected steps where output from one step becomes the input for the next ones. Researchers can find themselves in the predicament that, although it may exist software developed to perform each of the different steps they need, to coordinate all these discordant modules requires knowledge in Information technologies beyond their possibilities. Even in cases of groups with strong computer skills, sometimes it is more cost effective to reprogram the whole

¹² <http://linkage.rockefeller.edu/soft>

process than to reuse partial pieces of software developed by third parties.

The bioinformatics community, aware of the problem, has fostered a series of initiatives in order to try to create a common working frame for the reutilization and sharing of software developments. Bioconductor¹³ is a platform for the creation of tools for the analysis of high-throughput genomic data under the R statistical programming language¹⁴. BioPerl¹⁵ defines itself as a "*community effort to produce Perl code which is useful in biology*". These projects, achieve a certain degree of standardization by creating a set of rules of programming and documenting functions and data structures that makes easier their interaction. However, different projects cannot interoperate fluently between them and there is still an important human intervention in all the assembling process.

Web services represent a further level of integration. Functions and services, independently of the programming environment where they are developed are published in the Web in such a way that any program following a HTTP/SOAP protocol can connect to them and ask information from them. Web services define in an unambiguous way their functions and data structures using a protocol called WSDL (Web Services Language Description). Using this information, applications can be developed that use the functions that web services provide and, thus, users do not have to program them again.

Taverna¹⁶ provides a software application that allows users to create working pipelines calling successive services. These can be either web services or services created by the own user. Data flow through the pipeline and are transformed in successive steps until the final result is obtained. Taverna also includes visual tools to help organize and run the pipelines.

¹³ <http://www.bioconductor.org>

¹⁴ <http://www.r-project.org>

¹⁵ <http://www.bioperl.org>

¹⁶ <http://www.taverna.org.uk>

Another workflow management system is Galaxy¹⁷ (Goecks et al. 2010). It offers a web based system of creating, using and sharing genomic pipelines which is gaining growing acceptance among the bioinformatics profession.

Taverna and Galaxy are two relevant examples of the current interest in workflow and data structure standardization. There are, however, dozens of similar applications.

All these efforts, although they provide comprehensive solutions to the problem of sharing and reusing biological tools, have the problem that they require a certain level of specialization. Sometimes bioinformaticians are producing tools that can only be used by other bioinformaticians. This is important and necessary but there are people in the biomedical community actively working in such areas as genetic epidemiology or gene mapping, that lack extensive IT knowledge and that are sometimes left aside by the above approaches. A simple, friendly user application that integrates the usual steps in genetic mapping and genetic epidemiology analysis should be of big help to this group of researchers.

1.4 Objectives

The general goal of this thesis is to develop new algorithms, methods and tools to cope with the challenges posed by the ever growing amounts of data generated by modern high throughput technologies. Methods should be useful in themselves, but they should also be general proofs of principle about how information technologies can help the omics sciences. Additionally they should be made available to the research community. The usefulness of our developments will be shown by their application in particular projects. Particular interest is devoted to the development of methods that create visual representations of data that leverage on the natural visual computation powers of humans to spot patterns and draw conclusions.

¹⁷ <http://main.g2.bx.psu.edu/>

This general goal materializes into tree specific objectives.

1- Creation of an integrated suite for epidemiological genetic analysis that is easy to use and hides all informatics and formal complexities from the user under a common and simple interface. The particular objective here is to create a tool that enables biological researchers lacking extensive training in Information Technologies (IT) to perform usual procedures followed in genetic epidemiology. The application, in addition, should be of utility for IT literate researchers, since the automation of processes should enhance significantly their productivity.

2- Development of an interactive visual application to help in the process of calling copy number variations (CNV) using Hidden Markov Models (HMM). Visual presentation of data should help users to assess the quality of the calls obtained and to choose optimal HMM parameters particularly in situations where the calling process can be quite complicated by the amount of noise of the experiments.

3- Development of an interactive visual application to show the results of exhaustive haplotype association tests in a way that can help the user to locate visual patterns that can complement the information obtained from purely analytical methods. Visual aids provided by the application should allow the recognition of patterns associated to previously hidden true associations and help distinguishing real positives from methodological artifacts.

In parallel to these developments, collaborations in particular genetic analysis studies should be conducted in order to test in real situations the applications developed and to get insights of the actual needs of the research community. This information should constitute a valuable feedback to improve and add new functionalities to the tools being developed

2. SNPator

This chapter is not intended to be a manual or a tutorial of the application and, consequently, it will not present all analysis options of the program nor instructions how to use it. It has the purpose of explaining the objectives of the project, the challenges that arose during its execution and the solutions that were found to overcome them.

2.1 Conception and Objectives

SNPator (SNP Analysis To Results) was conceived in the context of the Centro Nacional de Genotipado (CeGen)¹⁸, Spanish National Genotyping Center, with the double objective of satisfying the data management needs of the institution and, at the same time, creating a user friendly environment for the quick and easy analysis of genomic data.

This origin, as we shall see, determines some of the technical and structural decisions in the development of the tool and helps understanding its differences with other applications that, having similar objectives but different motivations and goals, have appeared in recent times.

CeGen was created at the end of year 2003 by Fundación Genoma España¹⁹ with the objective of offering to the Spanish scientific community an easy access to the SNP genotyping technologies that had appeared in recent times. CeGen was organized into three different Genotyping Nodes placed in different Spanish cities (Barcelona, Madrid and Santiago) and a central Coordination Node in Barcelona.

The workings of the institution are simple. A user/customer, which has a set of DNA samples relevant for the study of some biological hypothesis, approaches either the CeGen's Coordination or one of the Genotyping Nodes. The user, with the help of the institution's staff and by means of a set of bioinformatics tools put at their disposal, designs the exact experiment to be carried out. This task involves basically selecting which SNPs to genotype in which samples and with which technology.

¹⁸ <http://www.cegen.org>

¹⁹ <http://www.gen-es.org/>

This process is called the "Pre-genotyping phase". Some public applications are available to make it easier. Some of these tools were developed by the bioinformatics team of CeGen, working together with the Instituto Nacional de Bioinformática (INB)²⁰. Pupasuite²¹ and SySNPs²² are two examples of tools designed to help users in the selection of SNPs.

In a second phase, when the DNA samples of a certain project reach the genotyping platforms, they are processed and genotyped ("Genotyping phase") according to the design created in the Pre-genotyping phase. A crucial point is that not necessary all the samples of a project are going to be processed in the same platform.

Once genotyping is finished, the Post-genotyping phase starts. This phase generates the bioinformatics needs that SNPator tries to solve:

- All data of all projects have to be stored in a centralized way and have to be quickly accessible from all nodes. Nevertheless, data uploaded from a certain node has to be protected from modification from other nodes.
- Genotyped data has to pass a quality control process to ensure the reliability of the results offered to the client and minimize the incidence of genotyping errors. Since different datasets from one project can be genotyped by different nodes and the quality control process has to take into account the totality of data, this process has to be performed in a coordinated way.
- Final genotypes have to be handed to the user in an easy way that enable him to download them in different formats suitable for genomic or statistical packages used frequently in scientific studies.
- Given the importance of confidentiality issues in relation with sample management, a strict system has to be implemented to ensure that no set of data, which are the

²⁰ <http://www.inab.org/>

²¹ <http://pupasuite.bioinfo.cipf.es/>

²² <http://www.sysnps.org/>

property of each user, can be accessed by anyone else without the corresponding permits.

These are the basic needs of the CeGen institution that SNPator intended to solve. On top of all that, we tried to add, in a fully integrated way, a suite of SNP data analysis tools which allows performing typical genomic studies in a quick and easy way without having to bother about computer issues. In this sense, the goal of the application is to liberate users from hardware and software dependencies, and from complex management and IT knowledge tasks, allowing biologists in particular and biomedical researchers in general to develop their projects in genetic epidemiology without extensive informatics expertise.

It is from the integration of all these goals in a single tool that the architecture and technical design of SNPator have emerged. As an added requirement, when the program is used by external users with no relation with CeGen, no interference in its efficiency and easiness of use should come from the fact that it can perform also, when required, the data management of the Institution.

2.2 Web Application. Easy Access Everywhere.

The Institutional requirements of SNPator, with a centralized data storage system and users of very diverse profiles accessing from different locations, make it a clear candidate to be developed as a web application. That involves that the program runs in a central sever and users can connect to and run any processes with the help of a standard web navigator. Using this approximation, the final user is free of any software installation requirements and operative system or hardware dependence. Connectivity issues are also simplified, since web applications use the standard Internet infrastructure access.

The basic architecture of our system follows the standard architecture of web applications in the industry environment²³ (see Fig. 13)

²³ http://www.ibm.com/developerworks/ibm/library/it-booch_web/

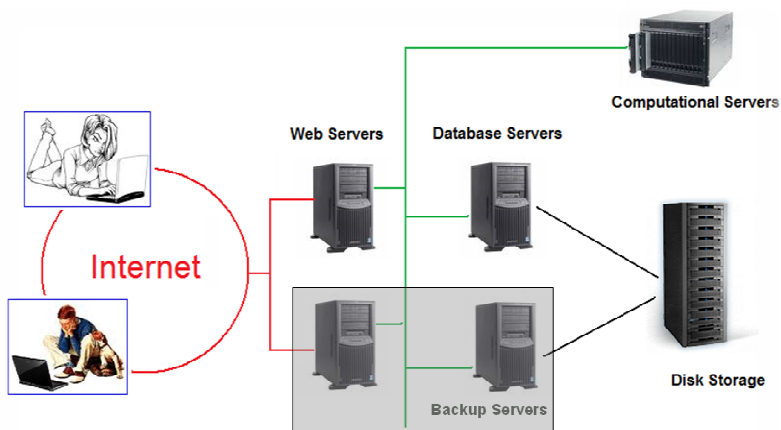


Fig. 13 The basic structure elements of a typical web application

Anyone connected to Internet using a web browser can connect to SNPator web servers using the URL <http://www.snpator.com> or <http://www.snpator.org>. Web servers are the only machines accessible from outside the system and their function is to provide a first layer of processing of the user requests.

Besides of the Linux operative system (SuSe Linux Enterprise Server) as in all our servers, the software installed here is basically an Apache http server²⁴ and all functionalities are programmed using PHP. No confidential data are stored in these machines and no complex calculations are performed here due to considerations both of security and performance. The Web Servers are responsible essentially of the user/password authentication of connections and of keeping the communication with the SNPator user. They delegate all tasks to be performed to the database and calculations servers.

One of the two web server machines is running providing service and the other is idle with functions of backup. It is due to take the functions of the primary in case of any incidence of service. SNPator URLs are linked to a virtual IP independent of the physical

²⁴ <http://httpd.apache.org/>

IPs of both web servers. This virtual IP, configured in the running server, gets all the web requests from users. In case of incidence, a script is executed that turns up all services in the backup computer, transfers to it the virtual IP and turns down the main machine.

The rest of elements of the architecture are installed in safer parts of the network structure of the institution. An internal isolated network ("Demilitarized Zone" or DMZ in the IT jargon), painted in green in Fig. 13, with no access from Internet has been established.

The Database Servers are connected to this DMZ. They are running MySQL database management software (community edition)²⁵ and are responsible for providing to the Web Servers all the data which those may require. InnoDB²⁶ MySQL tables have been used in the design of databases to allow for transactional queries. Transactional modification of databases allows for a consistent reconstruction of the previous data state in case that some process was interrupted by some external incident. They are routinely used in data critical environments for this trait although their performance is slower than other table technologies.

Given the huge amounts of data managed by the system and its high rate of increase, a Storage Area Network²⁷ (SAN) IBM DS4200 with Fiber connection has been installed. Disks are placed in external cabins with redundancy systems that allow to add new disks or to substitute damaged ones without interrupting the service.

As in the case of the Web Servers, a backup database server is ready to take the functions of the main server in case of a malfunction of it. Fiber connection to the disk storage is switched to the backup Database Server and the Web Servers are reconfigured to change their data requests to this machine too.

²⁵ <http://www.mysql.com/>

²⁶ <http://www.innodb.com/>

²⁷ <http://www.redbooks.ibm.com/abstracts/sg245470.html?Open>

Finally, most heavy computations are redirected to the Computational Servers to avoid overcharge of the web server or database machines and so avoid that the rest of users connected to the system may be affected by the overload caused by a certain user. It will be later shown how this delegation of work is actually implemented in the logic of SNPator.

There are some consequences of the initial design decision that have determined in an important way some characteristics of the application. First of all, all data is centralized. All genotype, SNP and phenotype information is stored in the Database Servers. When users want to use SNPator analysis capabilities, they have to upload data and only afterwards begin to work with it. In principle, this should not be cause of concern because of the strict confidentiality policies implemented in the application and because data can always be uploaded in a coded form. Nevertheless, given the crucial importance that data ownership has in some fields, this could be an issue in some circumstances.

The web application basis of SNPator influences also heavily the general aspect of the application and limits, unfortunately, the kind of visual approaches that could be possible in a local application programmed, let's say, in C++ or Java. Of course, everything is possible also in a web application frame, but the complexity of programming increases in some areas to prohibitive levels.

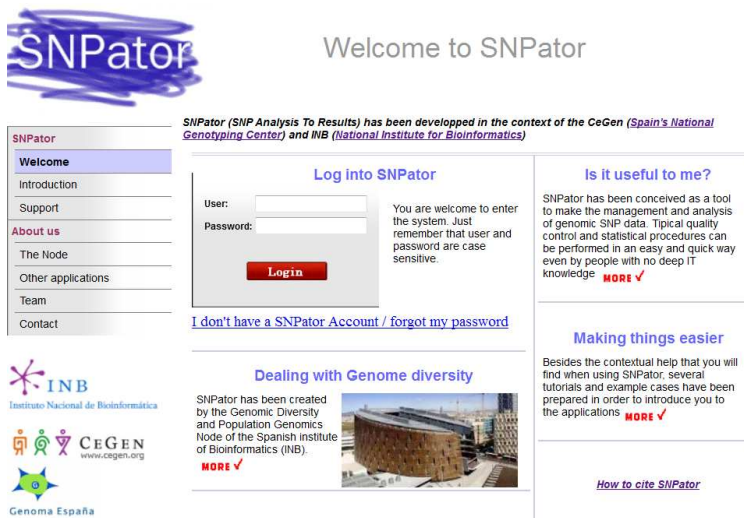
Also, as a consequence of placing an application on the web for users to connect remotely, a lot of traffic data will be generated flowing through the net. This has been a hot issue, particularly as the size of genomic studies has grown bigger and bigger, and a non trivial amount of the time that the creation of SNPator has required, has been devoted to cope with these questions.

The complexity of SNPator overall structure, finally, makes it quite difficult to clone it in other centers. To install "private SNPators" in other institutions for internal use, an important level of IT knowledge and resources has to be available and a strong commitment with the project is needed. This is a question that has become a priority in the next planned developments for the application.

2.3 Privacy Issues. Who can Access the Data.

Although they may not seem important from a strict research or technical point of view, privacy issues have to be seriously taken into account if SNPator has to be accepted by the scientific community as a tool for genetic analysis. Users must have the certainty that no one but people authorized by them will be able to access their data while, at the same time, the system must provide a flexible way to tackle the collaboration and delegation issues that appear in the research process.

All SNPator users are provided with a username and password which identifies them and that have to be introduced when connecting to the system. (See Fig. 14)



The image shows the SNPator web page. At the top left is the SNPator logo. To its right is the text "Welcome to SNPator". Below the logo is a navigation menu with items: Welcome, Introduction, Support, About us, The Node, Other applications, Team, and Contact. The "About us" section is expanded, showing logos for INB (Instituto Nacional de Bioinformática), CEGEN (Genomic Diversity and Population Genomics Node of the Spanish Institute of Bioinformatics), and Genoma España. The main content area is divided into three columns. The left column is titled "Log into SNPator" and contains a login form with fields for "User:" and "Password:", a "Login" button, and a link: "I don't have a SNPator Account / forgot my password". The middle column is titled "Is it useful to me?" and contains text about the tool's purpose and a "MORE ✓" link. The right column is titled "Making things easier" and contains text about tutorials and a "MORE ✓" link. At the bottom of the main content area is a link: "How to cite SNPator".

Fig. 14 SNPator web page with general information about the application and users' main entry point.

Two layers of control in the way users can access data have been implemented. On one hand, there is a system of user groups that determine which *studies* (data sets) can be accessed. This organization into groups allows the easy creation of communities of

people working with the same data. On the other hand, there is a hierarchy of privileges that determine the kind of access that any given user may have. Thus, although a full group of people can access a *study*, it is usually the case that some among them can add, edit and delete data, while others are only allowed to work with existing data, and yet others will only be able to look up what is being done, without being able to introduce any changes.

The management of user and group privileges is performed by the System administrators and is treated as a very sensitive issue, not only as far as CeGen users is concerned, but in relation with all users.

CeGen users get automatically a SNPator username and password when they have their genotyped data at their disposal but anyone can have a username to take advantage of SNPator analysis capabilities using their own data. Access can be requested from the SNPator web page.

2.4 Data Structure. Transparency.

SNPator has been created with a rather complex software and database design in order to ensure the proper execution of all its functions (Fig. 15). However for SNPator to achieve its goal of being an intuitive and easy to use tool, all its technical complexities have to be hidden from the user and a virtual simple interface and data structure has to be offered to him.

We define a *study* as a set of SNPs and samples and the genotypes resulting from genotyping those SNPs on those samples. A *study* can be understood as a project or a space work inside the program which includes all the relevant data, together with all work and analysis results that users produce working within it.

A *study* is the basic unit of work. It is at the level of the *study* that all access privileges are granted. No information can be shared between *studies*. There is no process in SNPator that can work with data belonging to different *studies* at the same time. This trait,

The *SNPs table* contains the list of SNPs of the *study* with fields that can store information about these SNPs including, among others, *Chromosome*, *Position*, *dbSNP*, etc (See Fig. 16). Different analysis options within SNPator will need different kinds of information so, depending on the kind of analysis that the user wants to perform, the corresponding fields will have to be filled.

Since SNP chromosome and position information is necessary for a lot of procedures and are quite cumbersome to manage, an automatic system to retrieve these data from the dbSNP public database²⁸ has been implemented. If the user requires it, the *Chromosome* and *Position fields* in the user's *study* are filled up following the SNP codes previously introduced.

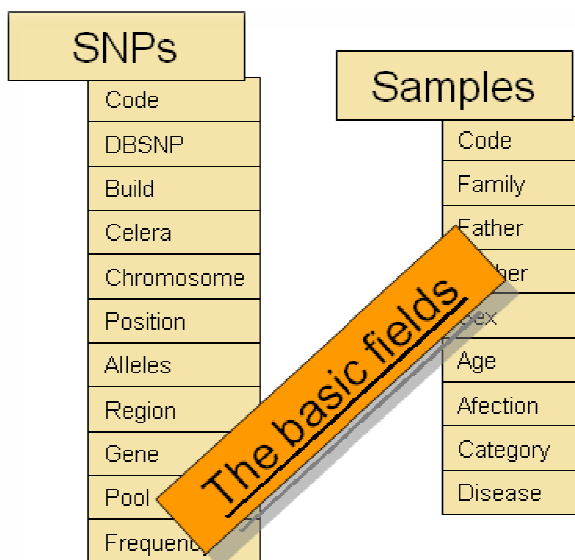


Fig. 16 All SNPator studies will contain these fields in the SNPs and Samples tables. New customized fields can be added by the user to any particular study.

Similarly to the *SNPs table*, the *Samples table* contains the list of samples of the *study* with fields that contain information describing those samples (Fig. 16). *Sex*, *Age* and *affection* are the most

²⁸ <http://www.ncbi.nlm.nih.gov/projects/SNP/>

frequently used fields in association studies, while pedigree information fields are used in family-based analysis.

Code	Family	Father	Mother	Sex	Age	Afection	Category	Disease
SPL6985	1			M	46	N	CFT_1	Lupus
SPL6991	1			W	44	N	CFT_1	Lupus
SPL6993	1	SPL6985	SPL6991	M	12	Y	CFT_1	Lupus
SPL6994	1	SPL6985	SPL6991	W	8	N	CFT_1	Psoriasis
SPL7000	1	SPL6985	SPL6991	W	2	N	CFT_1	Lupus
SPL7019	2			W	11	Y	CFT_2	Lupus
SPL7022	2			W	46	Y	CFT_2	Psoriasis
SPL7029	3			M	13	N	CFT_1	Lupus
SPL7034	4			M	78	N	CFT_2	Psoriasis
SPL7048	4			W	81	N	CFT_2	Psoriasis

[Download Data](#)

Fig. 17 Example of data included in the Samples table

Most fields, including *sex* or *affection*, have no pre-established codification rules, so users can distinguish male from female or case from control using whatever code they choose. It is in the moment of performing the analysis, if information in these fields is required, that the user will specify which values correspond to each concept.

Since each project has its own particularities and needs, beyond the basic fields that are common to all SNPator *studies* (Fig. 16) the user will have the possibility of adding an unlimited number of additional fields to the *SNPs* and *Samples tables*. This new fields, that we call *descriptors* are added with the help of a user-friendly interface which allows to configure them (See Fig. 18). Once descriptors are defined they will appear in the virtual *SNPs* and *Samples tables* as if they were one of the basic fields and will be treated so throughout the whole application.

It is not reasonable to modify the physical MySQL tables during program execution every time a user wants to add a *descriptor*. To circumvent this problem, new tables were designed to store all *descriptor* configurations and contents. Since *descriptors* and basic

fields have to behave in the same way, a set of functions was created to access the different physical tables and combine them to create virtual tables where descriptors appeared to be ordinary fields of the *SNPs* and *Samples* tables. Functionalities in SNPator do not access database information directly, but operate calling to the functions of this *Data Access Module* and work with the information retrieved from it. Lots of time and resources had to be invested in the development of this data access module but, without it, as SNPator gained complexity, the application would have become chaotic and uncontrollable.

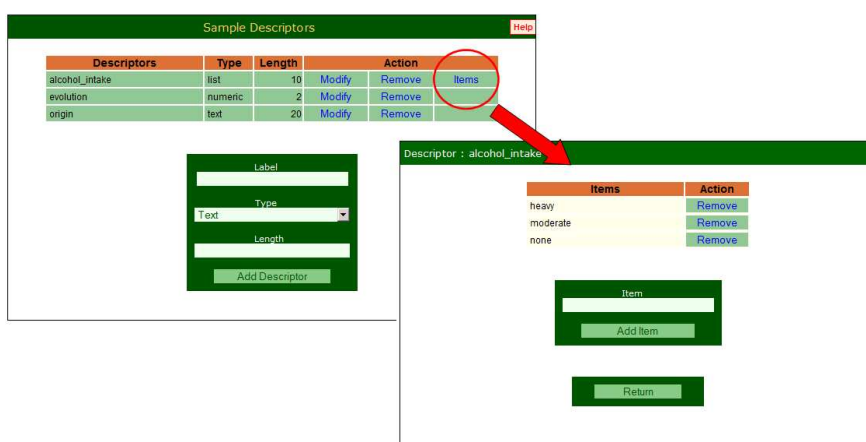


Fig. 18 Descriptors of different types can be easily added to each study.

The last of the three virtual tables perceived by the user is the *Genotypes* table. It contains the results of the genotyping of the SNPs listed in *SNPs* table on the samples of the *Samples* table or at least part of them. Each record in this table has the simple structure “SNP-Sample-Genotype” which means that you need a record for each genotype entered in the system. To organize the table in this way was a very tough decision. We had to choose between compacting the size of the data stored and making, as a consequence, the processes of analysis far more complex or, alternatively, to let data size expand as the price for a much quicker and powerful analysis engine. Since easiness and speed were always the main objectives of the project, this later option was selected.

In order to maintain the consistency of data introduced into the system and provide a natural way of data checking, it has been added as a general restriction that no *Genotype* records can exist if they refer to SNPs or samples which are not present in the *SNPs* or *Samples tables*.

2.5 Uploading Data

All data entered into SNPator, whatever their origin, has to be processed and homogenized to consistently fit into the program's data structures in order to allow easy posterior analysis. Entering data into the program, on the other hand, has to be as easy as possible for the users.

Users with a set of processed data, that is, a list of samples and SNPs with associated information and the corresponding genotypes, can download from SNPator a set of modified excel files with embedded macros. After introducing their data into these excels, execution of the macros will generate XML files²⁹ containing those data.

The XML files containing all the information can be uploaded into SNPator in an easy way using the corresponding form (Fig. 19). Data showing inconsistencies will be rejected. Once the information is uploaded, the analysis process can begin.

Data Management / Upload / SNPs

SNPs XML File

Overwrite SNP if already exists

Fig. 19 SNPs XML file uploading form

²⁹ The basics of XML format are described in section "Massive data transfers".

The most usual case, however, is that users have their data in the form of the output files generated by some automated genotyping machine. Those files do not contain clean and unique genotype information but are an unintelligible mix of results, control tests, redundancies and technical issues.

SNPator provides an easy way to perform a quality control of all this data which generates final clean genotypes prepared for the analysis step through its *Data Caring Module*.

2.6 Quality Control. The Data Caring Module

When genotyping projects are planned, there are a lot of standard common sense procedures to try to guarantee that generated data will have certain degree of quality and the genotypes obtained are the real genotypes of the sample.

In projects using familiar data as the case of linkage studies, Mendelian consistency among members of a family provides an easy way to assess genotyping success, besides false paternity issues and lab mislabeling.

In contrast, when working with non familiar data, for example in a population case-control study, a lot of indirect quality estimators and procedures have to be set up. Among them stand Hardy-Weinberg equilibrium, repeated genotyping of some samples to check consistencies, genotyping of samples with known values, etc.

The output files produced by genotyping devices will be full of repetitions, inconsistencies, virtual SNPs included for technical reasons, and so on. Independently of the analysis method or software to be used, all this information has to be processed and cleaned. In the case of SNPator, as said before, the *Genotypes virtual table* has to be clean, ordered and consistent to allow all the analysis processes to go smoothly.

That is why the Data Caring module has been added as a previous processing pipeline for the genotype information before reaching the analysis zone (Fig. 20).

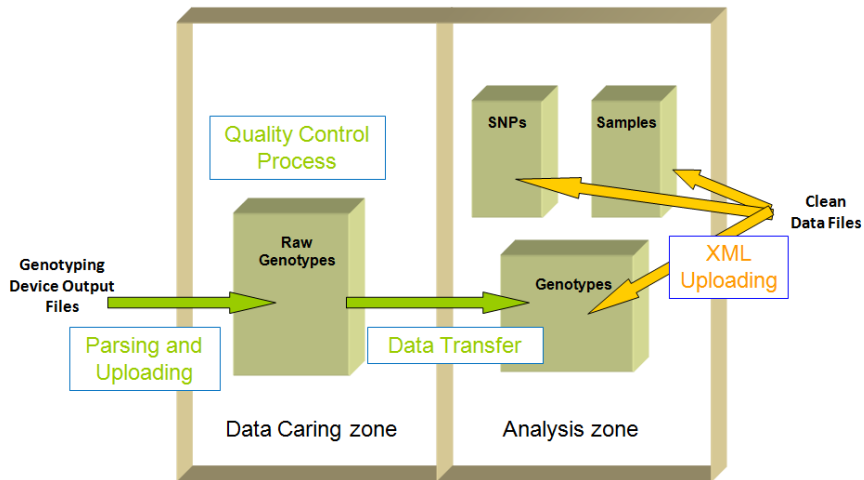


Fig. 20 The two paths for uploading genotypes into SNPator. Clean data can be uploaded directly to the Analysis zone. Genotyping devices output, however, have to be uploaded to the Data Caring module and only after processing can be transferred to the Analysis Module.

Data caring works as follows. When working with a genotyping platform, work is usually divided into different physical plates for processing it. The physical nature of each plate depends on every technology, sometimes it is a chip array with hundreds of thousands of SNPs genotyped simultaneously for a single sample, other times it is a combination of SNP probes and multiple samples, and so on. At the end of the process, each technology will generate output files containing the results of the plates. And a set of plates constitute a whole project.

SNPator has been prepared to be able to read the formats of output files of widely used current technologies. Users will upload the files to a *Study*, combining, if they want to, different technologies and even adding other genotypes from a different origin (public databases, for example) that they want to be analyzed together (Fig. 21).

Data Management / Genotypes / Plates / Upload

Technology

Plate

Genotyping date

File

Taqman Coding File

Translation File

Fig. 21 A view of the Plates Upload form showing a list of technologies whose output files SNPator can read and process.

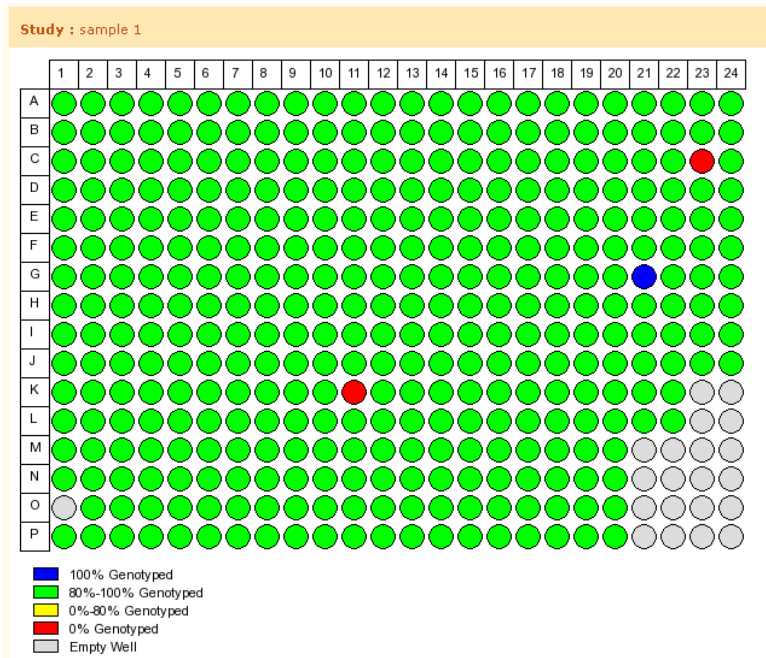


Fig. 22 Graphical visualization of the genotyping success of a plate uploaded into SNPator

Parsers have been developed in SNPator to interpret output files from technologies used in CeGen centers together with others that users have been suggesting. New ones are being developed as the need for them appears.

All the information from the plate files, with all its repetitions, contradictions and unnecessary technical details is processed and stored into a table called *raw genotypes*.

Once the information is there, users can begin the quality control procedures. They can visualize, for instance, the genotyping success of samples included in a certain plate to try to pinpoint patterns in the appearance of problems (Fig. 22)

Reports can also be generated for a single plate, all of them or a particular combination of plates, with descriptive statistics of selected data (Fig. 23).

```

Plate/s Information
-----
Plate       : SNPlex
Study      : sample 1
User       : advanced
Date       : 2005-08-04 17:14:32
Technology : SNPlex
Source     : Placa.txt

Final Report
-----

Genotypes ....
-----
Unique genotypes      : 16225 (96.97%)
Empty genotypes       : 461 (2.76%)
Concordant repetitions : 46 (0.27%)
Non concordant repetitions : 0 (0.00%)
Total genotypes       : 16732
Total tested          : 100.00 %
Total genotyped       : 97.24 %

SNPs ....
-----
SNPs with more than two alleles : 0(0.00%)
Total SNPs                     : 47

Samples ....
-----
Total samples                : 356
...

RS1054936  99.17% |*****|
RS12437400 99.45% |*****|
RS12579353 99.45% |*****| (GG)
RS1325290  99.45% |*****|
RS1414482  99.45% |*****|
RS145527   99.45% |*****|
...

```

Fig. 23 Plate Report

A set of tools is at disposal of the user to check the quality of the genotypes uploaded.

To check whether replications worked correctly, for example, SNPator can look for all contradictions among tests for the same genotype that may arise in the data and provide an interactive graphical way to solve them (Fig. 24).

The screenshot shows the SNPator interface. At the top, there are controls for 'All Plates' (set to 'Plate 001'), 'Show All', 'Add Report', 'Report Only', and a 'Go' button. Below this are color-coded status indicators: 'Genotype ready' (green), 'Genotype not ready' (orange), and 'Genotype modified' (blue). A 'Show/Hide Warnings & Notices' button is also present.

The main section is titled 'Non coherence repetitions (errors)' and contains a table with the following data:

SNP	Sample	Action
rs10888558	d341	Modify
rs10888558	na10860	Modify
rs17859389	d341	Modify

A red arrow points from the 'Modify' button for the second row to a detailed view window. This window is titled 'Data Management / Genotypes / Quality Control / Coherence' and shows details for 'SNP rs10888558 , Sample na10860'. It contains a table with the following data:

Plate	Well	User	Date	Technology	Genotype	Score	Cancelled
Plate 001	M21	advanced	2009-11-04 17:24:36	SNIPlex	CC	0.89	<input type="checkbox"/>
Plate 001	N21	advanced	2009-11-04 17:24:36	SNIPlex	CG	0.9938	<input type="checkbox"/>
Plate 001	M22	advanced	2009-11-04 17:24:36	SNIPlex	CG	0.9969	<input type="checkbox"/>
Plate 001	N22	advanced	2009-11-04 17:24:36	SNIPlex	CG	0.9895	<input type="checkbox"/>

At the bottom of the detailed view is a 'Change' button.

Fig. 24 Visual tool for reviewing and solving contradictory results for repeated tests. SNPator shows the details of every test and allows changing them until consistency is reached.

Considering the many cases in which HapMap (International HapMap Consortium 2005) samples have been used as controls, the entire set of HapMap data is stored locally in our servers and, if the user selects this option, SNPator will check the consistency of the results in the user's *study* with those of the public databases. As in the former case, simple graphical tools will allow the contradictions to be solved.

When the person in charge for the cleaning and processing of the genotypes, whether it is an external user of SNPator or the technical services of CeGen processing data of some client, considers that genotypes are clean and consistent, they can be transferred into the "Analysis zone" of SNPator. SNPator only allows this transfer after checking and confirming the consistency of the data.

2.7 Validation. Deciding the Ploidy.

There is a data-management problem associated with ploidy. Some genotypes are diploid, that is, simplifying, they come from autosomic chromosomes in humans so each individual carries 2 alleles for each genotype. Some are haploid, that is, they come from sex chromosomes and carry only one allele. With all added complications, of course: X chromosomes are diploid in women and haploid in men except in pseudoautosomal regions, some users will use non human SNPs, etc.

Since most analysis take into account the ploidy and act according to it, it was decided that all genotypes in SNPator will be recorded as single (e. g. "A", "T") or double (e. g. "AA", "AT") and that all the processes working with these genotypes will behave differently in one case or the other.

The problem arises because most genotyping machines do not distinguish homozygous ("AA" in an autosomal SNP) from hemizygous ("A" in a sexual SNP). That is so because their techniques are prepared to detect the presence/absence of a certain marker of each allele but cannot quantify it with accuracy.

Thus, when plate output files are uploaded to the *Data Caring zone* of SNPator, they will present genotypes in a confusing way depending of the technologies, "A" and "AA" meaning the same thing which can be either "A" or "AA" depending on the SNP and the technology.

There was a lot of deliberation on this issue and the final solution adopted is far from satisfactory, but no nice exit can be found to the

trap posed by the lack of definition of the results obtained from genotyping technologies.

The user is forced, before beginning any analysis of the data, to decide the ploidy of the genotypes with the help of some tools developed to this end. We call this process *validation*. For the simplest cases (all data are diploid, for instance) *validation* can proceed quickly, just stating the type of ploidy for the whole data set (Fig. 25). For more complex situations, if chromosome information for SNPs and sex information for samples is introduced, SNPator will update the ploidy status of genotypes when requested. Any exceptions to the general rule can be modified afterwards in a case-by-case procedure.

Genotypes Validation / Perform Validation / Global

a) All genotypes belong to : Autosomes ▼
[Select item]
Autosomes
Y or MT
X

b) Validate using SNP and Sample Data.
Male : [Select item] ▼
Female : [Select item] ▼

c) Validate all genotypes with no change.

Ok

Fig. 25 Genotypes Validation Form

Only once the haploid/diploid definition of each genotype has been stated users can begin to exploit all the analysis possibilities of the application.

There is, however, a single procedure that users can perform with non validated data: they can download them from SNPator. This

allows CeGen clients to retrieve their project results to be processed outside the application. It's understood that in this case, A/AA differences are not crucial, since experienced users that can process their own data are surely aware of the problem.

2.8 The User Results Section.

Whenever SNPator users work with the application, downloading data, performing analysis, and so on, they will be generating lots of information in different formats that somehow have to be transferred to them. Unlike typical local running applications like Excel or SPSS, which store all generated data as files in the local disk, a web application like SNPator cannot do the same thing because of asynchrony³⁰ and security issues.

Asynchrony, because most processes are executed in background and when they finish the user who launched them may not be connected anymore to the website. Security, because web programming languages do not allow a web page to write directly into the user's computer. Otherwise, every page opened with a browser on the Internet could fill the computer with viruses and trojans.

A *User Results section* has been designed as a common interface for all kind of information that SNPator generates for the user.

Every time SNPator finishes a requested action, a set of files with the corresponding information will be created. All these files will be packed together into a compressed file, coded into a base64 representation to avoid special characters and stored in a particular table of the database. Users will be able to access them using the *User Results section* (Fig. 26). Of course, all privacy issues referring the access privileges apply to results in the same way than to the rest of data of a *study*.

Users can consult all the information stored here whenever they want and download it locally into their computers. All items will be kept here indefinitely until users decide to remove them.

³⁰ Asynchrony, in programming, means the simultaneous execution of different processes instead of following a strictly sequential order.

Furthermore, one of the most valuable features of this section is that in case of multi-step analysis procedures, SNPator can use results from previous analysis stored here as input for further analysis down the pipeline.

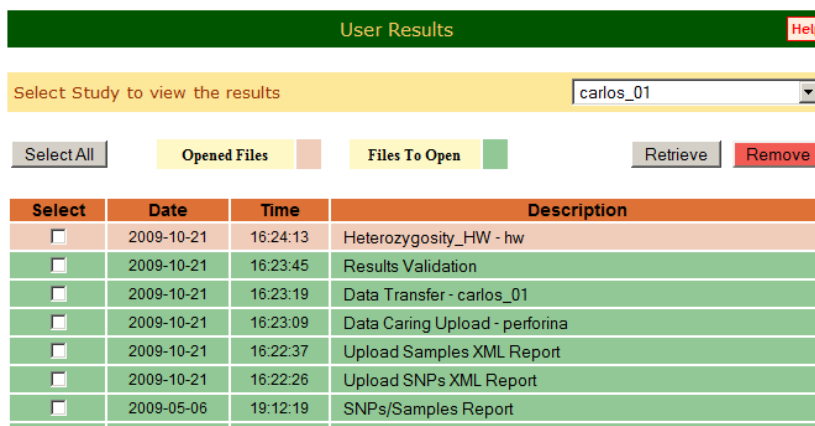


Fig. 26 User Results Section

2.9 Filters

From the feedback of users working with early version of SNPator, it became clear that having an effective system to filter and segregate the particular set of information that is to be analyzed was a crucial requirement for this kind of application.

A filter can be understood, from the user's point of view, as a formal description, usually taking the form of a Boolean expression, of a criterion to select a subset of data. On the other hand, from the application's point of view, a filter is the list of the actual data that satisfy the selection criterion.

Therefore, the creation of a filtering system will need the following components:

- An interface that allows users to enter the Boolean expressions that define the selection criteria of the filter.
- A system to map the previous logical expressions to actual data

- A system to record the data selected in the previous step and a mechanism to use these data when the filter is activated.

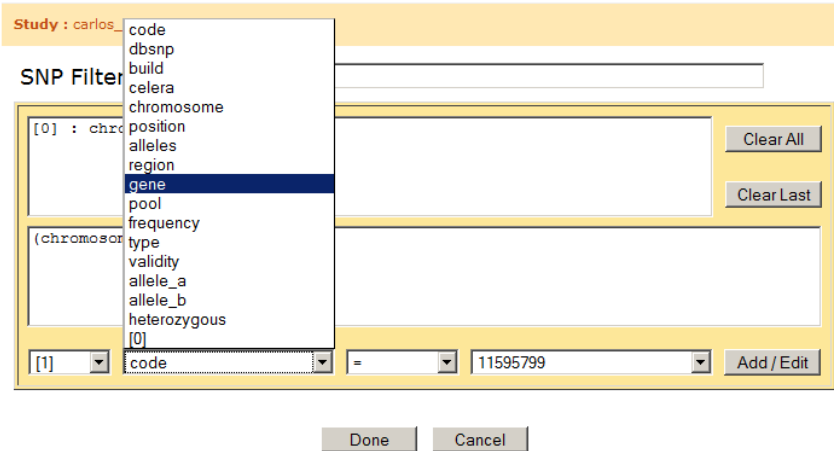


Fig. 27 Definition of logical statements to be used in filters

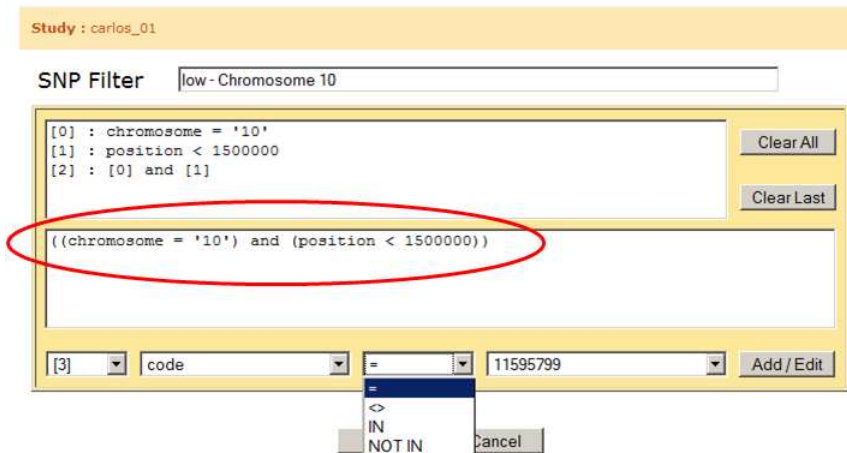


Fig. 28 Combination of logical statements to create a complex query.

2.9.1 The Boolean Interface

Since a *study* in SNPator is a list of SNPs and samples and their resulting genotypes, a filter will be a subset of SNPs and samples and their genotypes.

An interactive interface has been programmed that allows building Boolean expressions of unlimited complexity that define the selection criteria for SNPs and for samples separately. Fig. 27 and Fig. 28 show how logical SNP defining statements can be visually composed and combined to create final complex Boolean statements. In the program, the resulting expressions are referred to, quite unfortunately, as *SNP* and *Sample subsets*.

2.9.2 Creating the Filter

The mapping between the logical expression and the current data is the key point in the strategy to create a filter system. There are basically two approaches.

In the first one, only the filter description is stored in the system when it is created and it is in the moment of its use that the actual selection of data is done. The advantages of this approach are that creation of filters is immediate, that no computing has to be done for filters that are not used and that the number of filters can be practically unlimited. However, at the moment of the execution of any analysis that uses a filter, this strategy imposes a penalty in the performance. This would be particularly harmful in the case of SNPator since the virtualization of the data through the use of the *Data Access Module* would add a further delay in the execution.

That is why a second model in the management of filters was used in SNPator. Unlike the previous one, in the moment of the creation of a filter, all data is marked as belonging or not to this filter and afterwards this marks are used as a quick way to retrieve the data when needed. In this way, analysis and data retrieval work with SNPator can be done quickly even using filters and the user's subjective feeling using the application improves. The disadvantage is that, when creating the filter a lot of time is needed because all data of a *study* (even those which are never going to be used) have to be marked. But since filter creation can be let

working alone in the background, this time should not be considered as crucial as the analysis performing time.

Once the *SNP* and *Sample subsets* have been defined, they are used in the creation of filters. In the *filter Creation section*, users can select the subsets that define the data they are interested in and they can also set a minimum genotyping success for SNPs and samples to be included in the filter (Fig. 29).

Study : carlos_01

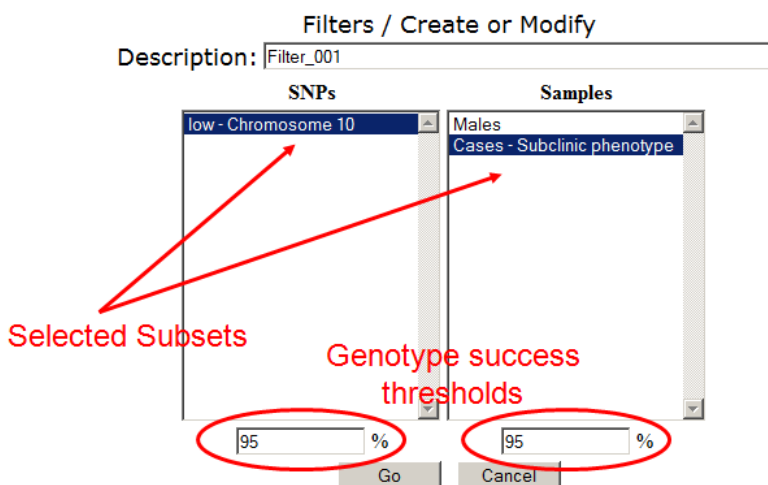


Fig. 29 Creation of a filter

Calculating the SNPs and Samples that are affected by the genotype success threshold is surprisingly tricky.

As an example, let's suppose a study composed by 100 SNPs and 100 samples (i.e. 10,000 potential genotypes) where 94 SNPs have worked perfectly (they have genotypes for all samples) and 6 of them have completely failed (0% genotyping success). A filter of 95% threshold for both SNPs and samples is defined. The result will dramatically vary if SNPator begins the filter building process with SNPs or with samples as shown in Table 1.

Beginning with SNPs:	Beginning with samples:
6 SNPs with 0% are excluded 94 SNPs with 100% are selected Now, taking into account only the remaining SNPs, 100 samples are selected with 100% genotyping.	100 Samples are excluded with 94% (under the 95% threshold)
Final result: 94 SNPs / 100 Samples included in the filter	Final result: No data is included in the filter.

Table 1 Genotyping success filtering can behave differently depending of the algorithm used.

To solve this problem a round-robin algorithm was created that looks for the sample or SNP with the lowest genotyping success value, excludes it and recalculates SNP and sample genotyping success values for all remaining data. The process is repeated until no SNP and samples are left below the exclusion threshold.

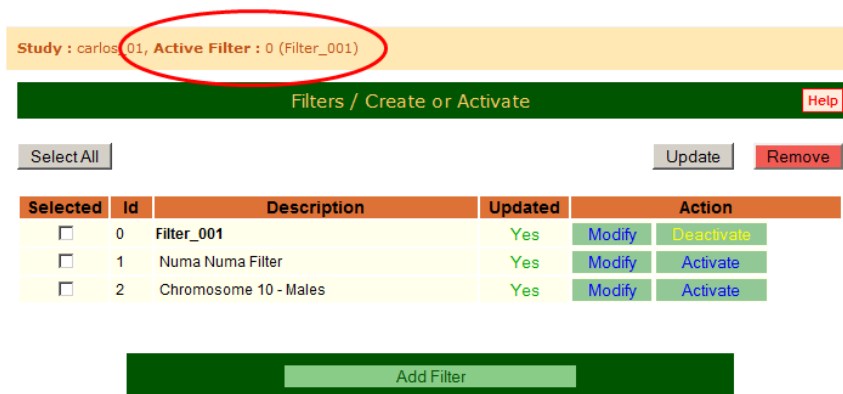
2.9.3 Marking the Data

The *SNPs*, *Samples* and *Genotypes* tables in SNPator have a 64-bit length field called *Filter*. Each bit of the field in a record indicates if that record is included in the corresponding filter. To illustrate it, if the 24th bit of the *Filter field* in the record of SNP1 is = 1, then the filter number 24 contains SNP1. If it is =0, then the SNP1 is excluded in that filter. The 64-bit length limits the number of filters simultaneously defined in SNPator to 64. *SNP* and *Sample subsets* have no limitation in their number.

The process of creating a filter marks all *filter* fields for all data in the *study*. The time it will take will depend on the size of the *study* and, for that reason, the process runs in background and, once completed, a report is added into the *User Results* section.

2.9.4 Using the Filters

All filters created will be accessible in a *Filter Management section* (Fig. 30). Here they can be activated, deleted or modified. When a filter is activated, all operations performed with SNPator will apply only to SNPs, samples and genotypes included in that filter. Only one filter can be active at any given time, but this poses no problem, since complex filters can be constructed that are the combination of any set of previously built logical expressions.



Study : carlos_01, Active Filter : 0 (Filter_001)

Filters / Create or Activate Help

Select All Update Remove

Selected	Id	Description	Updated	Action
<input type="checkbox"/>	0	Filter_001	Yes	Modify Deactivate
<input type="checkbox"/>	1	Numa Numa Filter	Yes	Modify Activate
<input type="checkbox"/>	2	Chromosome 10 - Males	Yes	Modify Activate

Add Filter

Fig. 30 Filter Management Screen

The *Data Access Module*, which centralizes all accesses to the database, takes into account the active filter when returning data to the request of a process by means of applying a bitmask to the field *filter* in the queries that it builds.

2.10 Batch Mode

The *Batch Mode* is an extension of the filtering system that allows users to have a process executed repeatedly selecting, for each execution, different subsets of data. The *Batch Mode* option is available in almost every data retrieval, format or analysis option in SNPator where it makes sense, and can be applied to SNPs, samples or both.

Here is an example using some basic heterozygosis and Hardy-Weinberg equilibrium analysis. When requesting the analysis in the

corresponding window (Fig. 31), if no *Batch Mode* is selected, all data in the *Study* that are included in the active filter is processed together. Thus, Heterozygosis and HW statistics are written into a file which is compressed and stored into the *User Results* section.

Study : carlos_01, Active Filter : 0 (Filter_001)

Basic Analysis / Heterozygosity - HW Equilibrium Help

Description

SNP Batch Attribute

Sample Batch Attribute

Send an e-mail when job finished to :

Fig. 31 Hardy-Weinberg Analysis Section

Study : carlos_01, Active Filter : 0 (Filter_001)

Basic Analysis / Heterozygosity - HW Equilibrium Help

Description

SNP Batch Attribute

Sample Batch Attribute

Send an e-mail when job finished to :

Fig. 32 *SNP Batch Mode* option in the Hardy-Weinberg Analysis Section

In contrast, if the *SNP Batch Mode* option is selected and the *gene field* is selected from the associated combo box (Fig. 32), SNPator will look at the content of this field in all SNP records, will list the different values that it presents and will perform a separate analysis for each value, taking only into account those SNPs which contain that value. All reports generated are put together into a compressed file and stored, as usual, in the *User Results section*. Of course, only data selected by the active filter is taken into account.

The same process applies to *Sample Batch*. If both *Batch Modes* are activated as many processes are run as the combinations of the values of the fields selected.

2.11 Retrieving Data

Data downloading mechanisms in SNPator have been developed with two basic scenarios in mind. The first one is related to CeGen clients. After the corresponding lab has genotyped the samples provided by a client, has uploaded the results into SNPator and performed all the quality control procedures, the user gets a username and a password to log into SNPator and download the results.

If the user is interested in his or her data without any particular format, there are some basic file options (lists, matrix, tab separated, space separated, etc...) that allow downloading of SNPs, samples or genotypes data. If no filters are used and no validation is performed, all data present in the study is retrieved by the users. They can stop here the use of SNPator and continue their project using their own analysis software.

The second scenario involves taking advantage of the file formatting functions of SNPator. There are a lot of different genetic analysis programs that use diverse file formats. To have an automated way to create input files for those programs helps to save time and, as important, to avoid trivial processing human errors.

Data can be retrieved formatted as input for some of the current most popular software analysis (PLINK, PHASE, Linkage, Haploview...). That allows the users to upload data once, clean it, validate it and afterwards, using filters and batch mode, to create with no effort lots of input files prepared to work with external software.

In this way, the data management capabilities of SNPator can be used for purposes other than those which are actually implemented in the analysis structure of the Application. However, although this is an important help for genetic statisticians, they will still have to install, configure and run different programs with all the time investment that it requires. SNPator is designed to take this burden off the shoulders of the final user, which takes us to the data analysis options.

2.12 Analyzing Data

The primary goal in the design of SNPator has been to allow users to forget about all computer-related cumbersome tasks and make it possible for them to just give a command, let the software do all the work and be finally informed once the results are available in the *User Results* section.

In order to fulfill this objective as much as possible, a set of analysis options have been included in SNPator. They range from very basic statistical calculations like frequency counts or HW testing to more complex disease or genomic analysis.

Some of these options have been programmed as PHP code from scratch. For others, the approach has consisted in taking advantage of already working and well established methods implemented in outstanding, publicly-available software and integrating them into our platform.

For instance, when estimating haplotypes from genotype data for disease analysis purposes, the software PHASE (Stephens et al. 2001) is frequently used. PHASE is installed in our servers and what SNPator will provide when running haplotype estimations will be a transparent gateway to use this software.

What that means is that, when a haplotype estimation is requested, SNPator will access the database looking for the data that is needed (taking into account the options selected and any filters that may be active) , will create all input files needed by PHASE from these data, will execute PHASE in the server, will wait until the process ends, will retrieve all output files from the run, will parse them to create statistics and new output files with formats prepared for possible further analysis and, finally, will put everything into a zip file and store it into the *User Results* section.

All the process is transparent for the user. Independently of whether it is a simple internal calculation or a complex call to an external program, the experience for the user is always the same: "select options, ask for results and get them".

Of course, PHASE, like other programs used in our system, is a licensed product with its own rules of use. SNPator is nothing more than a gateway for using this program and so, in order to use this option every user has to ask for a regular license through the corresponding procedures. Acceptation of this fact is necessary in order to get an account in SNPator.

2.13 The Calculating Machine

SNPator is available in the web. Hundreds of people have users and passwords to access it and to make it work (detailed statistics about SNPator usage are provided below). There may be idle moments when few or no people are connected. However, the most common scenario is that many concurrent users are demanding that the application performs many different calculations at the same time. This threatens to overload the system, to delay its response time and, in the most extreme situations, to crash the system.

To avoid this scenario, a quite complex system has been designed and implemented to distribute the computer processing load which, at the same time, allows for a quick and easy scalability.

The technical characteristics that such a system needs to have are diverse. First of all, the system has to be available for any user

who wants to use it at any moment and, most importantly, its time response has to be satisfactory. It should never happen that, because a former user entered the system and asked for a lot of very long and resource consuming tasks, the following user is not able to log in or, in case he or she succeeds in logging in, the system is slow and malfunctioning. If somebody asks for complex things, it may well take long to perform these complex things but it must not, under any circumstances affect other users.

Second, the system has to organize all the tasks demanded by users in a sequential way instead of trying to execute all of them instantly. It has to keep track of these tasks and their progress, guarantee that all of them will be done and that the user will get the results. Executing policies have also to be implemented. Are tasks to be executed following exactly the order of requests or, on the contrary, small tasks requested by people that have nothing in the waiting list are to be prioritized?

Third, the system has to be easily scalable. When the number of users grows and current hardware and resources happen to be insufficient, it has to be possible to just add new computers that improve the power of the system without having to make any big changes in the current configuration, since that could entail interruptions in the service and inconveniencies to the users.

If the system is stopped, not only users may not be able to log for a while, which is undesirable but not fatal. The big problem with the system stopping is that this procedure can kill complex processes that have been running for a long time, causing important inconveniencies to users and undermining their confidence in the system.

From the user perspective, when some task is asked to SNPator, it will take the form of a job waiting to be executed. All jobs that a user has requested are listed and can be consulted in the *Jobs section* containing information about status, request time, execution time etc... (Fig. 33 and Fig. 34)

Jobs							Help
Select Study to view the results						carlos_01	
Waiting/Processing		Finished		Remove			
Job	Description	Status	Request Date/Time	Start Run Date/Time	End Run Date/Time	Action	
Phase	Haplotype estimation	waiting	2010-01-26 17:28:12			Cancel	
Heterozygosity_HW	hw	finished	2009-10-21 16:24:05	2009-10-21 16:24:06	2009-10-21 16:24:12	-	
Results_Validation	Results Validation	finished	2009-10-21 16:23:45	2009-10-21 16:23:45	2009-10-21 16:23:45	-	
Data_Transfer	Data Transfer	finished	2009-10-21 16:23:18	2009-10-21 16:23:18	2009-10-21 16:23:19	-	
Data_Caring_Upload	perforina	finished	2009-10-21 16:23:07	2009-10-21 16:23:07	2009-10-21 16:23:09	-	
SNP_Samples_Report	SNPs/Samples Report	finished	2009-05-06 19:12:13	2009-05-06 19:12:13	2009-05-06 19:12:19	-	
Genotypes_Matrix	Matrix Format Report	finished	2009-05-06 19:10:11	2009-05-06 19:10:11	2009-05-06 19:10:16	-	

Fig. 33 Jobs Section. All jobs are listed with execution details.

In this section the progress in the execution of jobs can be followed and, if necessary, mistakenly requested tasks can be cancelled.

Jobs							Help
Select Study to view the results						carlos_01	
Waiting/Processing		Finished		Remove			
Job	Description	Status	Request Date/Time	Start Run Date/Time	End Run Date/Time	Action	
Phase	Haplotype estimation	parsing	2010-01-26 17:28:12	2010-01-26 17:28:20	2010-01-26 17:28:30	-	
Heterozygosity_HW	hw	finished	2009-10-21 16:24:05	2009-10-21 16:24:06	2009-10-21 16:24:12	-	
Results_Validation	Results Validation	finished	2009-10-21 16:23:45	2009-10-21 16:23:45	2009-10-21 16:23:45	-	
Data_Transfer	Data Transfer	finished	2009-10-21 16:23:18	2009-10-21 16:23:18	2009-10-21 16:23:19	-	
Data_Caring_Upload	perforina	finished	2009-10-21 16:23:07	2009-10-21 16:23:07	2009-10-21 16:23:09	-	
SNP_Samples_Report	SNPs/Samples Report	finished	2009-05-06 19:12:13	2009-05-06 19:12:13	2009-05-06 19:12:19	-	
Genotypes_Matrix	Matrix Format Report	finished	2009-05-06 19:10:11	2009-05-06 19:10:11	2009-05-06 19:10:16	-	

Fig. 34 Jobs Section. Job status is updated as the process advances.

Once a job is finished, it is marked as such in the *Jobs Section* and all its results will appear in the *User Results Section*.

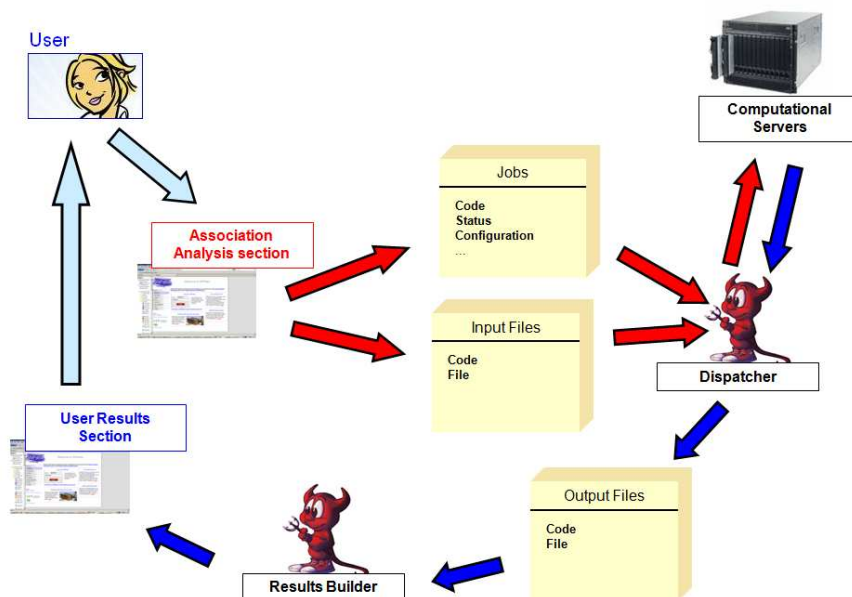


Fig. 35 The basic workflow of the process that each job follows inside SNPator from the moment when the user initiates it to the delivery of the final results.

This simple process is implemented through a multi-step internal workflow whose basic details are depicted in Fig. 35. The image depicts an example where a user logs into SNPator, selects the *study* she wants, begins to work with it and decides that she needs an association analysis of her data.

Using the menu, she enters, into the *Association Analysis section*, fills all check and combo boxes to define the parameters of the analysis and presses the *Go button*. At this time, an Association Analysis process of SNPator is launched. The process goes to the database and, using the *Data Access Module*, takes all data from the *SNP*, *Samples* and *Genotypes* virtual tables needed for the analysis as defined by the user specifications. All this information is written into a set of text and XML formatted files.

Once this is done, the process registers a job to be run into the *Jobs table*, and inserts all the files needed for this job into the *Input Files table*. In the *Jobs table* there is an entry for each job, and in the *Input Files table* there will be an entry for each file with the *Code field* linking both tables and identifying which files belong to a particular job. The process stops here. It has written all information needed for the analysis into the two before mentioned tables but no association analysis was done yet.

Now it is the turn of the *Dispatcher*. *Dispatcher* is a daemon. That is, it is a process that never dies. It is always running reading the *Jobs* and *Input Files* tables and if there is something there that can be executed, it processes it, otherwise it goes on reading. There are other daemons that watch closely this *Dispatcher* and restart it in case it accidentally dies. In case they cannot restart it, they send an e-mail to the system administrator.

While reading the tables, *Dispatcher* sees that there is an association analysis waiting to be performed. If there is some Computational Server free at this moment to do this kind of analysis, it takes all data related to this job and sends it to the server. The Computational Servers will process it and, when everything is over, will send a set of files (also text or XML coded) back to the *Dispatcher*. (Details of *Dispatcher* communication with the Computational Servers are discussed in the next section). The last thing that the *Dispatcher* will do with the current analysis is to write all these result files into the *Output Files* table.

Now is the turn for another daemon to intervene: *Results Builder*. It has the purpose of reading once and again the *Output Files* table and if something is found there to process it. *Results Builder* will take the output files, will see that they belong to an association analysis job, and will process them in order to create the final files that are going to be written in a zip compressed form and inserted into the *User Results* section. The user can now access her results.

All the participants in this workflow, the Association Analysis process, the *Dispatcher* and the *Results builder*, update the *status field* in the record of this job in the *Jobs* table. It is using this

updated information that the user can follow the evolution of her processes in the *Jobs section* (Fig. 33).

2.14 Computational Servers. Web Services

The development of the communication system between the *Dispatcher* daemon and the Computational servers (Fig. 36) posed several challenges, some of which had to be addressed with non standard approaches.

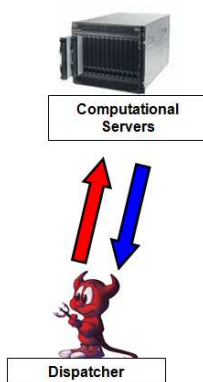


Fig. 36 Communication between *Dispatcher* and Computational servers is based on web services.

This communication is based on web services. A web service is a protocol of internet-based communication for the interchange of information between different machines without human intervention and using the same standard channels than the web so that no ad hoc infrastructure has to be created.

A web service takes the form of a communication between a client (the machine which asks the service) and a server (the one which provides it) using the open SOAP³¹ protocol, which interchanges information using XML coded files. Each web server definition has a list of requests that the client can make to the server and the

³¹ <http://www.w3.org/TR/soap/>

corresponding answers of the server. These requests are called methods.

The objective of web services is to distribute tasks through the internet and retrieve services from external software in an easy way. There are thousands of web services available in the Web. Google, for example, can be used as a web service³². That means that it is possible to create a program that makes automatic queries to Google and processes their results with no web browser, no clicking and no human intervention whatsoever.

In our structure, web service technology is used in a slightly different way. We are using it not to distribute services in the web, but to distribute the processing tasks of our system into different machines.

All calculation tasks that SNPator can perform are coded as web services that are installed in our *Computational servers*. The *Dispatcher* daemon will act as a web service client and will ask them to perform the tasks required.

A common set of methods have been developed to create the interaction between *Dispatcher* and *Computational servers* (Table 2).

When the *Dispatcher* daemon, who reads non-stop the *Jobs table*, sees a new job to be executed, it checks what type of analysis the job requires and looks up in its configuration which *Computational Servers* have the corresponding web service installed. Using the different methods of the web service, *Dispatcher* will look for a free machine and will send the task to execute. Afterwards, it will ask every few seconds if the task is already finished and, once it is, it will retrieve the output files from the web service and insert them into the *Output Files* table. There, as said before, the files are detected by the *Results Builder* daemon, processed and inserted into the *User Results* section.

³² <http://channel9.msdn.com/coding4fun/articles/Using-the-Google-Web-Service>

Name	Parameters	Answer	
busy	usr pwd	yes/no	Client asks if it is possible to run a process. Every server has a maximum number of processes that can run simultaneously. "Yes" means that it can accept a new execution, "No" means that it cannot.
execute	usr pwd, configFile, Files	id	Client sends data to execute the process. Server begins its execution and returns an ID code to identify it.
finished	usr, pwd, id	yes/no	Client asks if a certain task has already finished.
retrieve	usr pwd id	Output File	Client request the output files of the execution of the process. Server sends them.
cancel	usr pwd, id	yes/no	Client asks for the cancelation of a currently running process. Server tries cancelation and informs if cancelation was possible.

Table 2 List of web service methods of the interface developed for the interaction between *Dispatcher* and *Computational Servers*.

For every conversation between client (*Dispatcher*) and server (Computational Server) a user/password authentication is used. This authentication is programmed into the web services protocol because, although they are being used internally exclusively for SNPator requests, they are prepared to be published in the web to be accessed by external software.

Now we have the whole picture of the job management system of SNPator. For all tasks that SNPator can do, there is a web service installed in one or more machines that will do it. *Dispatcher* manages all pending jobs, sends them to the *Calculation servers* depending of their kind and keeps track of all jobs running to retrieve the results when finished.

What are the advantages of this structure? First, it sends all processor intensive calculations outside of the SNPator web and

database server, which will stay fresh and prepared to respond to connecting users' needs.

A second advantage is that, by performing calculations away from the SNPator server, we avoid any jam caused by running processes and we can control their execution priorities. Jobs wait to be executed as entrances in MySQL tables until there are resources for them. Jobs are executed mainly following antiquity, but users with no running processes will have priority over users with jobs already working in the system.

A final advantage is that the scalability of the system is easy and limitless. Let us consider a case where there are 3 machines serving a certain web service with each machine offering 4 simultaneous executions. That means that for this particular type of analysis 12 calculations can be performed at the same time and that may be enough for the current workload. If demand for this analysis increases significantly, there will be a queue of jobs waiting for execution and delays will diminish the user satisfaction with the application. The solution in a centralized system (one where everything runs in a single machine) would be very complex and would include buying a new big machine, installing all the software structure and migrating from the old to the new computer. And, of course, cope with all unforeseen problems that always arise in this kind of interventions.

In contrast, with the system here described, everything is very easy. A new computer is bought and the web server is installed in it. The configuration of *Dispatcher* daemon is changed to let it know that a new machine is available and, automatically, without touching anything else, tasks will be sent to the new server. And with no interruption whatsoever our system can be scaled to infinity, since we only have to keep adding new machines when needed and funds are available.

If a Computational Server becomes obsolete and has to be retired from service, this can also be done without disturbance. *Dispatcher* is informed not to send new jobs to the obsolete server and, when the last one ends, the machine can be turned off with no further

problem. The same thing applies when a machine needs maintenance service or reparation.

2.15 Massive Data Transfers

The data flow and architecture above described is quite elegant and fulfills most of the needs of a robust system. However, when we tried to implement it, we encountered a serious problem with the amounts of data it involved.

The communication of web services is based on an exchange of XML files containing all the information needed for the protocol. XML files are text files where the information is structured and delimited using predetermined tags (marked between "< >"). An example of XML content file:

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes" ?>
<!DOCTYPE samples [
  <!ELEMENT samples ((sample)*)>
  <!ELEMENT sample
(code?,family?,father?,mother?,sex?,age?,afection?,cate
gory?,disease?)>
  <!ELEMENT code (#PCDATA)>
  <!ELEMENT family (#PCDATA)>
  <!ELEMENT father (#PCDATA)>
  <!ELEMENT mother (#PCDATA)>
  <!ELEMENT sex (#PCDATA)>
  <!ELEMENT age (#PCDATA)>
  <!ELEMENT afection (#PCDATA)>
  <!ELEMENT category (#PCDATA)>
  <!ELEMENT disease (#PCDATA)>
]>
<samples>
  <sample>
    <code>
      NA06985
    </code>
    <family>
      1
    </family>
    <sex>
      M
    </sex>
    <age>
```



```
    46
    </age>
    <afection>
      N
    </afection>
    <category>
      Ciudad
    </category>
    <disease>
      Lupus
    </disease>
  </sample>
```

...

XML files are useful because they structure and self-define the information contained in them. But, as the above example clearly shows, they can reach enormous sizes since, for every item, they repeat all the tags required to define it. The same data in the form of a text table would just need to be labeled once every row and every column.

The problem comes from the fact that, usually, web services are thought to transmit limited amounts of information and in this situation the XML coding structure is not a problem. But in our case web services are used for internal architecture organization and that generates a lot of performance problems because *Dispatcher* and the Computational Servers are moving too many data through the internal network.

This situation created several problems. First of all, there was an evident problem of transmission rate. Information moves very quickly inside a computer between its different components (from memory to CPU, from disk to memory, etc) but when information has to travel through a network the situation is quite different. The multiplication of size that XML code imposes creates a delay in the times of transmission that is clearly undesirable and fails to comply with the quality standards that SNPator is expected to have.

In addition, mere size is not the only problem. If a very big file has to be transmitted through the web services protocol and it takes a

lot of time to complete, it raises the odds that some error may occur that will interrupt the process, forcing it to start again.

Moreover, if a file grows and reach the size of 2 Gigabytes it can cause some instability. This is so because some versions of operative systems impose a maximum size acceptable for a file of 2 GB. We could control and ensure that nothing in all the computers and operative systems in our structure is incompatible with files greater than 2 GB. The problem is that these web services are prepared for being deployed in the future to be used by external independent programs. In this case, a 2 GB file should not be send to a foreign system because some of the multiple devices involved in the process could mismanage it.

To solve in a general way the problems derived from the excessive size of files, a new layer of complexity has been added to the protocol of communication between the web service client (*Dispatcher*) and servers in the Computational Servers (Fig. 37).

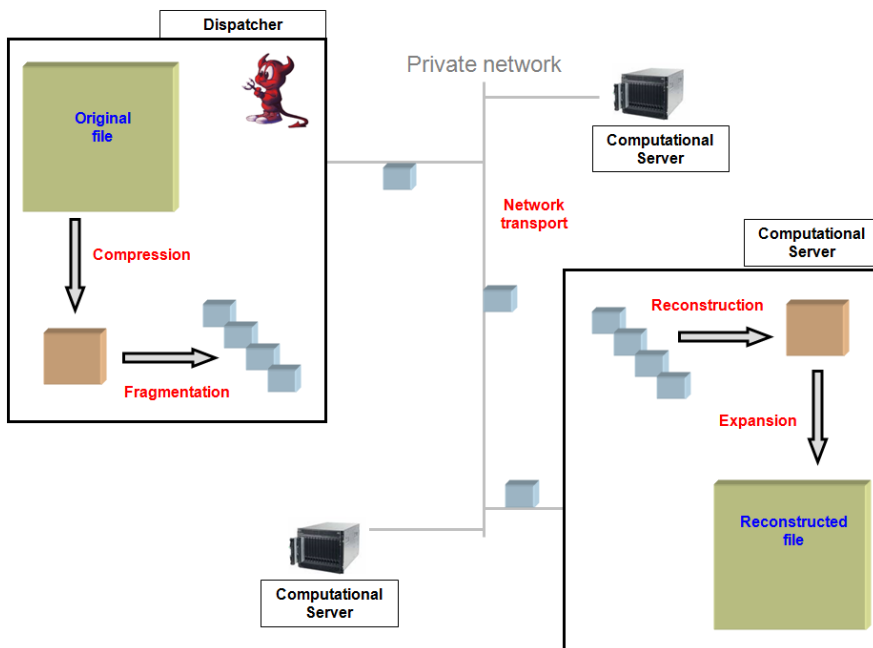


Fig. 37 Files are compressed and fragmented to allow efficient transmission of big amounts of data in the web service communication

Whenever a file has to be transmitted, the client first compresses it to zip form and then cuts it into fragments of 64Mb. These fragmented files are sent to the web service independently. Once they arrive there, the fragments are put together, the resulting zip files are decompressed and the original data reconstructed. This data is used and the calculations performed and when it is time to give back the results the same process takes place in reverse sequence.

2.16 Pipeline Oriented Tasks. Connecting Functions

A genetic study consists usually in a set of steps that are executed sequentially in such a way that results from one step become the input data for the next one, thus producing a particular workflow. Transference of data from one step to the next, when working with different tools, frequently needs of an additional effort to adapt data structures to the formats used by different applications.

Workflow processes are greatly simplified in SNPator. Since results from every action are stored in a known format in the *User Results* section, it has been possible to implement a mechanism for the users to re-use those results as inputs for posterior analysis, of course only for those types of analysis where it makes sense.

For instance, with the help of PHASE, haplotypes that some individuals present for a group of SNPs can be estimated from genotypic data stored in the application. Several estimations can be performed varying the parameters or the involved individuals with the help of filters. Once the results of the estimations are ready, they can be consulted in the *User Results* section.

This estimated list of haplotypes carried by the samples in one study can be used to check for differences in the haplotype frequencies between cases and controls. SNPator offers the option of performing a haplotype association test. The list of haplotypes used in the analysis is usually obtained from a local file indicated by the user in the moment of launching the process. However, there is also the option of selecting a previous estimation stored in the *User Results* section and data will be taken directly from there. (Fig. 38)

Description

Haplotype selection: SNPator data (haploid data only)

User Results

Phase Report File

Tab File

Phase - Haplotype Estimation 1 - filter 85

Phase - Haplotype Estimation 2 - All Populations

Phase - Haplotype Estimation 3 - 95 - European Populations

Minimum score

Send an e-mail when job finished to :

Fig. 38 Haplotype list selection in the Haplotype Association Analysis Section.

In this way, two SNPator functionalities have been connected becoming steps of a usual pipeline, avoiding the cumbersome work of adapting data from the first step output to the input of the following.

2.17 Development - Test - Production. Organizing the Work

Given the service oriented role of SNPator in their functions related with the CeGen institution, interruptions or malfunctioning of the application can have a very negative impact. Therefore, it has always been considered as a high priority to establish an organization of the work that minimizes the possibility and consequences of incidences in the system.

Following standards widely accepted in the industry³³, three stages have been established in the implementation of any change or new feature in the program: *Development*, *Test* and *Production*. Each

33

http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/index.jsp?topic=/com.ibm.wp.ent.doc/wpf/dep_intr.html

stage has its own computers and is completely isolated from the others.

Development is where the programs are created. It is a set of computers to which only programmers access and where code is constantly created and tested. In this environment there is no need of all the control mechanisms, like backup computers, monitoring, etc present in other stages, because incidences here do not affect the public service.

When a functionality or, better, a set of functionalities are considered ready, they are transferred to the next stage: *Test*. The *Test* environment has to be a replica as close as possible to the final public system. In there, changes and new features created in the development stage are tested manually. Non modified functionalities are also tested randomly to ensure that new changes have not affected unexpectedly older parts of the system.

No changes are made in *Test* in order to avoid inconsistencies between stages. If errors are detected, they are solved and tested again in the *Development* environment until they are considered prepared to be transferred again to the *Test* environment.

When the whole *Test* system is considered to work properly, it is labeled as a particular version and it is transferred in its integrity to the final stage: *Production*. From this point on, all modifications are available to the public.

It always may happen that some bugs have escaped testing in the previous stages and appear in *Production* because a group of many users carrying out work end up doing things that testers and programmers cannot even imagine or because their data structures are different than those used in the tests. In any case, when a user reports a bug and it is confirmed as such, an incidence is created with a "bug number" assigned and it is repaired in the *Development* environment, transferred to *Test* and, when considered ready, promoted again to *Production*. This is the better way to keep the consistency of the system.

Sometimes things are not that easy. For example, a bug can appear in a functionality that has been further modified in

Development and *Test* but is not ready to go to the public. In such a case, given that the solution to the bug has the highest priority, some "arrangement" has to be improvised with temporal fixes in *Production* waiting for the final solution. Fortunately, such situations are not the norm and are quickly solved.

2.18 System Management

To achieve a smooth and continuous running of SNPator with the lowest possible number of incidences an intense work has to be done at the system management level (servers, network, databases, etc). This implied the creation of control and monitoring systems.

Some scripts have been programmed that are running around the clock monitoring periodically the CPU use, memory, disk space, database free space, etc. These scripts store their measurements into files for later analysis and additionally, if the values cross certain thresholds, they send alarm mails to the members of SNPator technical team.

Other scripts have been prepared to monitor that SNPator daemons *Dispatcher* and *Results Builder* are working. If they are not, those scripts, besides sending mail alarms, restart them automatically. Sizes of PHP, Apache and MySQL logs are also monitored. This is particularly relevant because trivial errors in the execution of the program, from code bugs or system errors, can result in a massive insertion of data into the logs, fill the disk space, whatever their size, and collapse the whole system.

Even when the system is well managed and stable, there is always the need to make operations of maintenance and repair or software upgrades. It is important that these operations are as smooth as possible. As far as the *Computational Servers* is concerned, thanks to the web service structure detailed above, operations go fully unnoticed to the users. The only thing to do is to tell *Dispatcher* to send no more jobs to the affected machine, wait until the last process finish and the machine can be turned off without service affectation.

In the case of Web Servers and Database Servers, the service is transferred to the backup machines while the intervention lasts on the main systems and is returned to them when they are ready. In theory this should cause a cut in service no longer than a few seconds. The problem is that the backup machines, since they are not working actively, can accumulate unnoticed problems that show up only when the service is transferred to them. Periodical transfer of service to the backup machines should be done to ensure that they are ready to take over the service when needed.

In a data oriented environment as SNPator, it is fundamental to backup data constantly. In fact, this is the most relevant of the system administration tasks. In the case of a general failure in the disk systems in which all the information were deleted (all data introduced by users and CeGen platforms), if there were no way of recovering the data from backups, the whole project would be over.

Two kinds of information have to be preserved: the code of the program and the users' data. The code represents a limited amount of information placed in certain directories of the servers. Every day, compressed copies of those directories are stored in two different machines and there are kept in there following a system of rotation: copies from the last seven days, one from a month ago, one from a year ago, etc.

The backup of users' data is a much more complex thing. The amount of data is huge, close to 100 GB, and is stored in a database. In principle, information in a database can be dumped into files and these can be processed in a similar way as done with the code. However, during file generation, the database is stopped or responding very poorly. With the amounts of data involved it would represent 5-6 hours per day of SNPator shutdown. The option of doing it by night has to be discarded because there are users around the world from several time zones.

The solution implemented consists in having two synchronized database managers: the main one accessed by SNPator and a "slave" one that keeps a copy of the data stored in the former which actualizes constantly. When a backup has to be done, the two are desynchronized and the slave is stopped while the main server

continues to offer service. Backup is done from the slave and when it is over they are connected again and the slave actualizes all changes done in the meantime. This way, daily backups are done with no service interruption.

2.19 SNPator Management

Besides the work of strict system management with the infrastructure that keeps the environment running, there is a work of management with the SNPator tool itself. For every project of a CeGen Customer and for every external user, *studies*, usernames and passwords have to be created and everything has to be configured to guarantee access and data confidentiality. When necessary, for example, user groups have to be created to allow joint access to a single *study* and access levels for each user have to be set, following indications of the users themselves.

All this work, system and SNPator management, not only is cumbersome and boring but it is also prone to human error that can have important consequences in such sensitive issues as data confidentiality. For all those reasons, in a parallel way to the SNPator application that gives service to the users, a system administration web has been developed to make easier the tasks and minimize the risk of error.

The System Administration Web (Fig. 39 and Fig. 40) can be accessed only from the university private network, from specific IPs and, naturally, with user and password identification.

On that web there are options to create, modify and delete *studies*, *users* and *groups*, together with the necessary tools to create the interactions between them. The mails that are sent to the users on *study* creation can be also automatically generated

From this web, it is possible to create plots of the system resources as CPU or memory use and monitor what jobs are running in the *Calculating servers*. Reports with statistics of use of SNPator can also be created, including users or *studies* segregated by categories, login reports by months, number of jobs by month,



Fig. 39 Access page to the System Administration Web.

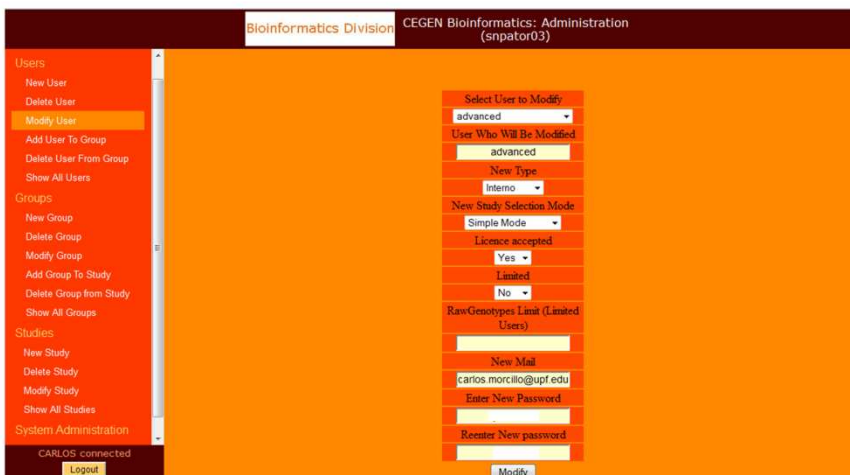


Fig. 40 System Administration Web. Page for user's data modification.

visits to main web page, etc. As a general rule, every SNPator management task that is foreseen to be performed repeatedly is implemented as an option in the administration web to make its execution easier and more secure.

After the experience acquired attending the needs of CeGen and external users, it can be said that without the administration web, SNPator management would not have been possible.

2.20 Implementation Effort

The modular orientation of SNPator allows to going on adding new analysis and data retrieving formats as the needs arise. It is therefore difficult to decide when the project can be considered finished. However, it can be taken as a reference the moment when all informatics infrastructures were set up offering data management to both CeGen and external users and providing a set of analysis options that allowed performing the most usual association studies.

To reach that point 3 years of work of 3 people were needed. Two profiles were involved:

- Project Manager (Carlos Morcillo Suárez). On charge of project leadership and coordination of people involved , requirements analysis, contact with users and platforms, statistical and biological design, software analysis, algorithm and data structure design, systems architecture design, web page design, functional testing of software and users management.
- Programmer. On charge of software analysis, algorithm and data structure design, code programming, creation of web interfaces, systems administration, functional and technical testing of software.

From the above list can be seen that "project manager" and "programmer" labels represent only a part of the functions assigned to each role. Moreover there are several functions shared by both profiles that usually were performed as a team task.

2.21 SNPator Use. Publication

A paper describing SNPator was published in December 2008 in Bioinformatics (Morcillo-Suarez et al. 2008). As to September 2011, SNPator has 653 user accounts, 360 of which are external

(non CeGen) users from international institutions. Since the publication of the SNPator paper, with a delay of around a year, the number of external users that request access to SNPator has steadily grown. The number of logins and jobs launched has also increased over the last few months (Fig. 41).

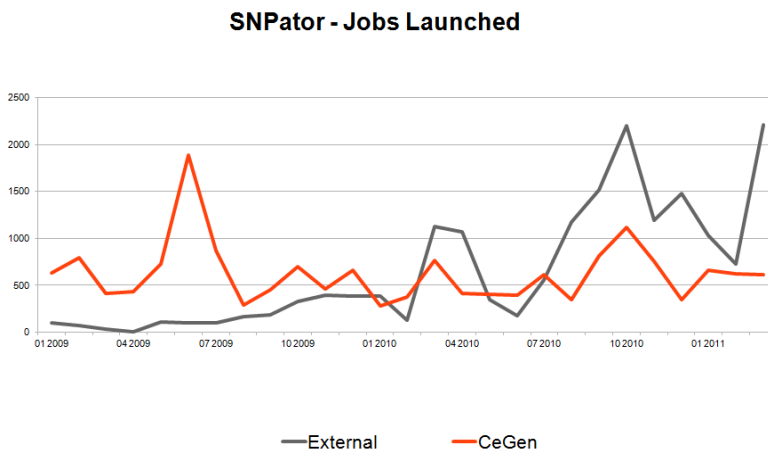


Fig. 41 Evolution of jobs launched by month since January 2009. Number of external jobs increase continuously while those from CeGen institution remain steady.

Also on date September 2011, there are 16 publications ISI citing SNPator. From user trends and the use of SNPator is reasonable to think that the number of publications will increase.

Within CeGen institution, SNPator has been used for quality control, data processing and data transfer to users of 580 projects. SNPator has allowed, at the technical level, to keep the functional unit among the different genotyping centers scattered in different cities.

3. CHAVA

CHAVA (CNV HMM Analysis Visual Application) is an application developed in JAVA that offers a visual environment to help in CNV calling from array-based Comparative Genomic hybridization (aCGH) data. CHAVA has two aspects: the visual environment that helps users to choose optimal parameters for a Hidden Markov Model (HMM) algorithm; and the HMM algorithm itself, also implemented in the program, that is the method used to perform the CNV calling.

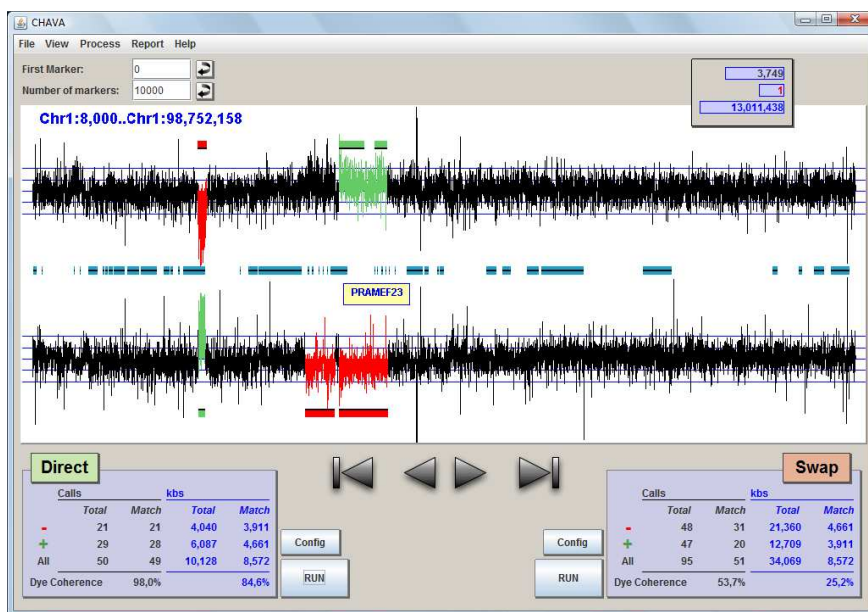


Fig. 42 General view of the main window of the CHAVA application.

3.1 The Problem

3.1.1 CGH

In array-based CGH, one of the common techniques for the study of CNV variation (Iafrate et al. 2004), DNA from two individuals is marked with different dyes, mixed and hybridized to an array of probes the come from known genomic positions. Intensity of each dye is measured for each probe and normalized. Differences in intensity of both dyes, usually expressed as a logarithm of their ratio, indicate a difference in the amount of DNA from each individual hybridizing a certain probe. This hints to the presence of

copy number variation between the two individuals at the position of the probe. Where there are no CNV differences between individuals, similar intensities are expected. Probes used in aCGH can present variable lengths ranging from Bacterial Artificial Chromosomes (BACs), spanning hundreds of Kbs, to oligonucleotides (Knuutila et al. 1998; Pinkel et al. 2005).

Experimental measures in aCGH always present a certain level of intrinsic noise that hinder the individual identification of the results of a single probe as significantly higher or lower than expected under the scenario of equal copy number between the two DNAs being compared. The levels of noise can vary greatly from one experiment to another. They can be particularly important when working with DNA of suboptimal quality or in interspecies hybridization. The level of noise increases the complexity of the CNV calling. To overcome this, all probe values are ordered according to their genomic position and evidence of CNVs is based in the presence of consistent biases in the values of consecutive probes.

To further overcome noise, and as a quality control measure, the experiment is often repeated swapping the dyes used in each individual. Results from both experiments should be consistent. Given that we always measure the intensity of one dye relative to the other one, regions with high values in one experiment should correspond to regions with low values in the other and vice versa.

There are a lot of different algorithms that can be used for CNV calling from CGH data. Their strategies range from simple threshold criteria to complex statistical models (Carter 2007). Approaches using Hidden Markov Models (HMM) are frequently used (Marioni et al. 2006; Day et al. 2007).

3.1.2 HMM

A Hidden Markov Model (HMM) assumes that a string of data has been generated by an imaginary advancing automaton which at each step has a certain state. At each step, the automaton generates data according to a fixed function of the current state. This function is referred in HMM literature as the "emission

probabilities". In the case of aCGH calling, every step would correspond to a probe, the generated data (emissions) would be the intensity ratio for that probe and the possible states would describe the CNV status of that probe (for example, three states could be duplication-same copy number-deletion).

When advancing a step, the automaton can change its current state to a new one following also a function of the current state, called in this case "transition probabilities".

Given a string of data, that are considered as if they had been generated by such an automaton, and given the emission and transition probabilities used by the automaton in generating the data we can use maximum likelihood methods to estimate the string of states that maximize the probability of having produced the observed data. In the particular case of aCGH data, given a string of intensity ratios coming from an experiment and certain emission and transition probabilities, the more likely CNV status can be estimated for the current data.

HMM is a very popular model for Bayesian statistics because it allows for an exact solution of the maximum likelihood combination of states to be found in short computational time using dynamic programming in the form of the Viterbi algorithm (Forney 1973).

The main challenge when calling CNVs using HMM is to find the optimal parameters (emission and transition probabilities) capable to produce an estimate of CNVs consistent with the real CNVs of the samples. The process is more complicated the higher the levels of noise.

3.2. The Application

CHAVA allows the user to manually select HMM parameters and to combine visual and statistical assessment of the quality of the resulting HMM calling in order to facilitate the search for the optimal HMM sets of parameters. The basic modus operandi with the program is to upload aCGH results for a single or a complementary pair of experiments and to run an initial HMM CNV estimation with some starting parameters, either the Program's default parameters

or user's introduced parameters. The results are then visually and statistically assessed by the user. From this assessment the user decides if HMM parameters have to be changed and if so, a new estimation is run and the process is repeated until the quality of CNV calling is considered adequate.

Only if users can navigate through the experiment in an easy and visual way, and only if they get immediate feedback about the effects of the parameter changes they perform, can this strategy be possible. That is why a standalone visual interactive application has been the chosen format, although CHAVA can also work as a command line program that allows simple parallelization of the HMM calling algorithm.

The JAVA platform³⁴ was selected to develop CHAVA because it has the adequate characteristics for this kind of application. Other development environments like PERL³⁵, R³⁶ or Python³⁷ are preferred by the bioinformatics community and have a lot of resources already developed that can be easily reused. They are, however, more oriented for scripting programming than for interactive visual applications.

Netbeans IDE 6.9.1³⁸ with Java Development Kit version 1.6.0.24 was used in the development. CHAVA incorporates the ssj library³⁹ for statistical distributions. CHAVA is released under the GPL license from GNU⁴⁰.

3.2.1 The Basic structure of the HMM

Although transition and emission probabilities are customized by the user, the basic structure of the HMM needs to be implemented during the creation of the program. This original definition has a crucial impact in the posterior behavior of the HMM based CNV

³⁴ <http://www.java.com>

³⁵ <http://www.perl.com>

³⁶ <http://www.r-project.org>

³⁷ <http://www.python.org>

³⁸ <http://netbeans.org>

³⁹ <http://www.iro.umontreal.ca/~simardr/ssj/indexe.html>

⁴⁰ <http://www.gnu.org/licenses/gpl.html>

estimation because it establishes limits to the sensitivity and precision of the calling.

When configuring the states that the HMM structure will contain, it has to be taken into account that their number has to be finite if the Viterbi algorithm is to be used. CHAVA reproduces the configuration of states of HMMSeg (Day et al. 2007). Three possible states are defined: *Deletion*, *Identity* and *Amplification*. This represents a simplification of the possible real CNV status of the experiment which can present a wide range of copy number differences between the two analyzed individuals.

Default values for transition probabilities between states are set in such a way that they foster permanence in the current state and avoid direct transitions between *Deletion* and *Amplification* without staying in the *Identity* state (see Fig. 43). This is a general principle of any HMM based estimation. If transitions of state were not penalized, each step would be assigned the most likely state given its particular value and neither context modulation nor noise correction would take place.

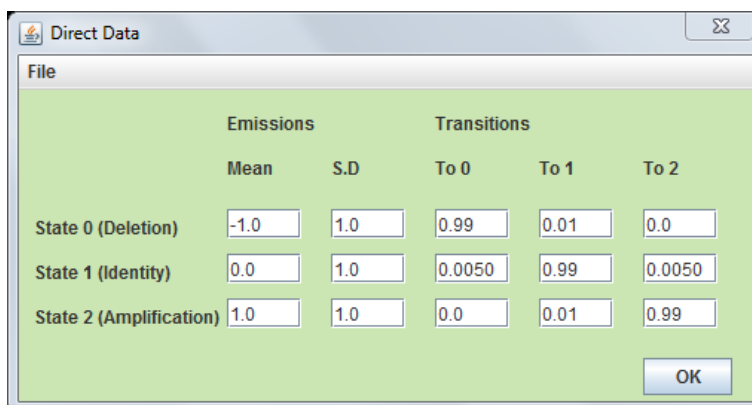


Fig. 43 HMM configuration window showing default HMM parameters.

Typically, HMM emissions consist in a finite set of values each with an associated emission probability. This is the case, for example, of HMM systems for the calling of genes, exons, promoters etc from the DNA sequence (Meyer et al. 2002). There are four

possible emissions: A,C,G,T and each has its own emission probability depending on the state: gen, no gen, exon, etc.

In CHAVA, however, emissions may have any real value. In order to calculate emission probabilities, they are supposed to follow a normal distribution defined by a mean and a standard deviation which are the parameters customized by the user (see Fig. 43).

3.2.2 The Visual Element

The main purpose of CHAVA is to present the aCGH and CNV calling data of the experiment in a way that allow users to naturally draw conclusions about the quality of the calling and can easily act to improve that quality if they deem it necessary.

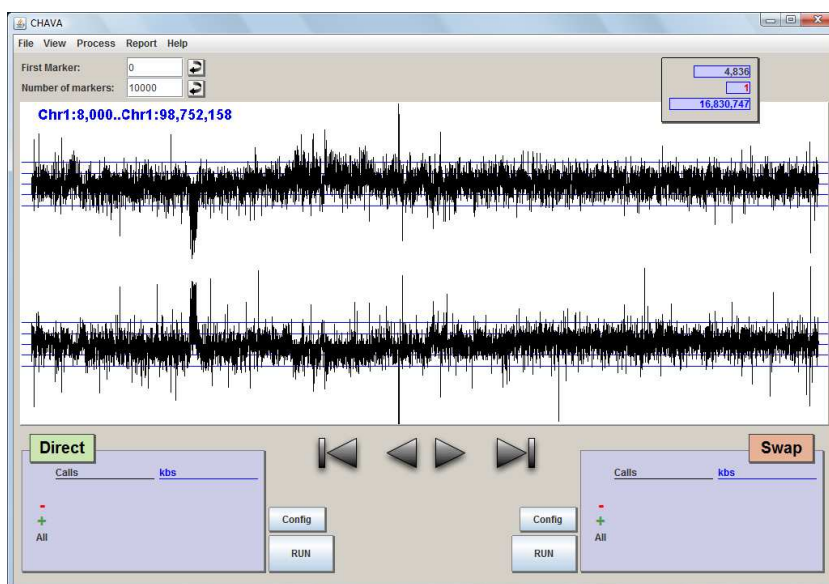


Fig. 44 Intensity ratios of the first 10,000 probes from two complementary experiments are shown as contiguous vertical segments.

For each experiment, intensity values for each of the probes included in the aCGH array are represented as two sets of contiguous vertical segments in the main window of the application (Fig. 44). Various informative and navigation tools can also be found here together with the *Information Panels* (gray bottom left

and right boxes) that will show updated statistics on CNV estimation. We will come back to these tools later. By now, let us focus in the probes themselves

CNV regions are expected to be recognized, in a first approach, by systematic bias of contiguous markers and, when both experiments are present, by consistency between both experiments.

Fig. 45 shows a zoom in to a region of Fig. 44. The highlighted region shows an easily detectable CNV. In such clear-cut instances, most of algorithms, even the most simple, are going to work well. Sometimes, however, the signs of CNV presence are much more subtle and there is need of precisely tailored algorithms.

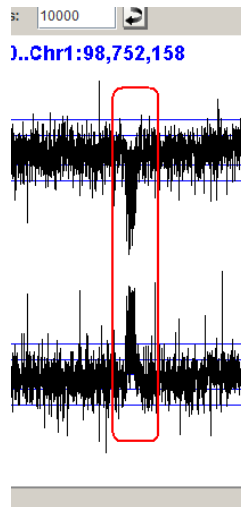
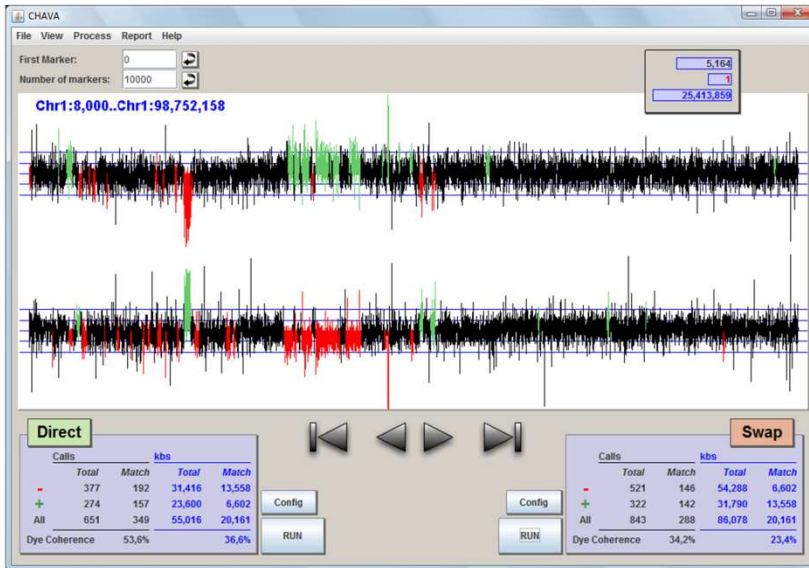
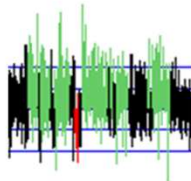


Fig. 45 CNV presence is detected by a consistent bias of contiguous probes. The coincidence of intensities with inverted signs between both experiments confirms the calling and discards an experimental artifact.

a)



b)



c)

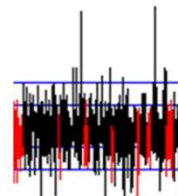
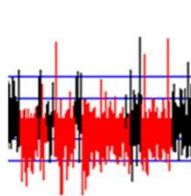
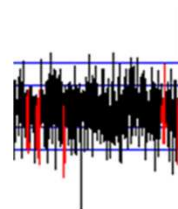


Fig. 46 The low quality of the CNV calling presented in this image can be assessed by the fragmentation of the calls (b) and the lack of consistency between complementary experiments (c).

When HMM estimation is performed, the probes considered to belong to putative CNVs are marked in color, red if they are deletions or green in the case of amplifications. In Fig. 46 a calling

run with default HMM parameters is shown. In this case, as usually happens with default settings, the results are quite unsatisfactory. The low quality of the calling can be quickly assessed by the level of fragmentation of calls (Fig. 46b) and by the lack of consistency between the two experiments (Fig. 46c).

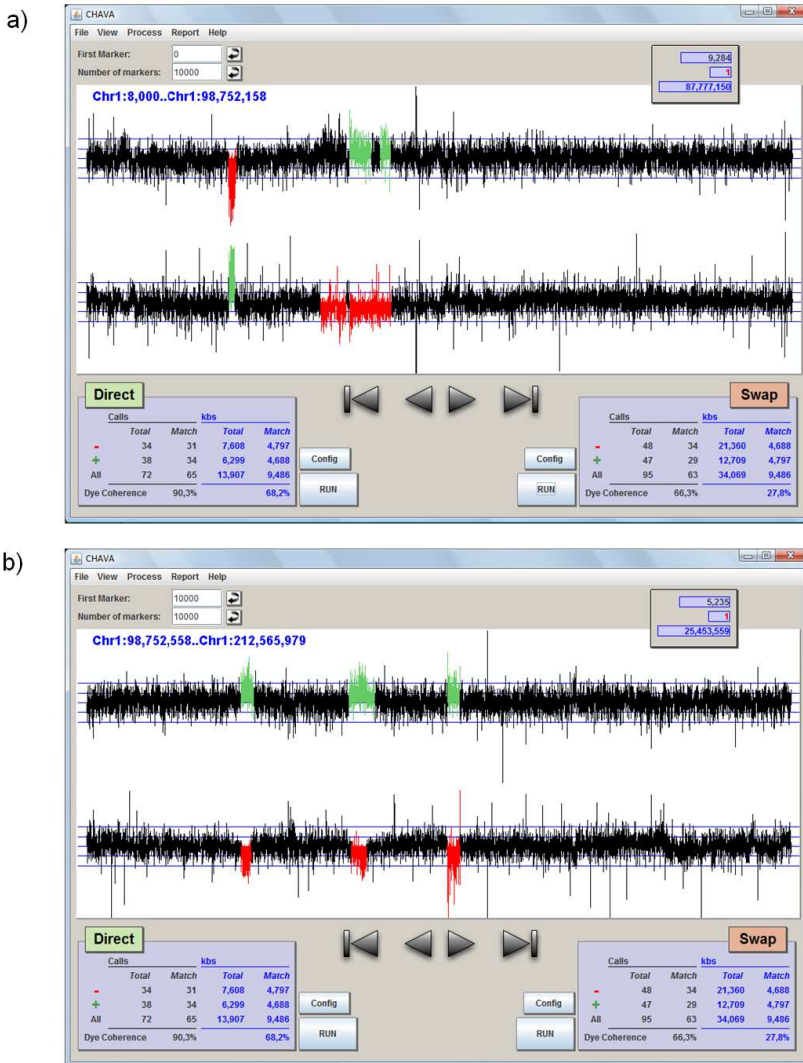


Fig. 47 Examples of CNV calls showing high consistency between both experiments.

When better HMM parameters are used after a process of refining, the improvement in the quality of calling can be perceived at a simple glance as shown in Fig. 47a and Fig. 47b where *Standard Deviation* default value = 1 has been changed to 3 for all three states.

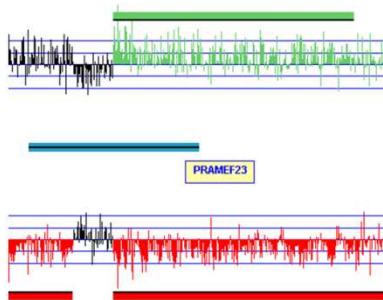
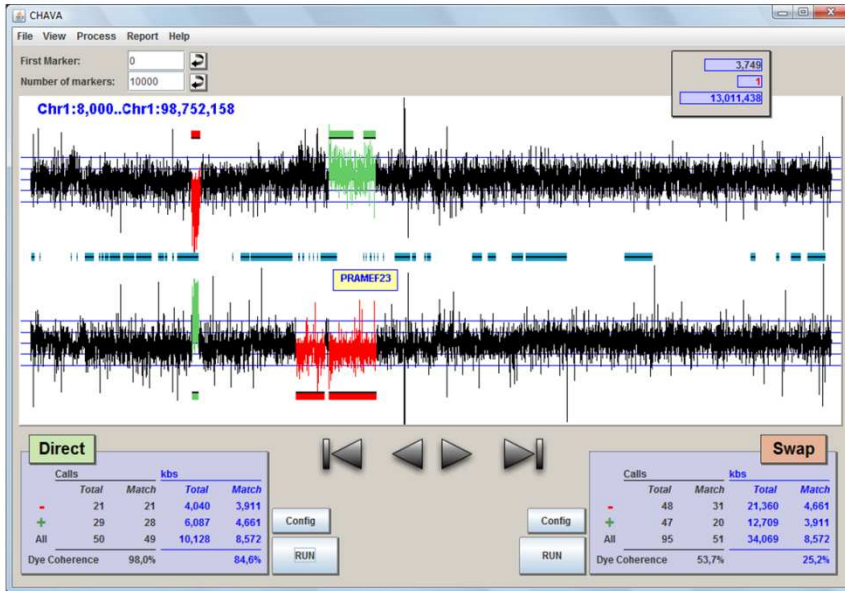


Fig. 48 General view and zoom in where three different tracks can be seen. Upper and lower tracks inform of a previous CNV calling in order to compare with the current one. Central track (blue) shows the genes present in the region.

Further information can be uploaded into the program and visually presented in the form of tracks (until four of them) which can also

pop up information at a mouse click (Fig. 48). Added information can include previous callings, genes or any other genome annotations that can be of help in the interpretation of results and in deciding parameter values.

Frequently, researchers use arrays that present, by design, a structure of clusters of probes in certain regions instead of a uniform distribution across the genome. Since the structure of the array can be important information, users can also have the program displaying genomic gaps between consecutive markers that lie at distances that are larger than a customized value. As a better alternative, a file with a definition of the array structure can be uploaded if it is available, which is the case for most commercial arrays.

All this visual presentation is presented interactively to the user who can navigate through the whole experiment using buttons or the keyboard, zoom in into zones of interest to any degree of resolution, save the current picture to PNG format files, add and remove tracks, etc.

3.2.3 Statistical Information

Together with the visual presentation of CNV callings, a set of statistics are calculated for every HMM estimation that allow assessing the quality of the current calling (Fig. 49). These include the count of calls of each type for every experiment and the DNA span covered by these calls. If both complementary experiments are present, consistency statistics between them are also calculated, including the number of calls and the DNA span that is consistent for each call type, together with a global consistency rate between both experiments. Statistics results together with call list can be saved as text files for future use.

3.2.4 Array Structure

When working with a CGH array which targets specific regions, there may be consecutive probes which appear together in the ordered sequence but map at long distances from each other. Performing a HMM-based calling taking the whole list of probes as a single unit can generate artifacts in the border between those

regions, since HMM takes into account nearby values when assessing a probe and, of course, the copy number status of distant probes will be independent. To overcome this problem, when a definition of an array structure is uploaded into CHAVA, besides plotting that structure, the program can use it to run HMM calling independently for each of the defined probe segments.

	Calls		kbs	
	Total	Match	Total	Match
-	521	146	54,288	6,602
+	322	142	31,790	13,558
All	843	288	86,078	20,161
Dye Coherence		34,2%		23,4%

Fig. 49 For each experiment calling quality statistics are shown. Global matching between experiments is calculated as a percentage of the number of calls and DNA span within calls consistent in both experiments.

3.2.5 Command Line Mode

CHAVA can be run in command line mode. Input files containing CGH data and HMM definitions are passed as parameters and output files are generated containing the resulting calls and the consistency statistics.

This allows for automatic procedures of HMM parameter optimization to be run as a complementary strategy to the visual approach. Systematic combination of different parameter values can be run in batch mode and resulting statistics can be checked for a preliminary approximation to optimal values. Genetic algorithms or any other strategy for finding local optima in complex multidimensional spaces can be applied here.

3.3 Use of CHAVA

CHAVA is described in an article in preparation attached to the annex.

CHAVA has been used for CNV calling of great apes in a recently published work (Gazave et al. 2011). The development of the application has run in parallel to this research and some key design features of CHAVA are result of the challenges it posed.

In an initial phase, DNA from 24 individuals (chimpanzees, gorillas and orangutans) were hybridized against a reference from the same species using a tiling-path 32K human BAC array covering the whole genome. Several regions were considered to harbor putative CNV polymorphisms.

In order to validate the called CNVs and refine their genomic positions, a customized oligonucleotide NimbleGen array was designed to cover specifically the regions detected in the initial phase, plus other regions from the literature. Using this array, DNA from 29 individuals (bonobos were included in this second phase) were hybridized against reference individuals from the same species. Dye swap hybridizations were performed.

Results from the aCGH experiments showed important levels of noise due to the use of human arrays with non human samples, the variation in the reference individuals used due to DNA availability and probably the suboptimal quality of some DNA. The difficulty of CNV calling of the data obtained was increased by the clustered structure of the array used.

CHAVA was used to find out appropriate parameters to perform an HMM based CNV calling of the data coming from the hybridization of NimbleGen arrays. Information on the array structure was uploaded to CHAVA and all CNV estimations were done in a segmented way.

Transition probabilities were fixed at 0.01 chance of changing state and 0.98 of remaining in the same state. Emission probabilities were customized for each experiment. After initial visual inspection of data, CHAVA was repeatedly run in command-line mode for

each experiment under a wide range of emission probabilities. Results of each run were assessed and parameters were selected that maximized the number of long consistent CNVs detected and minimized the detection of short fragmented calls. Once a call was decided, the two complementary dye-swap experiments were visually compared and only those CNV polymorphisms consistent in both were selected.

3.3.1 Simulations by Evolutionary Algorithms

During the process of development of the application, in order to gain insight in the relative relevance of the HMM parameters and help guiding the initial steps in the optimization process, a set of simulations using evolutionary algorithms was performed. We were particularly interested in the relative importance of *Mean* and *Standard Deviation* parameters for all states in the optimization process of the calling of actual aCGH data from primate samples.

All simulations began with the default CHAVA HMM configuration (Fig. 43) as generation 1. At each generation, 10 new configurations were created by randomly mutating the *Mean* and *Standard Deviation* parameters values for the 3 states. Each *Mean* and *Standard Deviation* value was modified independently adding a random value coming from a uniform distribution from -0.3 to 0.3. All configurations of a particular generation, the parent and the 10 offspring, were tested by running CHAVA CNV estimation on the two complementary dye-swap aCGH experiments comparing two individuals. The fraction of DNA length included in the calls of one experiment that was consistent in the other was calculated for both experiments and both values were added up to get a consistency score. This score can present values ranging from 0 to 2, with 0 standing for total lack of consistency and 2 meaning that all markers called in the first experiment are also consistently called in the second and vice versa.

The configuration that obtains the highest score becomes the parent of the next generation and the rest are lost. In case of draw, offspring configurations are preferred to the parent configuration to allow for random drift. Ten new offspring are generated from the

new parent and all the process is repeated until a maximum of 500 generations or a repetition of 100 consecutive generations with the same score value. Parent configurations at each generation are saved for statistical purposes.

A total of 27 simulations were run showing big consistency in the results obtained. A typical simulation is shown in Fig. 50. The values of the consistency score show a rapid increase in the initial steps to converge later to a plateau corresponding to some local maximum from where further improvement is difficult. All simulations began with a score value of 0.6 (corresponding to the score that is obtained using the initial CHAVA HMM default configuration) and converged to final values of 1.41 (SD 0.06).



Fig. 50 Change in the consistency of CNV callings between experiments as HMM configurations evolve during a simulation.

The evolution of *Mean* parameter values showed also considerable consistency among different simulations, as shown in Table 3. A graphical representation for the evolution of these values can be seen in Fig. 51.

State	Initial Value of the <i>Mean</i> Parameter	Final Average Value of the <i>Mean</i> Parameter	SD
Deletion	-1	-1.48	0.22
Identity	0	0.17	0.10
Amplification	1	1.69	0.29

Table 3

Evolution of the "Mean" Parameter

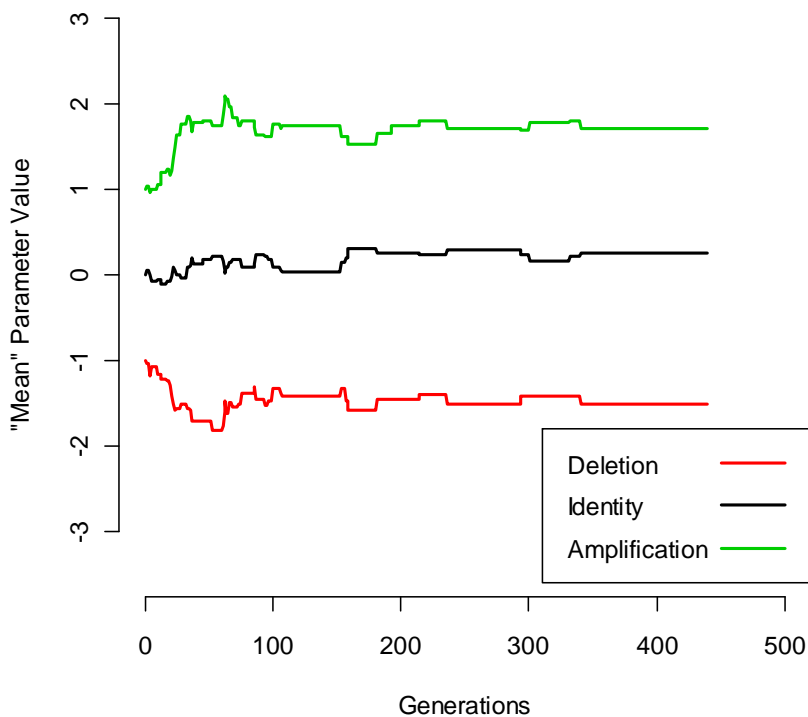


Fig. 51 Change in the *Mean* parameter of HMM configurations for the three states during a simulation.

The evolution of the *Standard Deviation* parameters showed a strong and consistent tendency to increase their values from the starting point before becoming stabilized (Table 4 and Fig. 52).

State	Initial Value of the SD Parameter	Final Average Value of the SD Parameter	SD
Deletion	1	1.83	0.34
Identity	1	2.09	0.31
Amplification	1	2.31	0.48

Table 4

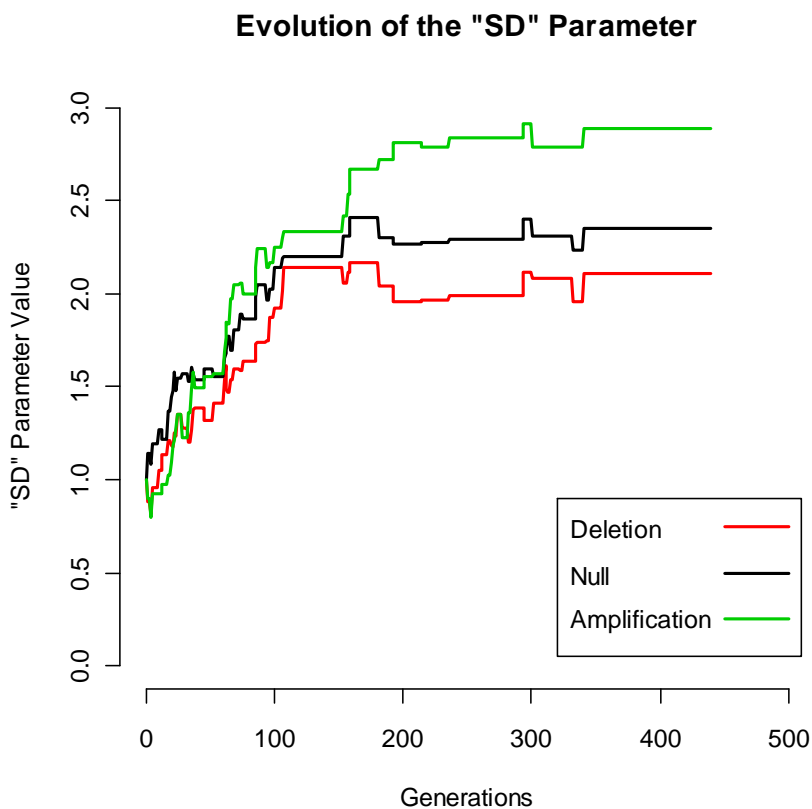


Fig. 52 Change in the *Standard Deviation* parameter of HMM configurations for the three states during a simulation.

Evolutionary simulations showed that, while the values of the *Mean* parameters presented a soft tendency to depart from the original values, *Standard Deviation* parameter values evolved in a more strong and directed way. The fitness function of the simulations (the consistency score value) did not match all the possible visual evaluations that were performed during the primate study using CHAVA. However, the simulations taught us that the *Standard Deviation* HMM parameter, to our surprise, was far more important than the *Mean* and we consequently adapted the posterior strategies of parameter optimization to exploit this fact.

The use of the evolutionary simulations in this work illustrates the enormous possibilities of fruitful synergies that the combination of visual based methods and automated heuristic strategies can bring to the analysis of complex data. The visual and command line modes of CHAVA have been design to facilitate this cooperation.

4. Haplotype Association Pattern Analysis

4.1 Introduction

The term "haplotype" is used in two different senses in the biological literature. First, it may refer to the alleles that, for a given list of polymorphisms, an organism presents in the same physical chromosome. In that sense, haplotypes are inherited through meiosis essentially unaltered unless a recombination event breaks them. Since recombination events are rare (in humans, around 1 event per meiosis for every 100Mb of DNA), haplotypes keep relatively stable through generations and so they have been used in linkage analysis for the determination of recombinants and the fine mapping of Mendelian disorders (Schaid 2004).

A second meaning of the term "haplotype" is related to recombination hotspots and the resulting LD block structure of human genomes (International HapMap Consortium 2005), which translates into the segmentation of the genome in the so called haplotype blocks. Each of these blocks presents, in each human population, a number of allele combinations that is significantly smaller than the number that could be expected by chance. These combinations, consequently, have higher frequency than expected. The term "haplotype" is used, in this second sense, to make reference exclusively to these high frequency allele combinations in the general population.

Association studies, instead of looking for differential frequencies of individual marker alleles between cases and controls, could gain statistical advantages and reduce the number of statistical tests by looking to differences in haplotype frequencies (Clark 2004). Moreover, haplotype association could in theory detect recent rare mutations that are placed inside a given haplotype, while individual marker testing will not have the statistical power to detect such an event, since the level of LD between any given high-frequency marker and a rare variant is necessarily low.

At the moment, haplotypes cannot easily be ascertained experimentally and they usually have to be estimated by computational methods from genotype or sequence data. Different software and algorithms have been developed for the estimation of haplotypes, its visualization and association analysis. PHASE

(Stephens et al. 2001), Haploview (Barret et al. 2005) and PLINK (Purcell et al. 2007) are widely used for these purposes.

In haplotype association tests, the first step is defining the set of markers that constitute the haplotypes that are going to be estimated and tested. In order to minimize the multiple testing problems that may arise if several combinations are tried, a sensible approach is defining the haplotype structure once before the analysis and not modifying it afterwards. Since LD structure lies behind the rationale of haplotype analysis, besides taking into account biological function considerations, haplotypes can be defined to map LD blocks.

Here we follow another approach: to create all possible combinations of contiguous SNPs in the data to analyze. All possible sequential 2-SNP, 3-SNP... N-SNP groups are defined, with N only limited by computational power. For each group, haplotypes are estimated, each haplotype has its frequency compared to the frequency of all the others in cases and controls and the most significant p-value obtained is assigned to the group. All selected p-values are then plotted into a triangular graph similar to that in Haploview (Barret et al. 2005) which is visually and interactively examined. As a proof of principle, two Genome Wide Association Studies from the Wellcome Trust public available data (WTCCC 2007) have been processed according to this approach with the following objectives:

- To ascertain whether there are distinctive patterns in the plot that are indicative of true associations and that allow us distinguishing them from background noise; and, if these patterns exist, to obtain a list of genome regions harboring them
- To compare the candidate regions selected by the process above with the WTCCC results.
- To classify association patterns caused by genotyping artifacts to help in the process of quality control
- To find possible recent rare mutations embedded into a haplotype that cannot be detected by single marker analysis

- To determine the relation between haplotype association strength and LD structure

4.2 Materials and Methods

4.2.1 Data preparation

After requesting data access to Wellcome Trust Case Control Consortium⁴¹ (WTCCC), genotypes and samples data were obtained for the following sample groups:

- 1,500 control samples from the British Birth Cohort
- 1,500 control samples from the UK Blood service
- 2,000 Inflammatory Bowel Disease (IBD) cases (IBD was previously referred as Crohn's disease in the WTCCC original study)
- 2,000 type I diabetes (T1D) cases

coming from the 2007 WTCCC Genome Wide Association Studies (WTCCC 2007).

Data was processed following the indications of the WTCCC paper in order to reproduce the same datasets used in the original study. Genotypes with less than 0.9 score from the CHIAMO calling algorithm (WTCCC 2007) were deleted. SNP and samples showing dubious genotyping quality were removed following the lists provided by WTCCC.

After data cleanup, two datasets were built to perform the analysis:

- IBD dataset, containing the IBD samples and both controls samples.
- T1D dataset, containing the T1D samples and both controls samples.

In the case of T1D dataset, SNPs that in the original WTCCC work were considered as genotyping artifacts after performing the association tests because of their isolated significances were also removed. In the IBD dataset those SNPs are left.

⁴¹ <http://www.wtccc.org.uk/>

4.2.2 Analysis and Visualization

Trend test statistics of individual SNPs, a matrix of pair wise genotypic LD R^2 values with a window size of 50 SNPs and haplotype association analysis were calculated using PLINK (Purcell et al. 2007).

Haplotype association analysis were repeated for the whole genome with the PLINK “--window” option ranging from 2 to 30. All possible windows formed by 2 to 30 contiguous SNPs were considered for all SNPs in the data and haplotypes were estimated for each window using the Expectation-Maximization (EM) algorithm (Excoffier and Slatkin 1995). For each estimated haplotype an association test is performed against the rest.

Since this represented more than a billion association tests which required considerable computational resources, the institution’s computing cluster, an IBM BladeCenter⁴² with 104 processors and 308 RAM GB was used to run the tests in parallel.

The most significant test for each window, together with trend test results and LD information were stored into a MySQL database⁴³. To display this information, a JAVA⁴⁴ visual application was developed. Trend tests, haplotype associations and LD data are displayed in a Haploview-like plot that can be interactively navigated by the user (Fig. 53).

LD R^2 information can also be shown and joint LD-association pictures can be exported for posterior examination (Fig. 54).

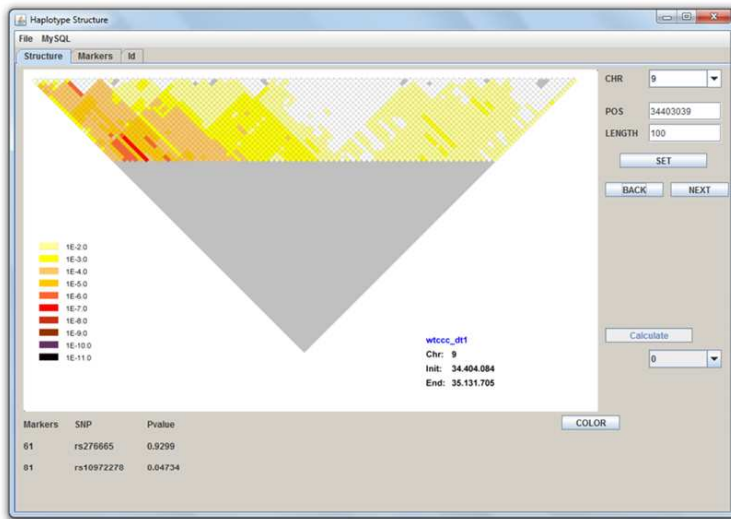
Trend tests and haplotype associations were calculated for the IBD and T1D datasets. To assess the false positive rate of the procedure, two new datasets were created with the original genotypes but with random assignation of affection labels to the samples (IBD_random and T1D_random datasets). After calculating trend tests and haplotype associations for the random datasets, visual inspection of the results with the JAVA application

⁴² <http://www-03.ibm.com/systems/bladecenter/>

⁴³ <http://www.mysql.com/>

⁴⁴ <http://www.java.com>

a)



b)

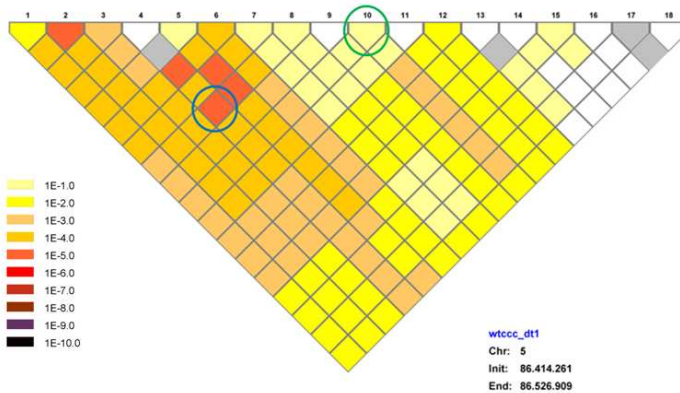


Fig. 53 a) General view of the visualization program showing a 100 SNPs span of chromosome 9. **b)** Closer view showing results for 18 SNPs of chromosome 5 from the T1D dataset. Individual SNP association values (Trend test statistic) are depicted in the top row cells. The pentagon highlighted in green indicates that association p-value for SNP 10 is between 10^{-1} and 10^{-2} . Lower cells show the p-values obtained in the haplotype association tests. The light-red diamond circled in blue means that haplotypes have been estimated for the group of SNPs from SNP 4 to SNP 8, that for each obtained haplotype an association test has been performed against all the rest and that the most significant association obtained presented a p-value ranging from 10^{-5} to 10^{-6} .

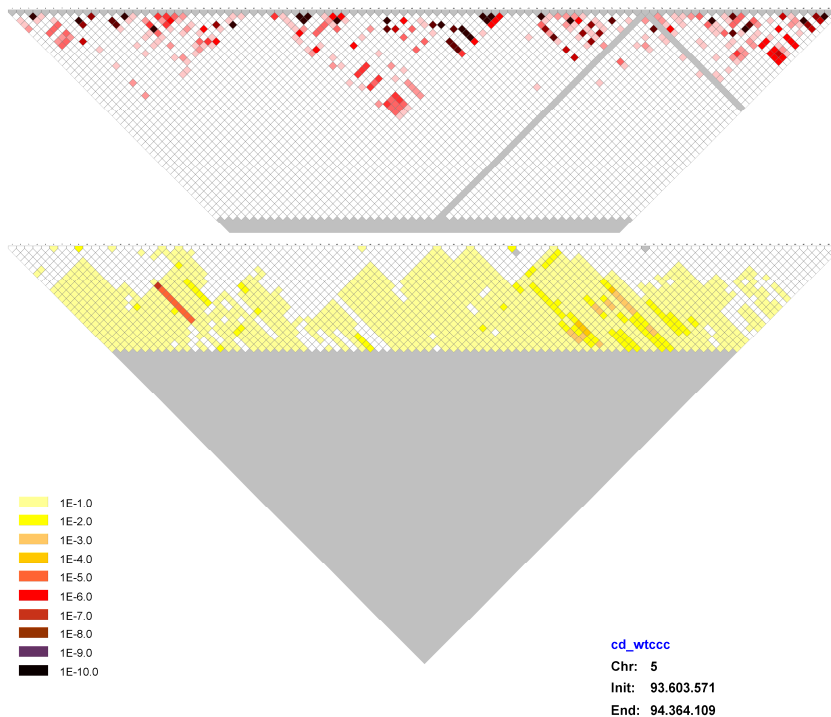


Fig. 54 LD (top) and association tests p-values plots from the IBD dataset shown together for 100 SNPs in chromosome 5.

was used to ascertain what patterns and significance levels could be expected by chance.

Using the previous criterion, IBD and T1D results were visually inspected and windows considered relevant were selected. To assess the possible origin of the observed patterns, 3 sets of simulations were run that consider different possible associations and artifacts. Simulations were based on taking a random, 100 SNP long, region from the IBD data and adding the modifications shown in Table 5. 200 simulations of each type were run.

LD was calculated in the IBD dataset and plotted together with haplotype association results in the selected regions to assess visually the consistency of both.

Simulated Cause of the signal detected	Modification
Genotyped Causative SNP	For each simulation: <ul style="list-style-type: none"> • A SNP is selected at random. • A p-value is selected at random between 0.01 and 10^{-10}. • Case-control labels in data are switched between samples until an association is obtained between the disease and the selected SNP with the significance level previously determined.
Hidden Causative polymorphism linked to a Haplotype	For each simulation: <ul style="list-style-type: none"> • A continuous set of SNPs is selected at random with a length ranging between 3 and 20 SNPs. • Haplotype estimation is performed and a haplotype is selected at random. • A p-value is selected at random between 0.01 and 10^{-10}. • Case-control labels in data are switched between samples until an association is obtained between the disease and the selected haplotype with the significance level previously determined.
Genotyping bias	For each simulation: <ul style="list-style-type: none"> • A SNP is selected at random • A value is selected at random between 0 and 0.05. • A fraction, equivalent to the previous value, of the alleles of case samples are modified to bias them towards one of its alleles while leaving the alleles of control samples untouched.

Table 5 Description of the simulations run to infer the origin of patterns found in haplotype association data.

Patterns detected in the IBD and T1D datasets were classified according to their putative origin (real effects or genotyping artifacts) and were compared with the list of significant regions published in the original WTCCC article.

The regions detected applying our method to the IBD and T1D datasets that were considered to be real effects and that had not been detected in the original WTCCC study were searched for

previous associations in the literature using the OMIM database⁴⁵ and the Phenotype-Genotype Integrator⁴⁶.

4.3 Results

The genome-wide general aspect of the visual representation of haplotype association results is of a continuous landscape of hill-like structures showing moderate significance levels (up to 10^{-4}) for both, the IBD and T1D datasets. Fig. 55 and Fig. 56 give an idea of what most of the genome looks like. Among the roughly 4,000 images generated for each disease, less than 100 depart from this landscape. The regions that in the original WTCCC studies were considered as significantly associated stand out as a span of highly significant hill-like superposed structures (see Fig. 57). Between both extremes there appear sporadic structures of diverse shapes with low significance values, as shown in Fig. 58.

Among the regions that stand out from the general background, most of them present a pyramid-like structure, where all significant cells are confined into a triangular space with vertex in one single SNP. Sometimes the significant cells in a pyramid reach the top cell, the trend test value, as seen in Fig. 59. The SNPs with highly significant associations that were rejected in the WTCCC study because they were isolated positives show up in the haplotype plot as this kind of pyramids. Sometimes, however, the significant cells cover only a part of the pyramid under a SNP (see Fig. 60 and Fig. 61) and could not be detected by an individual SNP test like the ones used in the WTCCC study.

⁴⁵ <http://www.ncbi.nlm.nih.gov/omim>

⁴⁶ <http://www.ncbi.nlm.nih.gov/gap/PheGenI#pgForm>

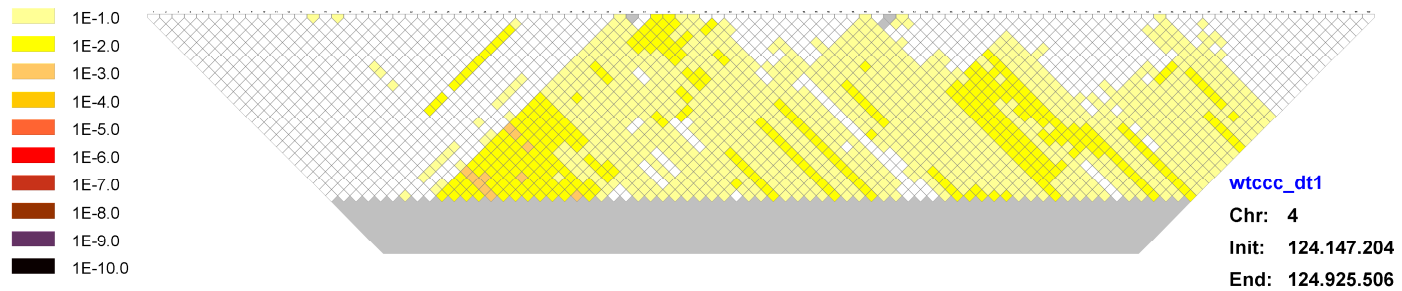


Fig. 55 A view of a randomly selected region of the T1D association results.

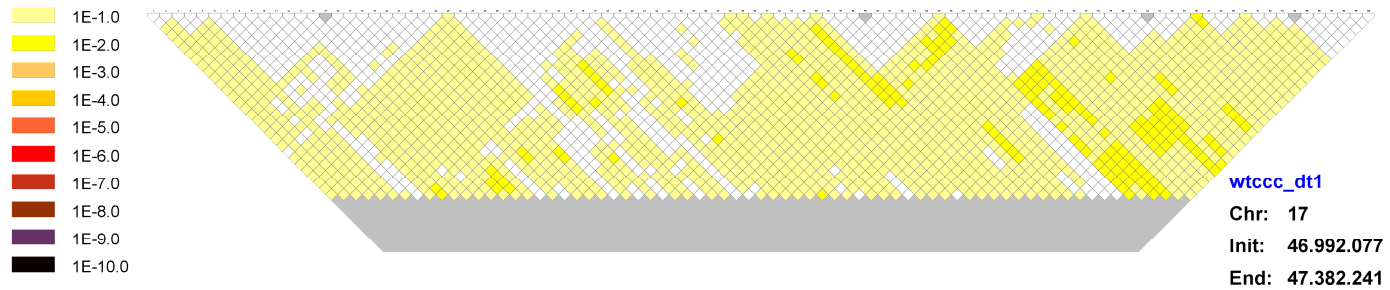


Fig. 56 A view of a randomly selected region of the T1D association results.



Fig. 57 A region associated with T1D in the WTCCC study in chromosome 1 appears in the haplotype association plot as a range of highly significant hill-like structures.

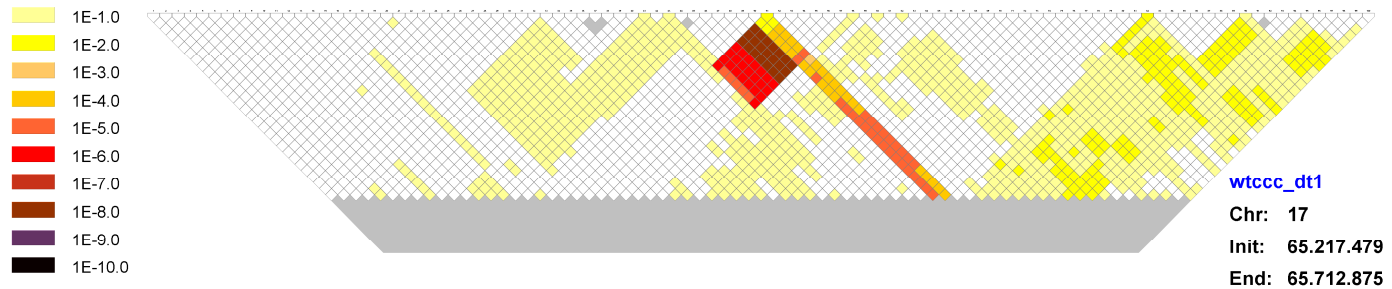


Fig. 58 Low p-values appear sporadically clustered into diverse shapes.

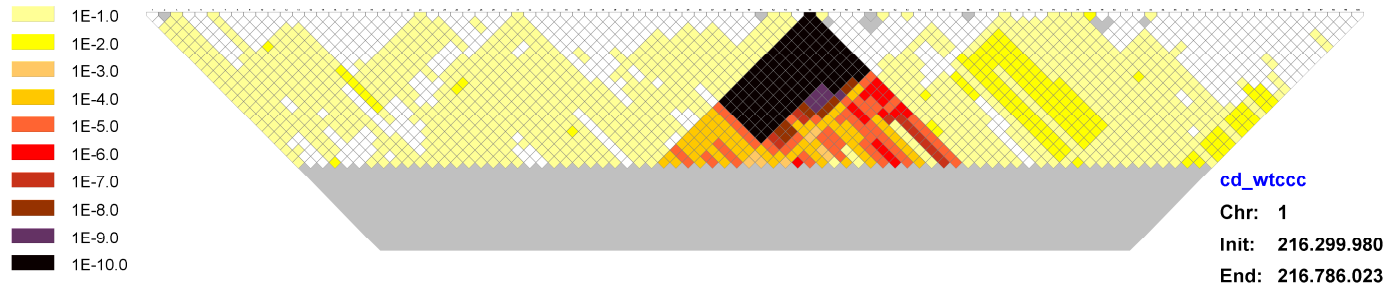


Fig. 59 A typical pyramid-like structure with significant values reaching the top row (single SNP trend test value) from the IBD dataset. WTCCC detected here a highly significant isolated SNP that was considered a putative artifact and rejected.

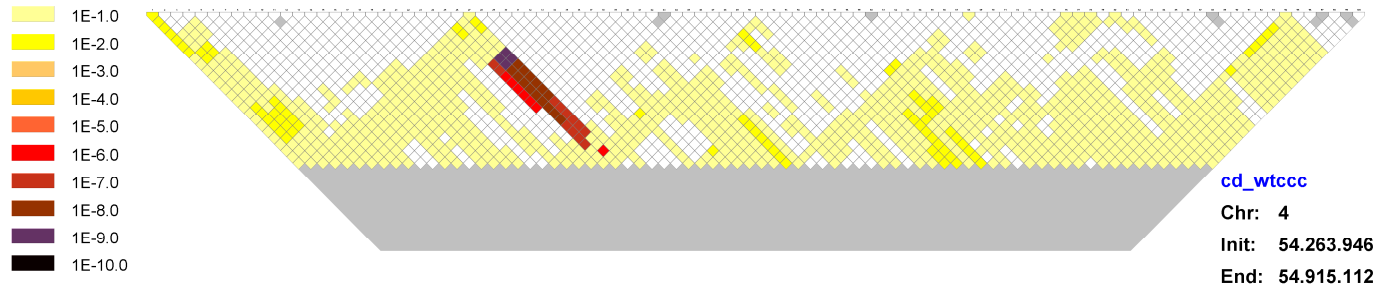


Fig. 60 Significant values forming part of a pyramid-like structure but not reaching the vertex in the IBD study. That implies that there are haplotypes of diverse lengths strongly associated with the phenotype but no single SNP shows significant association. WTCCC could not detect this structure.

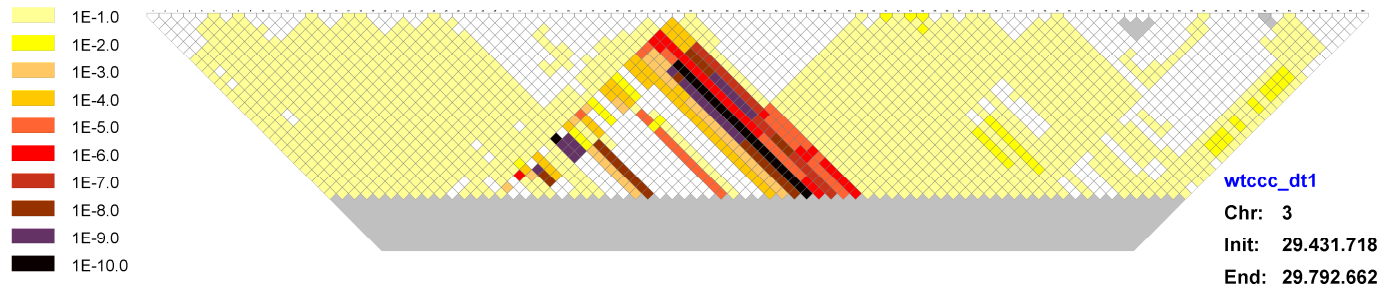


Fig. 61 Another example, in this case from the T1D dataset, of pyramid like structure not reaching the top row that displays the trend test for individual SNPs. WTCCC could not detect this structure.

4.3.1 Comparison between Real and Random Datasets

Random datasets were created and analyzed in identical manner as the originals in order to ascertain what association patterns could be expected by chance. IBD and T1D datasets analysis results differed from random datasets and presented more and stronger association signals. After visual examination, random datasets rarely showed significances beyond 10^{-5} - 10^{-6} while IBD and T1D showed several regions beyond 10^{-10} . Counts of regions presenting p-values under given thresholds are shown for all datasets in Table 6.

Datasets	$<10^{-6}$	$<10^{-7}$...	$<10^{-10}$
IBD_Random	3	0		0
T1D_Random	2	0		0
IBD	74	58		36
T1D	53	28		16

Table 6 Number of regions in each dataset showing p-values under the indicated threshold.

In order to minimize false positive rates, only regions which presented significances lower than 10^{-6} were selected from the IBD and T1D data for further analysis (74 and 53 regions, respectively). The higher number of relevant regions in IBD compared to T1D is consistent with the additional removal of SNPs that were considered artifacts by the WTCCC original study, since, as indicated above, these SNPs were removed during the preparation of the T1D dataset, but not in the case of the IBD dataset.

4.3.2 Simulations

Some of the simulations failed to show any effects since their strength was randomly determined as explained in Table 5. These simulations were not taken into account. The simulations that recreated a real effect, a SNP or haplotype association, generated two patterns that turned out to be quite easy to recognize:

- Pyramid like structures (in 24% of the simulations) where all cells with a significance $<10^{-6}$ are placed in the triangular space under a single SNP (see Fig. 62), and
- complex structures (76% of simulations) with significant cells scattered in irregular shapes not constrained to a single pyramid like-structure. In SNPs based simulations, as expected, significance level always reached the top row of cells while in haplotype simulations sometimes the significant cells remained in low rows (see Fig. 63).

The effect of simulations recreating an experimental genotyping error that affected differently cases and controls was always a pyramid like structure. Just as expected, no effect could be seen outside the pyramid area under the affected SNP. Outside that area, the original significance background values remained (e.g. Fig. 64). This clear-cut geometric constraint contrasts with the SNP and haplotype simulations where, even when they generated pyramid-like structures, there was a broader effect that apparently increased the general significance of the region (e.g. Fig. 65).

Only in 1.6% of the genotyping bias simulations, significances smaller than 10^{-4} were reached outside the pyramid (against 39% in haplotype simulations and 33% in SNP simulations). This 10^{-4} threshold was subsequently used as a criterion to classify pyramid-like structures originated in the real datasets as artifacts or putative true associations.

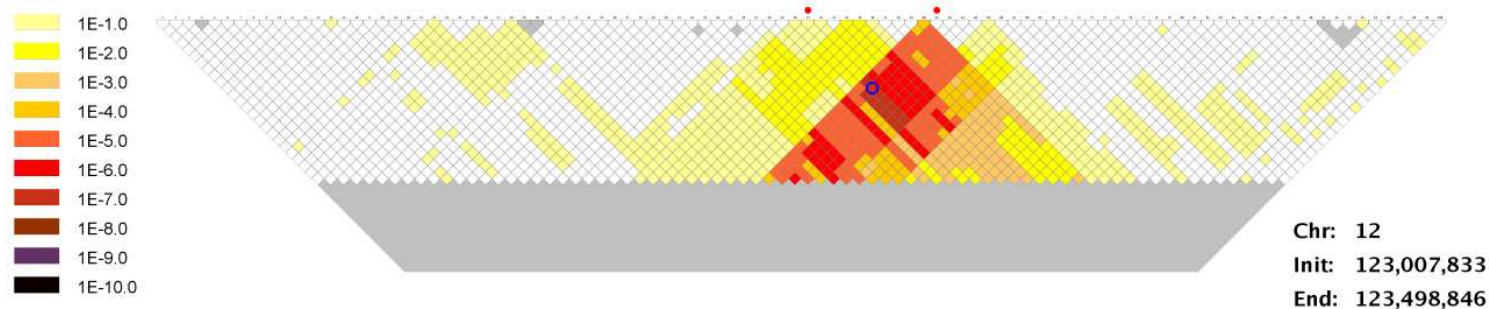


Fig. 62 Simulation of a haplotype association on IBD data. The haplotype length is defined by the little red dots over the top row. Simulated p-value: 5.96×10^{-8} (It shows up in the blue circled cell). A pyramid-like structure appears.

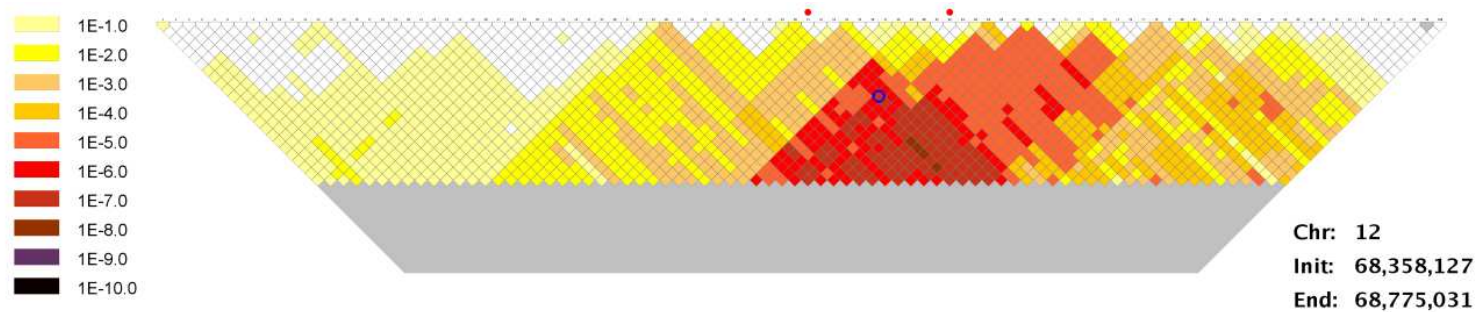


Fig. 63 Simulation of a haplotype association on IBD data. The haplotype length is defined by the little red dots over the top row. Simulated p-value: 7.86×10^{-8} . Hill-like high significance shapes appear but do not reach the top cells and therefore could not be detected by single SNP analysis procedures.

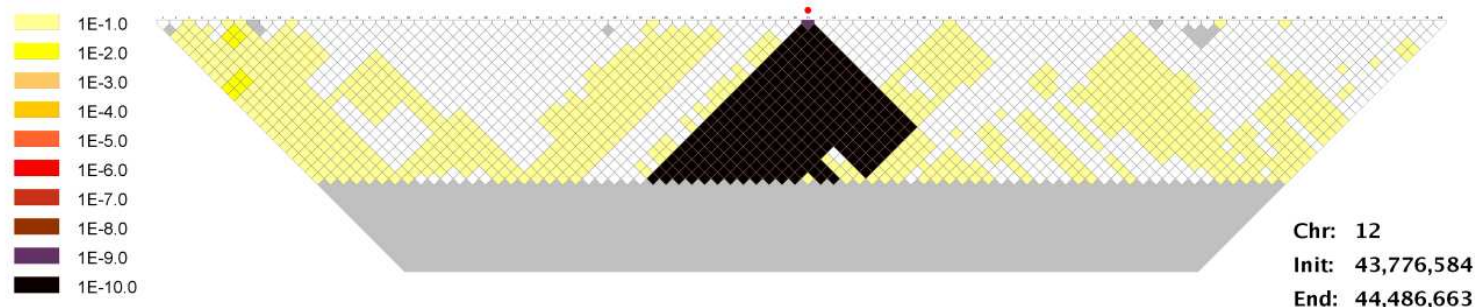


Fig. 64 Simulation on IBD data of a genotype bias affecting 0.02 of case alleles. A pyramid-like structure results. Outside the pyramid the background random significance levels are not affected.

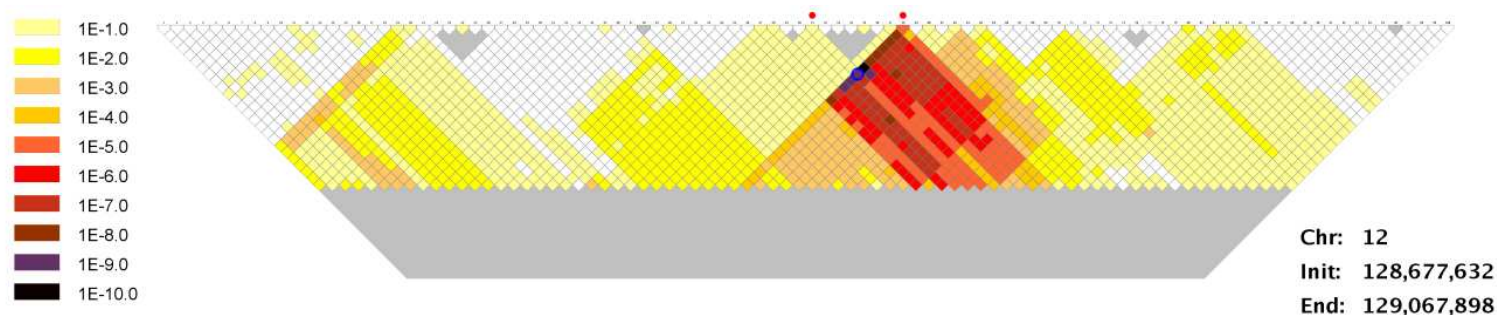


Fig. 65 A Simulation of a haplotype association on IBD data. The haplotype length is defined by the little red dots over the top row. Simulated p-value: 8.6×10^{-10} . The pattern that appears is a pyramid-like structure because all significances lower than 10^{-6} (dark red to black) are constrained inside the triangular zone. However a general increase in significance appears in the region in the form of "shadow hills" that do not appear in genotype bias simulations as, for example, Fig. 64.

4.3.3 Classification of relevant regions

Following the previous criterion, those regions in the IBD and T1D datasets that had been considered not to be random artifacts were classified as:

	Criterion
Putative true associations	<ul style="list-style-type: none"> • Cells with significances $< 10^{-6}$ are scattered forming a complex pattern or • Cells with significances $< 10^{-6}$ are constrained inside a pyramid-like structure, but outside its area cells with significances $< 10^{-4}$ can be observed
Genotyping artifacts	<ul style="list-style-type: none"> • Cells with significances $< 10^{-6}$ are constrained inside a pyramid-like structure and no cells with significances $< 10^{-4}$ can be observed outside the pyramid.

For the IBD data, of the 74 regions considered as non random, 17 were considered candidates to harbor real effects and the rest were considered as probable genotyping artifacts. Most of regions showing genotyping artifacts were not detected in the WTCCC study because they did not show up in the single SNP analysis (Fig. 61). From the 17 selected regions, 11 of them were coincident with the zones detected by the WTCCC study, including the nine that were reported as significant in that study. Six new regions not detected in the WTCCC appeared among the putative true associations (Table 7).

<u>Chromosome</u>	<u>Init Position</u>	<u>End Position</u>
1	50,367,172	52,133,652
4	80,746,038	81,447,315
5	116,949,518	117,469,873
5	158,524,269	159,040,477
14	87,424,731	87,814,495
17	40,215,852	42,405,984

Table 7 Regions from IBD data analysis considered candidates to harbor real associations that were not detected in the WTCCC study. (All genome references are based on the human 36.3 build)

For the T1D data, of the 53 initially selected regions, 13 were considered non artifacts. Nine of them had been detected in the WTCCC study including the five considered statistically significant. Four new regions are detected with the haplotype analysis (Table 8).

<u>Chromosome</u>	<u>Init Position</u>	<u>End Position</u>
4	57,244,340	57,804,878
5	167,653,043	168,170,257
8	4,374,644	4,663,842
9	34,017,923	34,813,809

Table 8 Regions from T1D data analysis considered candidates to harbor real associations that were not detected in the WTCCC study. (All genome references are based on the human 36.3 build)

Several regions that were highlighted in the WTCCC study without reaching significance level were discarded in our haplotype association analysis because did not differ from randomly generated signals (e.g. Fig. 66).

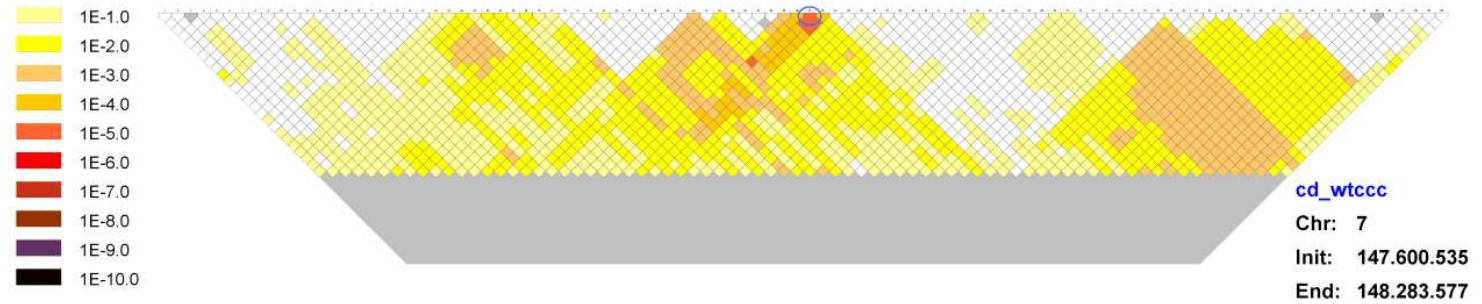


Fig. 66 Region selected by the original WTCCC study as a candidate of harboring an association with IBD but not reaching the corrected 5×10^{-7} significance level used in that study. The SNP with the lowest p-value in the WTCCC study appears here circled in blue. The haplotype pattern representation for this region, however, shows no cells with p-values $< 10^{-6}$ and thus our method considers it to be fully compatible with a randomly generated structure.

4.3.4 Literature search

The selected regions that were not identified in the original WTCCC study (six from the IBD data and four from the T1D data), were subject to a literature search. Two of these new regions in the IBD dataset were found to have been previously associated with Crohn's disease in other studies:

(a) In Fig. 67, we can see the selected region in chromosome 5, around position 158Mb (see Table 7). It was associated with Crohn's disease in a Genome Wide Association study using 3,230 cases and 4,829 controls (Barrett et al. 2008).

(b) Fig. 68 shows the selected region from chromosome 14 (see Table 7). It was found to be associated with Crohn's Disease in a meta-analysis comprising 6,333 cases and 15,056 controls (Franke et al. 2010).

No associations with the studied phenotypes were found in the literature for the other selected regions from IBD study or in any of the selected regions in the T1D study.

4.3.5 LD patterns

For the 17 IBD regions considered to harbor real associations, LD patterns and haplotype association results were plotted together. After visual examination, some rough correlation appears between both plots, but it seems not possible to obtain a reasonable prediction of what would be the better SNPs to analyze in order to maximize the significance of haplotype tests. Some examples are shown in Fig. 69, Fig. 70 and Fig. 71.

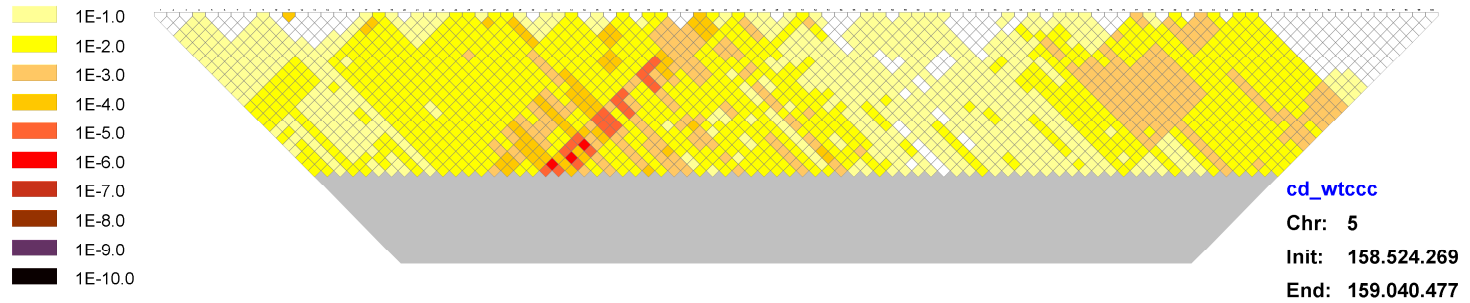


Fig. 67 Region on chromosome 5 of the IBD study previously associated with Crohn's disease.

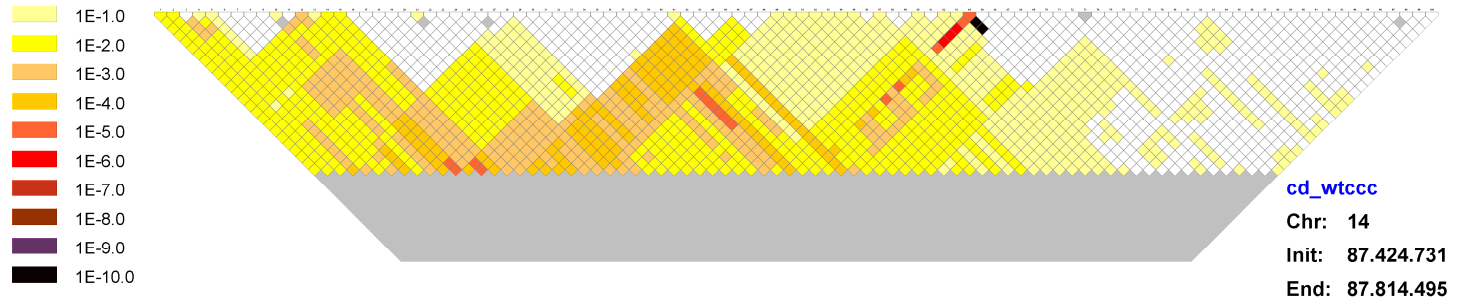


Fig. 68 Region on chromosome 14 of the IBD study previously associated with Crohn's disease.

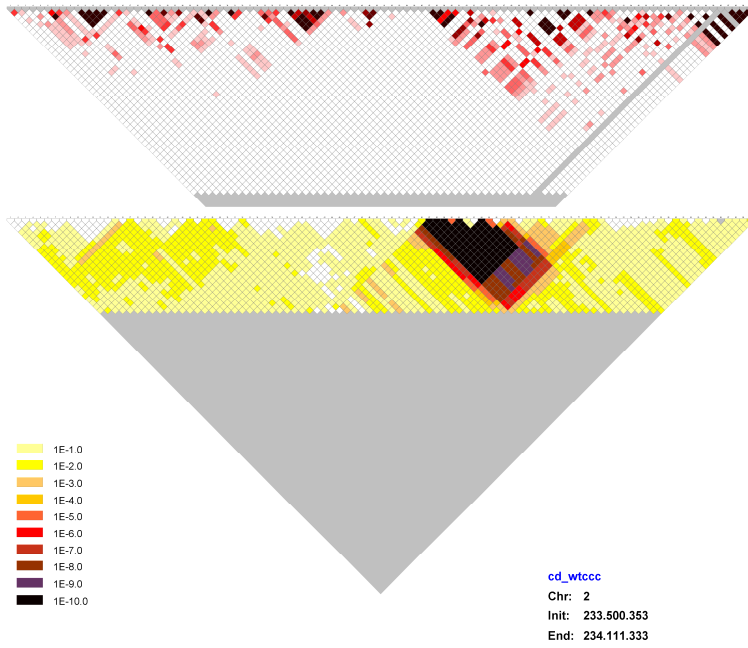


Fig. 69 Joint LD plot (top) and haplotype analysis plot.

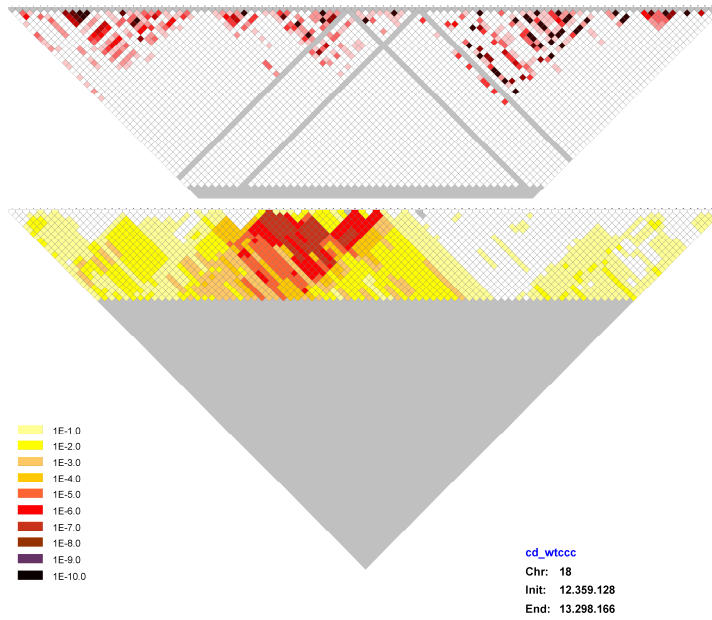


Fig. 70 Joint LD plot (top) and haplotype analysis plot.

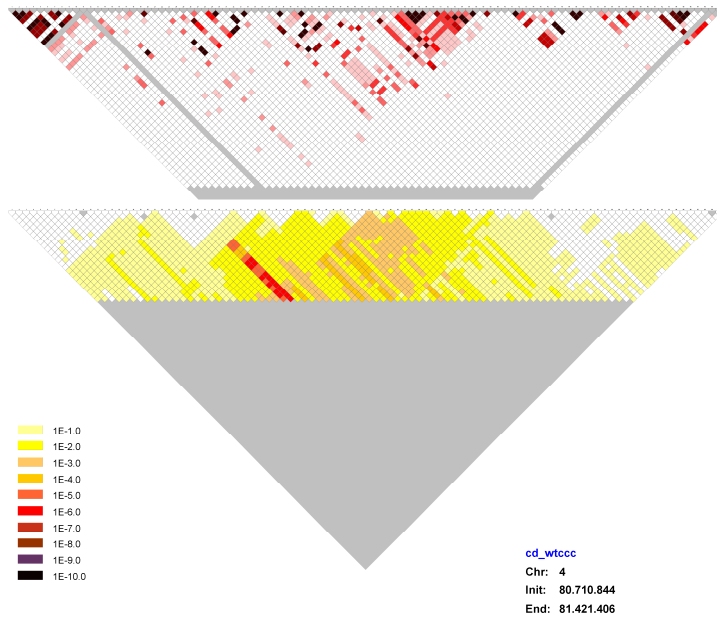


Fig. 71 Joint LD plot (top) and haplotype analysis plot.

4.4 Discussion

The method described in this work pretends to increase the amount of information that can be extracted from case-control SNP genotyping studies by means of presenting an exhaustive haplotype association analysis in a visual way that allows human intuitive pattern recognition to discover hidden effects that cannot be unveiled by a individual SNP analysis.

Random datasets and simulations have been used to train the eye in pattern reconnaissance to be able to pick up relevant signals that differ from noise and to classify them. Rules of thumb have been generated to systematize the decision process. Selection and classification of the relevant regions, has been treated in this work as an initial heuristic method of discovery of candidate regions without assigning significance value. However, once patterns of interest have been discovered through this procedure, they can be codified into analytical algorithms to automate and quantify the relevance of the findings. Further, more formal, developments should follow up.

Haplotype pattern analysis of IBD and T1D datasets has detected all regions that were considered as statistically significant in the original WTCCC study. Some of the regions that were considered as possible associations by WTCCC, but lacked conclusive evidence, have also been detected by our approach, while others turned up to be consistent with the range of signal obtained in the random simulations and therefore have been considered false positives.

Associations that were rejected by the WTCCC study because they consisted in isolated significant SNPs and therefore were likely due to genotyping artifacts were also detected with our method as clear-cut pyramids, thus confirming their artifactual status. We also detected many more putative genotyping artifacts that the individual SNP analysis of WTCCC could not detect because they presented themselves as pyramid-like structures not reaching the top. We calculate that with haplotype pattern analysis, we have detected a number of genotyping errors that is four times larger than those detected in the original WTCCC study.

The most important objective of the method was to discover new putative associations in the WTCCC data that had not been detected in the original study. Ten new regions, six for IBD and four for T1D, have been proposed as association candidates. After searching the literature, two of them, located in chromosome 5 and 14, turned out to have been found significantly associated with Crohn's disease. WTCCC SNP data, as can be seen in Fig. 67 and Fig. 68, when analyzed SNP by SNP, generates single isolated point values around 10^{-4} - 10^{-5} that cannot be considered strong enough evidence for association. In order to detect these two regions new studies with increased statistical power were necessary. This was achieved either by recruiting more cases or by performing meta-analysis. In sharp contrast, the Haplotype pattern analysis strategy followed in this work was able to point at these regions using exclusively the original WTCCC data.

Eight out of ten newly detected regions do not have, as yet, a replication in the literature. This should not be surprising given that most association studies are based in individual marker tests and are, thus, not able to discover any haplotype-based effects that do

not translate into SNP signals. In contrast, our method is well-suited to this kind of effects.

In theory, one important feature of the haplotype test pattern analysis is its power to detect recent mutations with low frequencies linked specifically to a haplotype. However, the lack of empirically phased haplotypes makes this task particularly complex since the imputation error rate can be particularly harmful for low frequency variants. Availability of real phased data should importantly improve, in our opinion, the power of this technique.

The next steps in order to further validate and refine our method should include the processing of independent datasets looking for consistencies in the selected regions, together with a detailed *in silico* study of those regions to explore the presence of putative functional elements that could explain the associations found. Another possibility is performing a functional analysis of nearby genes to look for enrichment of disease-related pathways.

4.4.1 LD ,haplotype patterns and recombination

Examination of the joint LD and haplotype association plots suggest a very complex relationship between both entities. It appears that LD block structure cannot easily be used to predict which SNPs will constitute the haplotypes with the strongest association. If a haplotype association test has to be performed in a certain region, therefore, we believe that it might be better to perform an exhaustive haplotype association test in the form presented in this work and to cope with multiple testing issues later by means of empirical methods like permutations.

Finally, it is interesting to observe that when genomic data containing a strong association, real or simulated, is presented as a haplotype pattern plot, an enigmatic landscape of hill-like structures of diverse sizes and colors appears forming a kind of mountain range where some hills seem to hide behind others. (see Fig. 72).

The particular way how an association manifests itself in a haplotype pattern plot is dependent of the haplotype structure of that region of the genome in the whole population of individuals involved in the association test. This population haplotype structure

is the result of a history of recombination events and, in a lesser degree, segmental duplications and rearrangements. So, in a certain sense, the hills and slopes of the plot are a representation of the genomic history of a population. It is our intuition, which cannot be proved at this stage, that our approach can be used to generate an approximate chronological map of recombination events in a given population.

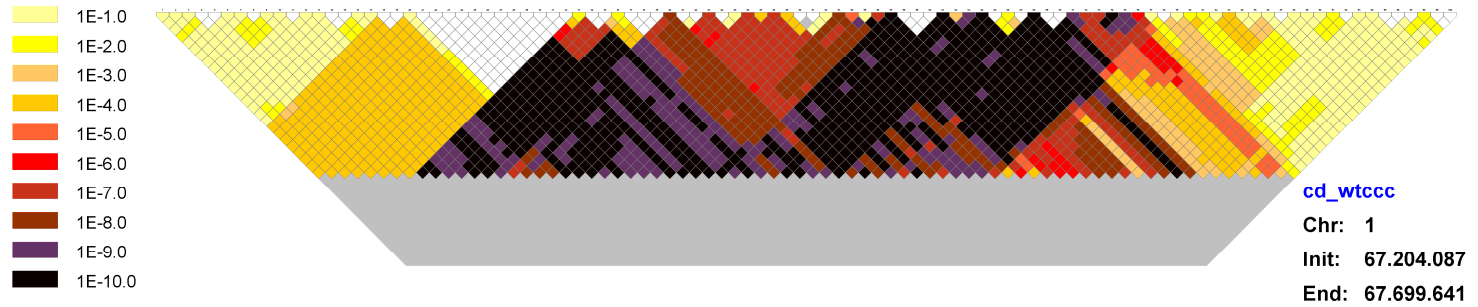


Fig. 72 The typical mountain range landscape that appears in regions with strong association. In this case corresponds to a region of chromosome 1 from the IBD dataset that was detected in the original WTCCC study.

5. Discussion

In this thesis, three applications have been developed and put into practice to illustrate how new technological approaches that simplify the interaction of the researcher with the analyzing process can greatly improve productivity in genetic epidemiology research. Each one of the applications has its own features, making the three of them different not only in goals and scope, but also in what refers to the kind of help they offer to researchers.

The first of these applications, **SNPator**, was born with the intention of creating a unified environment for the management and analysis of genomic data in the context of a multicentre institution. SNPator was designed to replace a set of partial approaches that, in our opinion, were not able to satisfy the needs posed by the appearance of new techniques of high throughput genotyping.

Out of the three projects undertaken in this thesis, SNPator is the one in which a greater and more complex technological effort, including distributed calculations, has been made. Additionally, and in a greater degree than the other two applications, SNPator focuses on enhancing the productivity of researchers and offering an easy-to-use set of tools, thus avoiding complex procedures that have become common amidst the deluge of data that currently floods biomedical sciences.

The central idea behind SNPator, the need for global integrated tools for genomic analysis, which prompted discussions about its convenience in the origins of the project, back in 2003, is today considered obvious.

Simultaneously to SNPator, other projects with the same fundamental philosophy have been developed although they differ from SNPator in both technical approaches and specific goals. One well known example is the package for genomic analysis PLINK⁴⁷ (Purcell et al. 2007), that has become a standard of the field and has been used in many Genome Wide Associations Studies (GWAS) over last few years.

⁴⁷ <http://pngu.mgh.harvard.edu/~purcell/plink/>

Beyond the common bet with those other parallel projects in trying to systematize data formats and in working always to simplify all processes, SNPator has tried to give answer to a set of particular needs different from those of other applications. SNPator has shown to be an effective tool for the process and quality control of data for genotyping platforms and has proved its capacity satisfying for several years those needs in CeGen with more than 580 projects of the institution processed using our application.

Besides CeGen service, SNPator has been published and put at disposal of the public and counts, as to September 2011, with 360 external users with a growing trend in increase of users. Since its publication in late 2008 a growing number of publications cite SNPator (16 as to September 2011) and, to our notice, several more are in preparation.

When analyzing the experience of users using the program, we find that the power of data management of SNPator with its system of filters and *Batch Mode* is one of the most successful parts of the application. This is odd and enlightening. The initial conception of SNPator and its first designs were orientated to build a machine for analysis. Filtering was hardly considered and was included, as also happened with the *Data Caring Module*, when real work began and the first versions of the program encountered real data.

Finally it turned out that these modules are highly useful. The case is quite frequent of projects in which data have been uploaded into SNPator and processed to end up generating input files formatted for other software (PLINK, MERLIN, Haploview...) that includes analysis options different from SNPator. So, instead of using SNPator as an analysis tool, it is used in this case as a quality control and data managing tool that makes much easier the work with other existing software.

Future developments for SNPator may follow several paths. One of them, which is obvious and has never been abandoned, consists in incrementing the range of utilities offered by SNPator. That is, adding new kinds of analysis options and data management tools as their need becomes known, particularly from the comments of the users.

On the other hand, SNPator was conceived at a time, almost eight years ago, when the size of projects was much smaller than what we can encounter today. Future developments should include adapting SNPator to work with current massive studies with samples in the range of tens of thousands and SNPs in the range of millions. This requires fundamental changes in its structure and, following this path, an application called GWASpi (Muniz-Fernandez et al. 2011) has been developed in our group to cover part of this need. The logical next step should be to articulate some kind of integration between GWASpi and SNPator but there are still many design issues about how this should be achieved.

A third major path that the development of SNPator may follow is that of making it a distributable application. From the point of view of the technical deployment that it requires, SNPator is a rather complex software package. Distributability would be necessary if some other researcher of institution wants to install their own version of SNPator to use it in a similar way as CeGen does or just to use it privately because of data confidentiality issues or any other reason. In order for this to be possible, a previous work of simplification and packaging of the application should be done to make its deployment easier.

CHAVA is neither so broad in scope, nor as centered in technological innovation as SNPator is. CHAVA is based on the principle of offering visual representations of complex data together with an interactive tool that allows users using their visual intuition as a constant feedback for improving the solution to a particular problem. CHAVA is described in a paper in preparation, added in the annex, and has been put at disposal of the public for its use together with the necessary documentation and help information.

CHAVA has been developed in parallel with real projects investigating Copy Number Variation that have served as a guide and source of information about the requirements of such an application. This parallel work ensures the practical orientation of the application. One of these projects, already published by our group (Gazave et al. 2011) presented a set of challenges due to the particularities of intraspecific hybridization, changes in reference samples due to DNA availability and suboptimal DNA

quality that increased the levels of noise and complicated the calling process. CHAVA, using a combination of heuristic methods (genetic algorithms followed by systematic search) and visual assessment succeeded in obtaining an acceptable CNV calling.

Use of CHAVA has highlighted some possible paths of future developments for the application. The possible number of states (currently only three: Deletion, Identity and amplification) should be increased in order to reflect in a more accurate way the wide diversity of copies of genome fragments that different individuals may carry. It would also be interesting to integrate into the application some of the heuristic procedures that have been used in the optimization process of the HMM parameter, particularly the genetic algorithms. This could allow for some kind of supervised evolution of parameters where the fitness value of each solution could be visually assessed by a human user.

The fruitful combination of approaches based on human intuition with heuristic algorithms constitutes a fertile field of future developments and is, once again, an unexpected outcome of the project since the original idea was centered exclusively in the visual concept.

Out of the three projects constituting this thesis **Haplotype Association Pattern Analysis** is the one with a greater degree of novelty and a stronger conceptual basis. Just as CHAVA, it is based on the hypothesis that visual representation of data in an adequate format can highlight patterns and relationships that otherwise are difficult to detect. It is also based on the hypothesis that phenotypes can be associated with particular haplotypes without showing any significant association in the individual SNPs that constitute the haplotype.

With these two ideas in mind, we began to process the IBD and T1D datasets hoping to find some regions of high significance in our Haplotype Pattern Plot buried in the bottom of the graph and unnoticeable in the top rows. This pattern would indicate the presence of some haplotype strongly associated with the disease without SNP associations.

Such an image has not been obtained as was originally imagined, although the processing of the WTCCC datasets with our approach has yielded some suggesting results. Some regions not detected in the WTCCC study have been selected that differ from what is expected by chance and that are compatible with the simulations of real SNP and haplotype association effects. It is encouraging that in two of the ten regions selected, independent studies had found associations with the corresponding phenotype (Cronh's Disease).

There are lots of avenues that can be explored in the study of haplotype patterns associated to disease. The work done until now consists in just an initial proof of principle of its possibilities. First of all, it has to be settled whether it is at all possible to detect rare haplotype associations using computer estimated haplotypes. It may well be that, without empirical information on the phase of genotypes our approach loses too much power to be of any use in the study of rare haplotypes. Simulations can be used to answer this question. Additionally, new approaches have to be implemented to take into account not only the haplotype of each window with the most significant association, but to include also all tests from all haplotypes present in that window. Results obtained need a functional follow up and to be replicated in independent datasets.

All three projects, SNPator, CHAVA and Haplotype Association Pattern Analysis, have shown that novel methods can help researchers to cope with the problems posed by the deluge of data under which we are currently trying to do science, which was the main hypothesis of this thesis. The usefulness of visual methods of displaying data has also been proven with CHAVA and Haplotype Analysis.

Finally, as a personal reflection, it is worth mentioning that some of the most interesting results of this research came unexpectedly from strategies and approaches that were far from what the initial planning was.

Bibliography

- Abecasis, G. R., et al. (2002). "Merlin--rapid analysis of dense genetic maps using sparse gene flow trees." Nat Genet **30**(1): 97-101.
- Almeida, M. A., et al. (2011). "An empirical evaluation of imputation accuracy for association statistics reveals increased type-I error rates in genome-wide associations." BMC Genet **12**: 10.
- Altshuler, D., et al. (2008). "Genetic mapping in human disease." Science **322**(5903): 881-8.
- Ashburner, M., et al. (2000). "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium." Nat Genet **25**(1): 25-9.
- Barrett, J. C., et al. (2005). "Haploview: analysis and visualization of LD and haplotype maps." Bioinformatics **21**(2): 263-5.
- Barrett, J. C., et al. (2008). "Genome-wide association defines more than 30 distinct susceptibility loci for Crohn's disease." Nat Genet **40**(8): 955-62.
- Benjamini, Y., et al. (1995). "Controlling the False Discovery Rate: a Practical and Powerful Approach to Multiple Testing." Journal of the Royal Statistical Society B **57**: 289-300.
- Bonetta, L. (2010). "Whole-genome sequencing breaks the cost barrier." Cell **141**(6): 917-9.
- Brookes, A. J. (1999). "The essence of SNPs." Gene **234**(2): 177-86.
- Browning, B. L., et al. (2009). "A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals." Am J Hum Genet **84**(2): 210-23.
- Carter, N. P. (2007). "Methods and strategies for analyzing copy number variation using DNA microarrays." Nat Genet **39**(7 Suppl): S16-21.
- Clark, A. G. (2004). "The role of haplotypes in candidate gene studies." Genet Epidemiol **27**(4): 321-33.
- Collins, F. S., et al. (1997). "Variations on a theme: cataloging human DNA sequence variation." Science **278**(5343): 1580-1.
- Cooper, G. M., et al. (2008). "Systematic assessment of copy number variant detection via genome-wide SNP genotyping." Nat Genet **40**(10): 1199-203.
- Chanock, S. J., et al. (2007). "Replicating genotype-phenotype associations." Nature **447**(7145): 655-60.
- Day, N., et al. (2007). "Unsupervised segmentation of continuous genomic data." Bioinformatics **23**(11): 1424-6.

- Donahue, W. F., et al. (2007). "Fosmid libraries for genomic structural variation detection." Curr Protoc Hum Genet **Chapter 5**: Unit 5 20.
- Donis-Keller, H., et al. (1987). "A genetic linkage map of the human genome." Cell **51**(2): 319-37.
- Donnelly, P. (2008). "Progress and challenges in genome-wide association studies in humans." Nature **456**(7223): 728-31.
- Duggal, P., et al. (2008). "Establishing an adjusted p-value threshold to control the family-wide type 1 error in genome wide association studies." BMC Genomics **9**: 516.
- Eichler, E. E. (2006). "Widening the spectrum of human genetic variation." Nat Genet **38**(1): 9-11.
- Elbers, C. C., et al. (2009). "Using genome-wide pathway analysis to unravel the etiology of complex diseases." Genet Epidemiol **33**(5): 419-31.
- Eleftherohorinou, H., et al. (2009). "Pathway analysis of GWAS provides new insights into genetic susceptibility to 3 inflammatory diseases." PLoS One **4**(11): e8068.
- Excoffier, L., et al. (1995). "Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population." Mol Biol Evol **12**(5): 921-7.
- Fellermann, K., et al. (2006). "A chromosome 8 gene-cluster polymorphism with low human beta-defensin 2 gene copy number predisposes to Crohn disease of the colon." Am J Hum Genet **79**(3): 439-48.
- Forney, G. D. (1973). "The Viterbi Algorithm." Proceedings of the IEEE **61**(3): 268-278.
- Franke, A., et al. (2010). "Genome-wide meta-analysis increases to 71 the number of confirmed Crohn's disease susceptibility loci." Nat Genet **42**(12): 1118-25.
- Frazer, K. A., et al. (2007). "A second generation human haplotype map of over 3.1 million SNPs." Nature **449**(7164): 851-61.
- Gazave, E., et al. (2011). "Copy number variation analysis in the great apes reveals species-specific patterns of structural variation." Genome Res.
- Goecks, J., et al. (2010). "Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences." Genome Biol **11**(8): R86.
- Griffiths, A. J. F., et al. (1993). An Introduction to Genetic Analysis. New York, W. H. Freeman and Company.

- Hindorff, L. A., et al. (2009). "Potential etiologic and functional implications of genome-wide association loci for human diseases and traits." Proc Natl Acad Sci U S A **106**(23): 9362-7.
- Hirschhorn, J. N., et al. (2002). "A comprehensive review of genetic association studies." Genet Med **4**(2): 45-61.
- lafrate, A. J., et al. (2004). "Detection of large-scale variation in the human genome." Nat Genet **36**(9): 949-51.
- International HapMap Consortium (2005). "A haplotype map of the human genome." Nature **437**(7063): 1299-320.
- Jiang, Y. H., et al. (2004). "Epigenetics and human disease." Annu Rev Genomics Hum Genet **5**: 479-510.
- Kidd, J. M., et al. (2008). "Mapping and sequencing of structural variation from eight human genomes." Nature **453**(7191): 56-64.
- Knuutila, S., et al. (1998). "DNA copy number amplifications in human neoplasms: review of comparative genomic hybridization studies." Am J Pathol **152**(5): 1107-23.
- Korn, J. M., et al. (2008). "Integrated genotype calling and association analysis of SNPs, common copy number polymorphisms and rare CNVs." Nat Genet **40**(10): 1253-60.
- Kruglyak, L., et al. (1996). "Parametric and nonparametric linkage analysis: a unified multipoint approach." Am J Hum Genet **58**(6): 1347-63.
- Lander, E. S. (1996). "The new genomics: global views of biology." Science **274**(5287): 536-9.
- Lathrop, G. M., et al. (1984). "Strategies for multilocus linkage analysis in humans." Proc Natl Acad Sci U S A **81**(11): 3443-6.
- Lee, S. H., et al. (2011). "Estimating missing heritability for disease from genome-wide association studies." Am J Hum Genet **88**(3): 294-305.
- Li, Y., et al. (2010). "MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes." Genet Epidemiol **34**(8): 816-34.
- Lupski, J. R., et al. (2010). "Whole-genome sequencing in a patient with Charcot-Marie-Tooth neuropathy." N Engl J Med **362**(13): 1181-91.
- Manolio, T. A., et al. (2009). "Finding the missing heritability of complex diseases." Nature **461**(7265): 747-53.
- Marchini, J., et al. (2007). "A new multipoint method for genome-wide association studies by imputation of genotypes." Nat Genet **39**(7): 906-13.

- Marioni, J. C., et al. (2006). "BioHMM: a heterogeneous hidden Markov model for segmenting array CGH data." Bioinformatics **22**(9): 1144-6.
- Meyer, I. M., et al. (2002). "Comparative ab initio prediction of gene structures using pair HMMs." Bioinformatics **18**(10): 1309-18.
- Michalewicz, Z., et al. (2002). How to solve it : modern heuristics. Berlin ; New York, Springer.
- Moore, G. E. (1965). "Cramming more components onto integrated circuits." Electronics **38**(8): 114-117.
- Moore, J. H., et al. (2006). "A flexible computational framework for detecting, characterizing, and interpreting statistical patterns of epistasis in genetic studies of human disease susceptibility." J Theor Biol **241**(2): 252-61.
- Morcillo-Suarez, C., et al. (2008). "SNP analysis to results (SNPator): a web-based environment oriented to statistical genomics analyses upon SNP data." Bioinformatics **24**(14): 1643-4.
- Muniz-Fernandez, F., et al. (2011). "Genome-wide association studies pipeline (GWASpi): a desktop application for genome-wide SNP analysis and management." Bioinformatics **27**(13): 1871-2.
- Ng, S. B., et al. (2010). "Exome sequencing identifies the cause of a mendelian disorder." Nat Genet **42**(1): 30-5.
- Pembrey, M. E., et al. (2006). "Sex-specific, male-line transgenerational responses in humans." Eur J Hum Genet **14**(2): 159-66.
- Peng, G., et al. (2010). "Gene and pathway-based second-wave analysis of genome-wide association studies." Eur J Hum Genet **18**(1): 111-7.
- Pinkel, D., et al. (2005). "Array comparative genomic hybridization and its applications in cancer." Nat Genet **37** **Suppl**: S11-7.
- Purcell, S., et al. (2007). "PLINK: a tool set for whole-genome association and population-based linkage analyses." Am J Hum Genet **81**(3): 559-75.
- Rakyan, V. K., et al. (2006). "Epigenetic variation and inheritance in mammals." Curr Opin Genet Dev **16**(6): 573-7.
- Reich, D. E., et al. (2001). "On the allelic spectrum of human disease." Trends Genet **17**(9): 502-10.
- Roach, J. C., et al. (2010). "Analysis of genetic inheritance in a family quartet by whole-genome sequencing." Science **328**(5978): 636-9.
- Schaid, D. J. (2004). "Genetic epidemiology and haplotypes." Genet Epidemiol **27**(4): 317-20.

- Sobel, E., et al. (2001). "Multipoint estimation of identity-by-descent probabilities at arbitrary positions among marker loci on general pedigrees." Hum Hered **52**(3): 121-31.
- Stephens, M., et al. (2003). "A comparison of bayesian methods for haplotype reconstruction from population genotype data." Am J Hum Genet **73**(5): 1162-9.
- Stephens, M., et al. (2001). "A new statistical method for haplotype reconstruction from population data." Am J Hum Genet **68**(4): 978-89.
- Tuzun, E., et al. (2005). "Fine-scale structural variation of the human genome." Nat Genet **37**(7): 727-32.
- Weiss, K. M., et al. (2000). "How many diseases does it take to map a gene with SNPs?" Nat Genet **26**(2): 151-7.
- Weiss, L. A., et al. (2008). "Association between microdeletion and microduplication at 16p11.2 and autism." N Engl J Med **358**(7): 667-75.
- Wetterstrand, K. A. "DNA Sequencing Costs: Data from the NHGRI Large-Scale Genome Sequencing Program Available at: www.genome.gov/sequencingcosts. Accessed [2011/08/04]."
- WTCCC (2007). "Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls." Nature **447**(7145): 661-78.
- Yang, J., et al. (2010). "Common SNPs explain a large proportion of the heritability for human height." Nat Genet **42**(7): 565-9.

Annex

Morcillo-Suarez, C., et al. (2008). "[SNP analysis to results \(SNPator\): a web-based environment oriented to statistical genomics analyses upon SNP data.](#)" *Bioinformatics* **24**(14): 1643-4.

Paper in preparation describing CHAVA application.

CHAVA (CNV HMM Analysis Visual Application): a visual approach to HMM based CNV calling from CGH data.

Carlos Morcillo-Suarez^{1,2}, Elodie Gazave¹, Fleur Darré¹ and Arcadi Navarro^{1,2,3,*}

¹Institut de Biologia Evolutiva (UPF-CSIC), PRBB, Doctor Aiguader 8, 08003, Barcelona, Spain. ²National Institute for Bioinformatics, Universitat Pompeu Fabra, Barcelona, Spain. ³Institució Catalana de Recerca i Estudis Avançats (ICREA). Catalonia, Spain.

*Corresponding Author

ABSTRACT

Summary: The discovery of Copy Number Variants (CNVs) and their genotyping are of increasing relevance in the study of genetic variation and disease. Techniques based in Hidden Markov Models (HMM) are frequently of choice when calling CNVs, particularly when array-based Comparative Genomics Hybridizations (aCGH) approaches are used. However, finding optimal emission and transition probabilities can be complex and especially cumbersome when working with suboptimal DNA qualities and/or interspecific hybridizations. CHAVA is a user-friendly visual application that helps users in testing HMM parameter combinations and deciding which are the most suitable to their particular data. The visual presentation of data, together with the statistics computed by CHAVA, allows for easy assessment of CNV calling quality, based on the consistency of results among close markers and between complementary experiments. Additional genomic data can be added as visual tracks to help in the interpretation of the results.

Availability: Free download from CHAVA webpage <http://bioevo.upf.edu/~cmorcillo/tools/CHAVA/CHAVA.htm>

Contact: arcadi.navarro@upf.edu

Supplementary information: Help pages and a detailed tutorial on CHAVA functionality can be found on CHAVA webpage.

1 INTRODUCTION

Copy Number Variations (CNVs) are part of the genetic diversity and are known to affect a range of phenotypes that includes relevant diseases (Fellermann, Stange et al. 2006; Weiss, Shen et al. 2008). Array-based Comparative Genomic Hybridization (aCGH) is frequently used in the study of CNVs as a main source of data (Gazave, Darre et al. 2011) or as a validation tool for CNV callings obtained using SNP array or ultrasequencing techniques. CGH array probes can range from oligonucleotides to long DNA clones as Bacterial Artificial Chromosomes (BAC). aCGH experiments are sometimes performed in pairs of complementary, dye swapped, tests as a method of confirmation of results.

Approaches based on Hidden Markov Models (HMM) are becoming the choice method to call CNVs from CGH data (Marioni, Thorne et al. 2006; Day, Hemmaplardh et al. 2007). The quality of the callings will depend on the degree of optimization of the HMM parameters and can be assessed by levels of consistency among markers mapping close to each other and between replicated experiments. Selection of optimal HMM parameters, a difficult task in itself given the complex biological nature of CNVs, can be especially arduous when working with interspecific hybridization in the field of comparative genomes or, quite simple, when the quality of the DNA used in the experiments is suboptimal. The process of CNV calling is further hindered by the use of general aCGH arrays that, for design reasons, harbor probes that are not evenly spaced within genomic regions and/or that target certain genomic regions that are of interest for a given research project. In general, thus, the relevance of putative CNVs found in any given calling process has to be considered within its genomic context, which can only be fully accomplished by human visual inspection.

All these processes can be complex and time consuming. CHAVA (CNV HMM Analysis Visual Application) graphically integrates a set of tools designed to perform HMM-based CNV calling from aCGH experiments. CHAVA enables the user to combine the use of a series of helpful statistical measures with his or her visual intuition to make optimal decisions in the calling process.

2 IMPLEMENTATION

CHAVA is a visual application developed in JAVA. It uses the `ssj` library (<http://www.iro.umontreal.ca/~simardr/ssj/indexe.html>) for statistical distributions. It can also be executed as a command line application.

CHAVA displays the set of log₂ intensity ratio values coming from a complementary or related pair of aCGH experiments in the form of two sequences of vertical bars corresponding to the intensity values for each marker of the array (See Fig. 1). It allows to navigate along the whole experiment and to zoom into the zones of interest to any level of resolution. After a run of the HMM method implemented in the application, the markers that are considered to belong to CNVs appear colored in red or green, allowing for a quick visual assessment of consistency among neighboring markers and between experiments.

Files with genomic annotations can be added to the working image in the form of tracks. Additionally, the CNVs called by any previous run of the HMM method can be transformed into visual tracks to be used as reference in the process of refining the calling.

The user configures the emission and transition probabilities of the HMM states for both experiments independently. After this has been done, the program estimates CNV presence for each experiment following this configuration. HMM parameters can be saved as files for posterior use. For each run of the HMM method, a series of summary statistics about the number of calls and the length of the DNA sequence included in these calls are computed and displayed. Consistency statistics between both experiments are also displayed to offer an assessment of the quality of the calling and the degree of optimization of HMM parameters that has been achieved by a particular calling

process. The process can be iterated until the user achieves a calling that is satisfactory at both the visual and statistical levels. CHAVA can also work with a single CGH although, naturally, no consistency statistics will be generated.

Because each array used for aCGH has its own design, the program allows uploading a file describing the structure of the array. The description can include the definition of targeted genome regions or sets of probes that map far from each other and cover specific regions. When estimating CNVs, every such segment will be subject independently to HMM estimation to avoid artifacts caused by the interference of markers close in the sequence of data but, actually, far away in the genome. All the images generated by CHAVA can be easily saved as image files. CNV callings and summary statistics can be saved as text files.

CHAVA can also be executed in command line option providing as options the names of files containing intensity ratio values, HMM definitions and Structure definition if needed. Files with the CNV callings for each experiment and with the statistics will be generated.

3 EXAMPLE OF USE

CHAVA has been used for CNV calling of great apes in a recently published work (Gazave, Darre et al. 2011). 24 non human primates were screened for CNVs using a tiling-path 32K human BAC array and a list of putative regions was selected. A customized oligonucleotide NimbleGen array was designed to cover specifically those regions for confirmation. CHAVA was used to find out appropriate parameters to perform an HMM based CNV calling for the custom oligonucleotide experiment. Given the interspecific nature of data, calling as not straightforward and could only be performed thanks to the visual help provided by CHAVA.

REFERENCES

Day, N., A. Hemmaplardh, et al. (2007). "Unsupervised segmentation of continuous genomic data." *Bioinformatics* **23**(11): 1424-6.

Fellermann, K., D. E. Stange, et al. (2006). "A chromosome 8 gene-cluster polymorphism with low human beta-defensin 2 gene copy number predisposes to Crohn disease of the colon." *Am J Hum Genet* **79**(3): 439-48.

Gazave, E., F. Darre, et al. (2011). "Copy number variation analysis in the great apes reveals species-specific patterns of structural variation." *Genome Res.*

Marioni, J. C., N. P. Thorne, et al. (2006). "BioHMM: a heterogeneous hidden Markov model for segmenting array CGH data." *Bioinformatics* **22**(9): 1144-6.

Weiss, L. A., Y. Shen, et al. (2008). "Association between microdeletion and microduplication at 16p11.2 and autism." *N Engl J Med* **358**(7): 667-75.

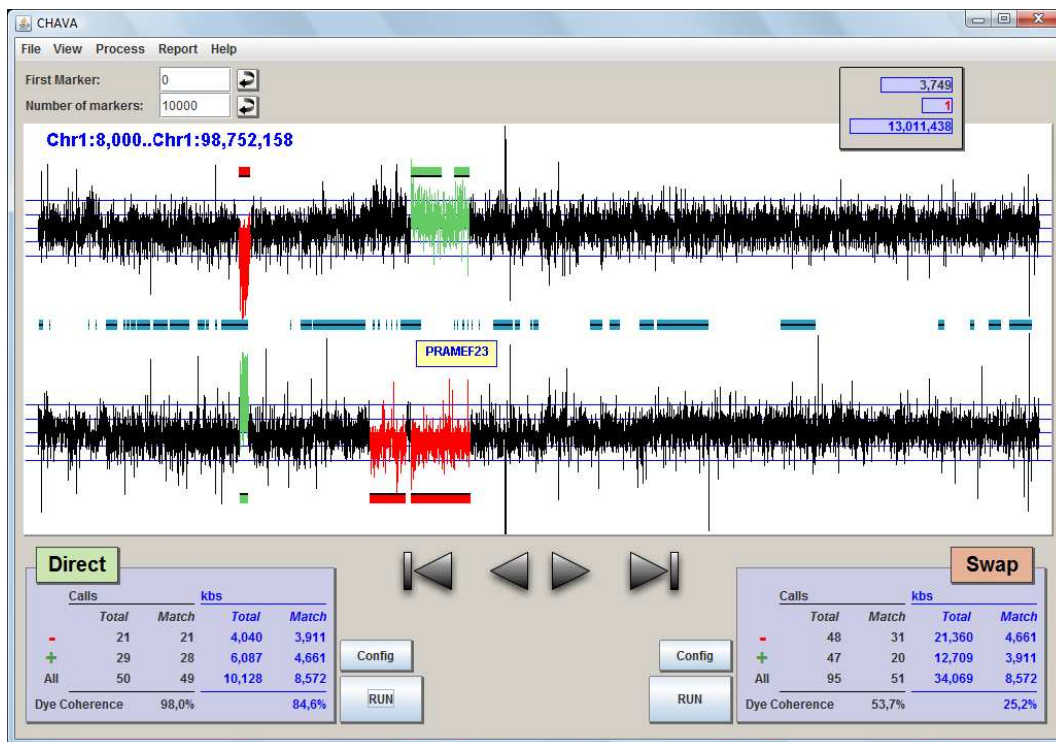


Fig. 1. Intensity differences of two samples for 10,000 first markers of an array based CGH experiment are plotted as contiguous black bars for the direct experiment (up) and swap (down). In red and green appear those regions that HMM estimation considers candidates of harboring copy number variations. Two tracks have been added showing previous CNV callings for the experiments (over the direct bar and under the swap bar) and another track (between the experiments and colored in blue) shows the genes present in the region. Bottom left and right boxes show calling statistics for both experiments and consistency values between them.

Gazave E, Darre F, Morcillo-Suarez C, Petit-Marty N, Carreno A, Marigorta UM, et al. [Copy number variation analysis in the great apes reveals species-specific patterns of structural variation.](#) Genome Res. 2011 Oct;21(10):1626-1639.

Goertsches, R., et al. (2008). "Evidence for association of chromosome 10 open reading frame (C10orf27) gene polymorphisms and multiple sclerosis." *Mult Scler* 14(3): 412-4.

Comabella, M., et al. (2008). "Identification of a novel risk locus for multiple sclerosis at 13q31.3 by a pooled genome-wide scan of 500,000 single nucleotide polymorphisms." PLoS One 3(10): e3490.

Marsillach, J., et al. (2009). "The measurement of the lactonase activity of paraoxonase-1 in the clinical evaluation of patients with chronic liver impairment." Clin Biochem 42(1-2): 91-8.

Bosch E, Laayouni H, Morcillo-Suarez C, Casals F, Moreno-Estrada A, Ferrer-Admetlla A, et al. [Decay of linkage disequilibrium within genes across HGDP-CEPH human samples: most population isolates do not show increased LD.](#) BMC Genomics. 2009 Jul 28;10:338.

Comabella, M., et al. (2009). "Genome-wide scan of 500,000 single-nucleotide polymorphisms among responders and nonresponders to interferon beta therapy in multiple sclerosis." *Arch Neurol* 66(8): 972-8.

Camina-Tato, M., et al. (2009). "Genetic association between polymorphisms in the BTG1 gene and multiple sclerosis." *J Neuroimmunol* 213(1-2): 142-7.

Camina-Tato, M., et al. (2010). "Genetic association of CASP8 polymorphisms with primary progressive multiple sclerosis." *J Neuroimmunol* 222(1-2): 70-5.

Marigorta, U. M., et al. (2011). "[Recent human evolution has shaped geographical differences in susceptibility to disease.](#)" *BMC Genomics* 12: 55.

Muniz-Fernandez, F., et al. (2011). "Genome-wide association studies pipeline (GWASpi): a desktop application for genome-wide SNP analysis and management." *Bioinformatics* 27(13): 1871-2.

Malhotra S, Morcillo-Suarez C, Brassat D, Goertsches R, Lechner-Scott J, Urcelay E, et al. [IL28B polymorphisms are not associated with the response to interferon-beta in multiple sclerosis.](#) J Neuroimmunol. 2011 Oct 28;239(1-2):101-104.