Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática
2007

Pedro Rafael de Sousa Gomes de Almeida

Desenho automático de sondas de microarrays para detecção de mutações

Automatic design of microarray probes for mutations detection

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática
2007

Pedro Rafael de Sousa Gomes de Almeida

Desenho automático de sondas de microarrays para detecção de mutações

Automatic design of microarray probes for mutations detection

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Biomédica, Ramo de Instrumentação, Sinal e Imagem Médica, realizada sob a orientação científica do Dr. José Luís Oliveira, Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dissertation presented to the University of Aveiro for accomplishment of necessary requirements for acquisition of Master degree in Biomedical Engineering, Field of Instrumentation, Signal and Medical Imaging, done under scientific supervision of Dr. José Luis Oliveira, associated Professor of Electronics, Telecommunications and Informatics Department from the University of Aveiro.

Dedico este trabalho à minha esposa, pais e familiares pelo apoio e incentivo ao longo da minha formação académica.

o júri / the jury

presidente / president

Joaquim Arnaldo Carvalho Martins
Full Professor at University of Aveiro

José Luís Guimarães Oliveira
Associate Professor at University of Aveiro

Laura Cristina da Silva Carreto
Assistant Professor at University of Algarve

agradecimentos/ acknowledgements

Agradeço ao meu orientador, Professor José Luís Oliveira, e ao professor Manuel Santos pelo apoio e orientação científica durante a execução da presente dissertação.
Agradeço também ao Biocant a possibilidade de utilizar o trabalho aí desenvolvido para dissertação, assim como o apoio institucional que permitiu testar a aplicabilidade do software e a obtenção dos resultados práticos

I thank to my supervisor, Professor José Luis Oliveira, and to Professor Manuel Santos for the support and scientific orientation during the execution of this dissertation.
I am also thankful to Biocant for the possibility to use the developed work for dissertation as well as the institutional support that have permitted to test the software applicability and to achieve the practice results.

palavras-chave

Bioinformática, microarrays, sondas, SNP, oligonucleótidos, BLAST, mutações, polimorfismos.

resumo

A Utilização de microarrays para detecção de mutações genómicas é uma das vastas aplicações desta tecnologia em biologia e medicina. Mutações são alterações em genes a nível dos nucleótidos que podem ser tão pequenas como uma diferença de apenas um nucleótido entre duas sequências, às quais neste caso se dá o nome de Polimorfismo Nucleotídico Simples (SNP). É possível detectar tais mutações recorrendo à tecnologia de microarrays, apesar de que requer um extenso e moroso procedimento (trabalho) associado ao desenho de sondas, devido a determinadas propriedades comuns que todas as sondas têm de respeitar e ainda o facto de terem de ser únicas.

Depois de avaliado diverso software de desenho de sondas, foi detectada uma evidente falta de ferramentas bioinformáticas para desenho automático de sondas para detecção de mutações. Analisados os procedimentos utilizados no desenho manual de sondas, desenvolveu-se um software de forma a automatizar o processo, reduzindo o tempo dispendido e aumentando a especificidade das sondas finais.

keywords

Bioinformatics, microarrays, probes, SNP, oligonucleotides, BLAST, mutations, polymorphisms.

abstract

Microarrays for genome mutations analysis is one of the wide variety applications of this technology in molecular biology and medicine. Mutations are changes in genes nucleotides, and can be as small as a single base difference between two sequences, known, in this case, as Single Nucleotide Polymorphism (SNP).
With microarrays technology it is possible to detect such mutations. However, it is a large time consuming work associated with the design of probes for such propose, due to the several properties that all probes must have in common and, at same time, the uniqueness of each one.
After evaluated several probes design software, it was evident the missing of bioinformatic tools for automatic probes design for mutation detection.
Analyzing the usual procedure of manual probes design, we developed software in order to automate the process, reducing the overall workflow time and increasing the accuracy of final probes.

Table of contents

Figures

Tables

Formulas

Glossary

Codon	- A set of three adjoined nucleotides (triplet) that codes for an amino acid or a termination signal.
Denaturation	- Dissolution of the DNA double strand into single strands.
Eukaryotic	- Cells that have nuclei and membrane-bound organelles such as animal and plant cells
Exon	- Region of a gene containing DNA that codes for a protein.
Genetic expression	- The effects of a gene's instruction on the cells of an organism.
Genome	All the genetic material in the chromosomes of a particular organism.
Genotyping	- The process of analyzing the particular genetic variations existing in an individual DNA sample.
Hairpin	- Part of single-stranded DNA that hybridizes with itself to form a needle-like structure.
Hybridization	- The annealing of two complementary nucleic acid strands to form a double-stranded molecule.
Intron	- A region of DNA that does not code for the synthesis of a protein.
Melting	- Denaturation of DNA.
Nucleic acids	- Large molecules, generally found in the cell's nucleus and/or cytoplasm that are made up of nucleotide bases.
Nucleotide	- A unit of DNA or RNA, consisting of one chemical base plus a phosphate molecule and a sugar molecule.
Oligonucleotide (oligo)	- Short fragment of a single-stranded DNA that is typically 5 to 50 nucleotides long.
Polynucleotide	- Long chain composed of nucleotides.
Probes	- The oligonucleotides on the surface of microarrays
Renaturation	- Reassociation of complementary single strands of nucleic acids into double-stranded helical forms.
Self-Annealing	- Single-stranded DNA of such sequence that is a complement of itself and therefore it hybridizes with same sequence.

Acronyms

BLAST	Basic Local Alignment Search Tool
bp	Base Pair
CSV	Comma-Separated Values
DNA	Deoxyribonucleic Acid
FTP	File Transfer Protocol
GUI	Graphical User Interface
HGVS	Human Genome Variation Society
IDE	Integrated development environment
MFC	Microsoft Foundation Classes
NCBI	National Center for Biotechnology Information
PCR	Polymerase Chain Reaction
RNA	Ribonucleic Acid
RT	Room Temperature
RTF	Rich Text Form
SDK	Software Development Kit
SNP	Single Nucleotide Polymorphism
Tm	Melting Temperature
XML	Extensible Markup Language

1. Introduction

1.1. Motivation

From the time as far as the experiments of Gergor Mendel studying the segregation of heritable traits in pea plants [1] we try to understand how and what regulates cells in living organisms.

Mendel observed in his experiments that inheritance is a discrete process where specific traits are inherited by the progeny from their parents. These basic units of inheritance are now known as genes [2].

In the cells of organisms, genes exist as linear sections of DNA containing information used to create and control the components of cells.

As we better understand the function of genes, and as more and more genomes are sequenced day by day, we need to analyze them by conducting large scale quantitative experiments with tools directed to the genome sequence itself [3].

Microarrays sophisticated technology [4] is increasingly becoming a valuable tool for genome assays [4, 5] and is being used to identify novel genes, binding sites of transcription factors, changes in DNA copy number, and variations from a baseline sequence, such as in emerging strains of pathogens or complex mutations in disease-causing human genes. They are also used to sort spatially the sequence-tagged products of highly parallel reactions performed in solution.

Microarrays technology is based on bimolecular hybrids specificity between nucleic acids and complementary nucleotides sequences and is today a valuable tool for genetic expression assay at a genome scale [3]. Microarrays have solid supports, usually glass, silicon or nylon where the probes, small sequences of DNA also called oligonucleotides, are placed following an orderly arrangement in microscopic spots. In the presence of complementary molecules, i.e. sequences that are able to base-pair with another, these will be immobilized in the respective spot where the complementary probes are. These microarray chips can have up to 40000 spots each, and for each spot a specific probe must be designed. Although redundancy is usually used for control, there are still thousands of different probes that are designed for one single chip.

However, the probes design is partially still a hand and time consuming work [6] and available software do not fulfill the necessary requirements and the commercial ones are typically very expensive for small or medium sized labs.
The design of oligonucleotide probes can benefit from the application of bioinformatics tools, which can overcome issues like time consuming and the ability to find the best set of probes for specific hybridization conditions [7].

1.2. Objectives

The main objective of this dissertation was to evaluate the bioinformatics needs for the automation and assistance of the designing process of probes for microarrays [3, 8], and more particularly for mutations detection.
The objective of this evaluation was to figure out if there is any tool that can support the design of probes for mutations detection according to a particular set of requirements. The software should take a predefined list of mutations, gene sequences and required probe parameters (such as melting temperature, oligos length, etc) to produce the set of probes most suitable for a given application. Several non-functional requirements should also be considered such as avoiding the possibility of hybridization with any other gene present in the assay, as well as the formation of secondary structures by hairpins or self-annealing.
Depending on the result of this evaluation, the final objective was to develop a novel software tool that would fulfill the specific needs in probes design for mutations detection, not covered by existing software.

1.3. Dissertation structure

This dissertation is divided into more 5 chapters, besides the introduction:

- **Chapter 2** – Scope and user requirements
  In this chapter, we introduce the biological concepts and the principles behind microarrays assays and microarrays probes design. These principles allow to better understand the problem and the requirements we want to tackle.
  Following the context description, we also present the functionalities that the users (mainly biologists) would like to find in the software that designs microarrays probes for mutations detection.

- **Chapter 3** – Evaluation of existing tools for probe design

  This chapter presents the research results about commercial and non-commercial existing software for probes design, and shows the specific missing features.
- **Chapter 4** – Mutation probe design software

  We describe the software architecture implementation, functionalities, structure and user interface, and also the algorithms that were used and implemented.
- **Chapter 5** - Results

  Here are shown the results obtained by using the developed software to design probes for a set of mutations in a specific gene.

  We also compared a set of probes designed by the software against the ones made for a specific real case.
- **Chapter 6** – Conclusions and future work

  In this chapter, we present the global conclusions of this work and some proposals for the future.

2. Scope and user requirements

Genetics is the science that studies the heredity and variations in living organisms [2]. It has its foundation in the classic genetics but gives more emphasis to the individual gene function at a molecular level, being so called as molecular genetics in order to distinguish it from other fields of genetics as the ecologic genetics and population genetics.

With the initiation of genome projects for various species and the consequent complete genome sequencing of several organisms, the genomics has appeared as an evolution of genetics which analyses genetic patterns that can be found in the whole genome of a particular species.

For such large genome scale analysis, the microarrays technology and bioinformatics have become the most important tools.

2.1. Genetic principles involved

Designing probes for microarray implies the knowledge of genetic principles behind microarrays technology, as well as of the hybridization phenomenon existing between complementary single-stranded DNA, which is the starting point to understand this technology.

Following this, we describe in more detail the most relevant biological and genetic principles involved in microarray technology.

2.1.1. DNA sequence

Inside eukaryotic cells there is a nucleus where are stored the chromosomes (Figure 2.1). Chromosomes are DNA (Deoxyribonucleic acid) molecules, condensed in super-coiled structures, where is all the genetic information that controls every cellular functions of an organism [9].

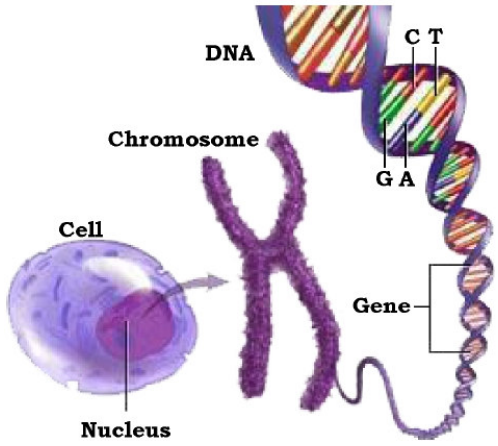


Figure 2.1 – DNA storage location inside eukaryotic cells.

DNA is structured as a double chain molecule twisted around a common axis (Figure 2.2-a). Each chain is composed of nucleotides connected sequentially. Each nucleotide is composed of an azoted base (A-Adenine, C-Cytosine, G-Guanine or T- Thymine) connected to a sugar and a phosphate molecule. One chain connects to the other by the formation of non covalent hydrogen bonds, consisting of a shared hydrogen atom between oxygen and nitrogen atoms from two spatially close azoted bases, always using the same pattern: Adenine pairing with Thymine (A-T) and Cytosine with Guanine (C-G) (Figure 2.2-b).

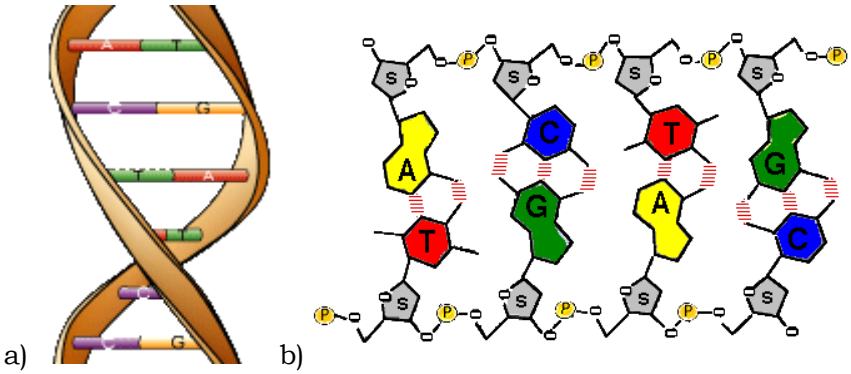


Figure 2.2 – a) DNA double chain twisted. b) DNA chemical structure. A phosphate group (P), a sugar (S) and an azoted base (A,C,G or T) compose a nucleotide.

The genetic information (genetic code) consists of triplets of nucleotides that are called codons. Therefore, as there are 4 possible bases for each of the 3 nucleotides of the codons, there are 64 ($4^3 = 64$) different codons. Each codon, with tree exceptions (stop codons), encodes one of the 20 amino acids (Table 2.1)

used for synthesizing proteins. As there are 64 different codons and only 20 amino acids, there is redundancy in encoding amino acids, and therefore, there are different codons that encode the same amino acid.

Table 2.1 – Conversion table of codons into amino acids: codons where the translation from DNA to amino acid sequence stops (stop codons) are highlighted in red, whereas the codon where translation starts (start codon) is highlighted in green.

First base of codon	Second base of codon: T		C		A		G		Third base of codon
T	TTT	Phenylalanine Phe	TCT	Serine Ser	TAT	Tyrosine Tyr	TGT	Cysteine Cys	T
	TTC		TCC		TAC		TGC		C
	TTA	Leucine Leu	TCA		TAA	STOP codon	TGA	STOP codon	A
	TTG		TCG		TAG		TGG	Tryptophan Trp	G
C	CTT	Leucine Leu	CCT	Proline Pro	CAT	Histidine His	CGT	Arginine Arg	T
	CTC		CCC		CAC		CGC		C
	CTA		CCA		CAA	Glutamine Gln	CGA		A
	CTG		CCG		CAG		CGG		G
A	ATT	Isoleucine Ile	ACT	Threonine Thr	AAT	Asparagine Asn	AGT	Serine Ser	T
	ATC		ACC		AAC		AGC		C
	ATA		ACA		AAA	Lysine Lys	AGA	Arginine Arg	A
	ATG	Methionine Met (start)	ACG		AAG		AGG		G
G	GTT	Valine Val	GCT	Alanine Ala	GAT	Aspartic acid Asp	GGT	Glycine Gly	T
	GTC		GCC		GAC		GGC		C
	GTA		GCA		GAA	Glutamic acid Glu	GGA		A
	GTG		GCG		GAG		GGG		G

On DNA, there are codifying regions, corresponding to genes, and non-coding regions, corresponding to RNA coding regions, binding sites for transcription regulatory factors, as well as some regions with yet unknown function. In its turn, genes possess regions which code for the amino acid sequence that will determine the product protein, called exons, and non-codifying regions called introns. The codifying regions, the exons, are the ones where codons have meaning, that is, where codons are translated into amino acid sequence. The introns are eliminated during a process known as splicing. The start of the exon regions are identified by a specific codon (ATG) and terminated by one of the three codons (the three exceptions mentioned above) called STOP codons that do not code for any amino acid: TAA, TAG, TGA.

It is now obvious the importance of finding mutations in DNA since a single nucleotide change in a DNA sequence can lead to an encoding of an undesired amino acid, which will lead to a wrong protein synthesis.

2.1.2. Hybridization

When DNA fragments are subjected to increasing temperatures, the connections between complementary connected bases are broken. The separation of the two chains is known as melting or denaturation (Figure 2.3-A). The temperature at what this melting occurs is designated as melting temperature (Tm -2.3.2).

When the DNA is denaturated, the two resultant chains are called polynucleotides.

The reverse process, i.e. the pairing of two polynucleotides by the connection of complementary azoted bases from each other, is designated as hybridization, or renaturation (Figure 2.3-B).

Figure 2.3 – a) Melting of DNA structure. b) Hybridization of two polynucleotides.

The hybridization is therefore the formation of a double chain from nucleic acids of different origin. This is the basis of several laboratorial techniques used to study the relation between two DNA samples or to detect specific nucleotide sequences within a DNA sample. These laboratorial techniques generally use probes, small polynucleotides with a known sequence (also called oligonucleotides, or just oligos), that are used to detect, by base pairing, other molecules, the targets, within a heterogeneous mixture of several fragments of nucleic acids.

The DNA sequence is read in the 5'→3' direction (Figure 2.4). When we want to detect a gene, we design probes complementary to the gene sequence called antisense probes. On the other hand, when we pretend to detect the product of such gene, in its mRNA form, then we need a probe with its own gene sequence, called sense probe.

A sequence is called sense if it is the same as that of a messenger RNA (mRNA) copy that is translated into protein [9]. Therefore, assuming that the sequence of the Figure 2.4b is the sense sequence (5'-TCA-3'), the anti-sense sequence will be its complement: 5'-TGA-3'.

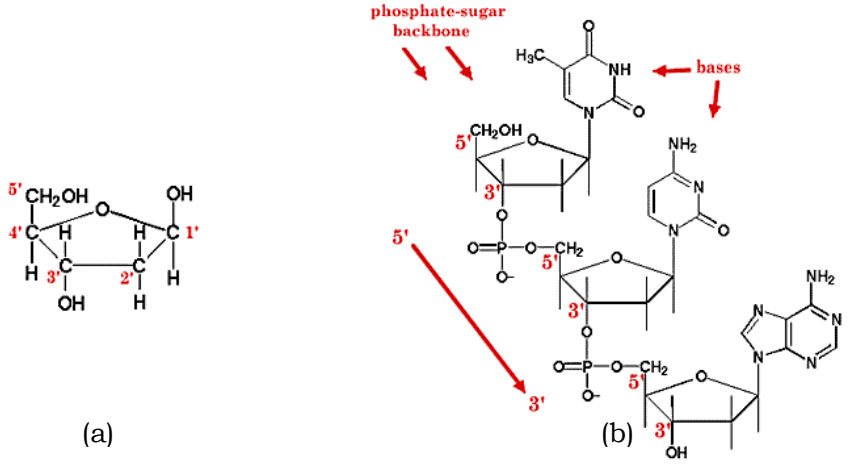


Figure 2.4 - a) DNA sugar (deoxyribose) carbon numbering. b) Single strand of nucleic acid (TCA).

2.1.3. Mutations and polymorphisms

Mutations are variations in a DNA sequence, changing it some times to a rare and abnormal variant [10].

These mutations can, in some cases, lead to a wrong production of amino acids, which in turn will synthesize other proteins than the ones needed.

When a certain DNA sequence variation is common among individuals, more specifically when they occur in more than 1% of a population, it is defined as polymorphism [10]. When polymorphisms are as small as a single base difference between two sequences, they are called as Single Nucleotide Polymorphism (SNP) (Figure 2.5).

The genetic sequence of an organism considered as “normal” is called wild sequence (Figure 2.5-A), while on the other hand, a sequence that has polymorphisms comparing with the wild sequence is called mutated sequence (Figure 2.5-B).

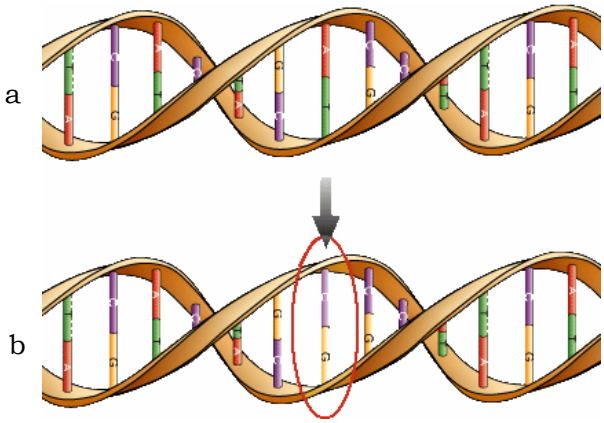


Figure 2.5 – Polymorphism in a single nucleotide (SNP) where an A changed to a C. a) Wild genetic sequence. b) Mutated genetic sequence.

The description of mutations should follow a standard nomenclature in order to allow the clear identification of published mutations and improve compatibility between software that need mutations as input. There is a recommended nomenclature for description of sequence variants [11] presented by the Human Genome Variation Society (HGVS).

Mutations can be of different types. However, six types are sufficient to describe the most common ones that can be found with microarrays. On Table 2.2 there are the six types of mutations used the each respective description according to the HGVS recommended nomenclature [11].

Sequence variations should always be described at the most basic level, such as coding DNA sequence or genomic sequence. In this case we will be describing variations having as reference a genomic sequence, which is recommended to start the description with “g.” like g.76A>T (means a substitution on position 76 of an “A” with a “T”).

Table 2.2 - Different types of mutations and respective human genome variation society (HGVS) recommended nomenclature. α represents a specific nucleotide position within a sequence and β represents a azoted base (A,T,G or C).

Mutation Type	**Description**	**Recommended Nomenclature**	**Examples**
Substitution	Substitution of a base with another in a specific sequence position	$g.\alpha\beta_1>\beta_2$ In position α, the base β_1 was changed to a base β_2	g.34A>T g.2044G>A
Deletion	Deletion of one or more sequential bases in a specific sequence position	g.αdel The base in position α was deleted from sequence $g.\alpha_1\_\alpha_2del$ The bases between positions α_1 and α_2 (inclusive) were deleted from sequence	g.46del g.3455del g.34_35del g.345_346del
Duplication	Duplication of one or more consecutive bases of a sequence	g.αdup The base of position α was duplicated to forward $g.\alpha_1\_\alpha_2dup$ The bases between positions α1 and α2 (inclusive) were duplicated to forward. The same as: a sequence equal to the one between positions α1 and α2 (inclusive) was inserted after position δ of sequence.	g.65dup g.1254dup g.34_36dup g.98_101dup
Insertion	Insertion of one or more bases into a sequence on a specific position	$g.\alpha_1\_\alpha_2ins\beta$ The base β was inserted between positions α_1 and α_2 $g.\alpha_1\_\alpha_2ins\beta_1\beta_2\beta_3$ The sequence $\beta_1\beta_2\beta_3$ was inserted between positions α_1 and α_2	g.46_47insT g.24_25insAT g.89_90insTTTA
Inversion	Inversion of a specific range of bases in a sequence	$g.\alpha_1\_\alpha_2inv$ The bases between positions α_1 and α_2 (inclusive) of the sequence were inverted	g.63_66inv g.333_334inv
Complex	Conjugation of a deletion followed by an insertion	$g.\alpha_1\_\alpha_2delins\beta_1\beta_2\beta_3$ The bases between positions α_1 and α_2 (inclusive) were deleted from sequence and sequence $\beta_1\beta_2\beta_3$ was inserted between positions α_1-1and α_2+1 $g.\alpha delins\beta_1\beta_2\beta_3$ The base at position was deleted from sequence and sequence $\beta_1\beta_2\beta_3$ was inserted between positions α-1and α+1	g.45_47delinsAT g.23_26delinsG g.34delinsATC g.68delinsATTG

2.2. Microarray principles

As mentioned before, microarray technology is a valuable tool for genome assays such as gene expression, genotyping and mutational analysis studies [12].
A microarray is an array of microscopic spots containing usually short, synthetic, DNA segments (called oligonucleotides) ”attached” to a solid substrate such as glass, plastic or silicon chip glass (Figure 2.6) [13] [14].

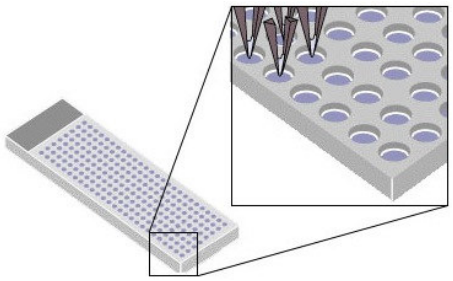


Figure 2.6 – Microarray spots – Microscopic spots are filled with oligonucleotides of a specific sequence. Each spot contains only molecules of a determined sequence.

Microarray technology is based on the specificity of bimolecular hybrids formed between nucleic acids of complementary nucleotides sequence. This allows the detection of a specific sequence within any given complex sample of DNA or RNA.
From a simplistic point of view, to identify the presence of a specific sequence in a DNA sample using microarrays, the following steps are typically taken:

1. The DNA sample is denatured (Figure 2.3) obtaining the single DNA chains (Figure 2.7) ready to hybridize with its respective complement.

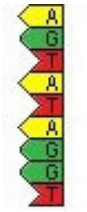


Figure 2.7 – Example of a single DNA chain – The DNA sample to be submitted to the assay.

2. The DNA sample is labeled with a fluorescent marker [15] (Figure 2.8) so they can be identified after hybridized with a probe.

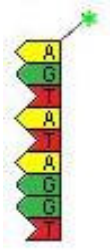


Figure 2.8 – Example of a single DNA chain labeled with green fluorescent marker.

3. On microarray spots there are immobilized oligonucleotides, the probes, with sequence equal to the complement of DNA sample that we want to screen for (Figure 2.9).

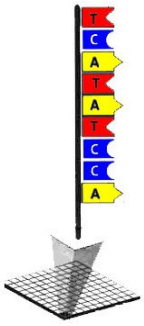


Figure 2.9 – Probe immobilized in a microarray spot.

4. The DNA sample is placed in the microarray in contact with probes. If the DNA sample has the complement sequence of a specific probe, it will hybridize, being the DNA sample "locked" with the probe (Figure 2.10).

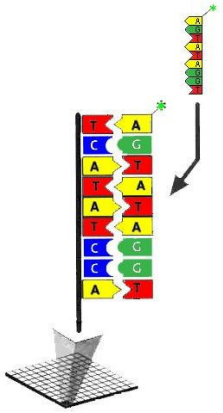


Figure 2.10 – Hybridization between the DNA sample and the probe containing the complement of sample sequence.

5. After the microarray has been washed, only hybridized chains will be present in each respective spot. Because the DNA sample has been labeled with a fluorescent marker it is now possible, with a laser scanner, to identify in which probes have occurred hybridizations. Figure 2.11 shows part of an image of a scanned microarray. The green dots are the spots where hybridization has occurred with DNA sample which was labeled with a green fluorescent marker.

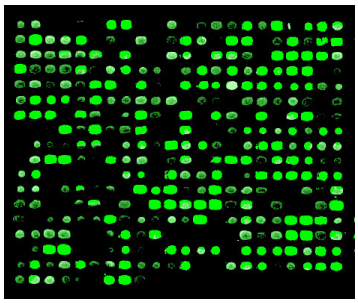


Figure 2.11 – Microarray scanned image.

2.3. Microarray probe design

The oligonucleotide probes that are placed in the microarray spots are designed with the propose to identify a specific sequence within the DNA sample. Although the concept seems intuitive, probe design has to consider several parameters in order to obtain optimal hybridization conditions.

The designed probes have to respect specific parameters (described individually below) like oligonucleotide length (2.3.1), melting temperature (2.3.2), hairpin-loops (2.3.4), self-annealing formation (2.3.5), Nucleotide Run/Repeats (2.3.6) and possibility of multiple matching sequences in the analyzed samples, evaluated by BLAST algorithm (2.3.7).

Particularly in cases like the mutations analysis, control probes have also to be designed to identify the wild type DNA as well as the mutated DNA.

2.3.1. Oligonucleotide length

The oligonucleotide length is the number of nucleotides of a probe. The number of nucleotides for each probe must be inside a specific range since the length of

probes are related to some effects on microarray assay [16] such as the match/mismatch ratio or the intensity of the fluorescent signals.

Beside being less expensive, short probes allow to be more specific for the detection of mutations such as SNP's, because in this case a mismatch of a nucleotide can be sufficient for the non-hybridization of the sample with the probe [17]. However, if a probe is too short, it is highly probable that its sequence appears in other locations of the DNA sample than the desired local target for which it was designed. An inconvenience of short probes is that, as shorter they are, less signal intensity they will produce when the microarray is scanned [18].

Longer probes have the convenience o n the specificity of generating higher fluorescent hybridization signals to be detected by the scanner, but they are more expensive and most of all they reduce the SNP effect of annealing i.e. the mismatch of one nucleotide can be insufficient for the non-hybridization and the probe will hybridize with the sequence even if the SNP for which it was designed to detect is present.

Therefore, the length of the probes must be chosen according to the particular conditions and objectives of the assay.

According to several publications [18-21], optimum probe length for oligonucleotide microarrays is a subject that didn't reach yet a consensus. The most common lengths used for microarrays probes are between 10 and 100 nucleotides. In the specific case of mutations/SNP's detection, the shortest probes are have usually between 10 and 30 nucleotides.

2.3.2. Melting temperature

As mentioned before, the melting temperature (Tm) is the temperature at which the dissociation of a double-stranded DNA molecule occurs. Therefore, the hybridization takes place when the temperature goes below the Tm.

When designing the probes, it is very important to guarantee that all probes in the microarray have a similar Tm, so that all of them hybridize at the same temperature.

There are different methods for the calculation of the melting temperature [22]. The most used ones for probes design are the basic Tm calculation [23] using the salt adjustment formula [22, 24]:

- The Basic Tm Calculation [23] is valid for oligos with 22 nucleotides or less [25] and it is calculated considering an increase in Tm of 2ºC for each A or T nucleotide, and of 4ºC for each G or C nucleotide:

$$Tm = 2 \times (A + T) + 4 \times (G + C)$$

Formula 2.1 – Basic Tm calculation – A, T, G and C are the number of each respective base in the oligo.

- For oligos longer than 22 nucleotides, the Tm is calculated using the Salt Adjusted formula [22, 24]:

$$\mathrm{Tm} = 100.5 + 41.0 \times \left(\frac{(\mathrm{G} + \mathrm{C}) - 16.4}{\mathrm{N}} \right) - \left(\frac{820.0}{\mathrm{N}} \right) + 16.6 \times \log_{10}([\mathrm{Na}^+])$$

Formula 2.2 – Salt adjusted Tm calculation – G and C are the number of each respective bases in the oligo, N is the number of nucleotides and $[Na^+]$ the monovalent salt molarity (usually $[Na^+]$=0,05M).

2.3.3. Stability of a DNA duplex structure

The stability of a DNA duplex structure is important in probes design to predict the formation of secondary structures as hairpins and selfannealing.

The method to calculate the stability of DNA duplex structure involves the use of Nearest Neighbor thermodynamics interaction [26]. The thermodynamic values of enthalpy (ΔHº), the energetic contribution, and entropy (ΔSº), representing the relative stability of the hybridization, used on Formula 2.3 for calculation of thermodynamic stability ΔG are presented in Table 2.3 [26].

The Formula 2.3 is used to calculate the stability of DNA duplex structure. The Δg_i parameter is the helix initiation free energy, and is 5 Kcal for duplexes containing G-C base pairs and 6 Kcal for duplexes composed exclusively of A-T base pairs. The Δg_{sym} parameter has a value of 0.4 Kcal for a duplex formed from a self-complementary sequence or 0 Kcal for a duplex formed from two complementary sequences. The Δg_x defines the stability of each two consecutive base pairs which is calculated with Formula 2.4.

Table 2.3 – Nearest-Neighbor thermodynamic values of enthalpy (ΔH°) and entropy (ΔS°) ΔH° units are kcal/mol and ΔS° is in cal/K per mol of interaction.

Interaction	ΔH°	ΔS°
$\overrightarrow{AA}/\overleftarrow{TT}$	9.1	24.0
$\overrightarrow{AT}/\overleftarrow{TA}$	8.6	23.9
$\overrightarrow{TA}/\overleftarrow{AT}$	6.0	16.9
$\overrightarrow{GA}/\overleftarrow{GT}$	5.8	12.9
$\overrightarrow{GT}/\overleftarrow{CA}$	6.5	17.3
$\overrightarrow{GT}/\overleftarrow{GA}$	7.8	20.8
$\overrightarrow{GA}/\overleftarrow{CT}$	5.6	13.5
$\overrightarrow{CG}/\overleftarrow{GC}$	11.9	27.8
$\overrightarrow{GC}/\overleftarrow{CG}$	11.1	26.7
$\overrightarrow{GG}/\overleftarrow{CC}$	11.0	26.6

$$\Delta G_{total} = -\left(\Delta g_i + \Delta g_{sym}\right) + \sum_x \Delta g_x$$

Formula 2.3 – Stability of DNA duplex structure [26].

Formula 2.4 calculates the stability Δg_x for each two consecutive base pairs with values from Table 2.3 where T is the environmental temperature in °K.

$$\Delta g_x = \Delta H° - T \times \Delta S°$$

Formula 2.4 – Stability of two consecutive base pairs.

The stability of a DNA duplex (ΔG_{total}) calculated by Formula 2.3 can be used as a limit condition to reject a probe for microarrays.

If undesired secondary structures, as hairpin-loops or self-annealing, are predicted to be formed in a specific probe, the probe can be rejected if these structures are above a certain level of stability, since this way the structures would be stable and the probe would not be available for hybridization.

2.3.4. Hairpin loops

Hairpin loops on microarray probes are a type of secondary structures [27] that occur when a probe has such a nucleotide sequence that it can hybridize with itself (Figure 2.12). This is not a desired happening, since this way the probe will not be available for hybridization with DNA test samples [28].

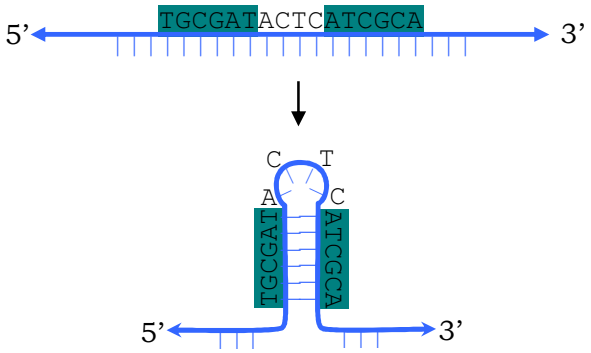


Figure 2.12 – Hairpin-loop formation.

The hairpin-loop formation must, therefore, be predicted by the software in order to accept or not a specific probe.

A specific probe will be rejected if the calculated Tm of any group of three or more consecutive bounded nucleotides is greater than the temperature at which the hybridization reaction takes place, designated here as room temperature (RT). This is to guarantee that possible secondary structures are not created at room temperature.

If the melting temperature calculated is less than RT, the connections predicted are denaturated (they are at a temperature above its Tm) and then the predicted connections will not occur. On the other hand, if the Tm of the predicted connections is greater than RT, the secondary structures can be already formed and they will not denaturate since they are at a temperature below it's melting temperature, and therefore, these probes will not be available for hybridization with the biological sample.

The stability of secondary structures (ΔG) can be calculated using Formula 2.3 with Δg_{sym} parameter equal to 0.4 Kcal since this is a duplex formed from a self-complementary sequence.

Either the Tm or the ΔG calculated for such group of nucleotides can be used as trigger to reject a probe.

2.3.5. Self-annealing

Self-annealing on microarrays probes is another type of secondary structures [27] that occurs when the probes in a particular spot of a microarray hybridize with each other. This can happen when the probes has regions of complementary sequence in the reverse direction (see Figure 2.13) [29].

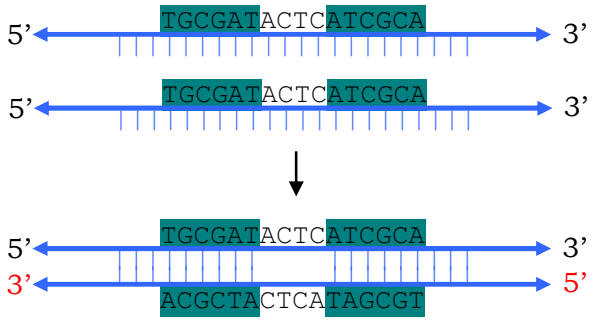


Figure 2.13 – Self-Annealing formation.

Similar to the Hairpin-loops formation, the Self-annealing formation must also be predicted by software and avoided.

Here probes will also be rejected if the calculated Tm of any group of three or more consecutive bounded nucleotides is greater then the room temperature for the same reasons.

The stability of this secondary structure (ΔG) can also be calculated using Formula 2.3, but now with Δg_{sym} parameter equal to 0 Kcal since this is a duplex formed from two complementary sequences.

As in Hairpin Loops formation, either the Tm or the ΔG calculated for such group of nucleotides can be used as a trigger to reject a probe.

2.3.6. Run/Repeat

Run/Repeat in the DNA sequence context represents the maximum number of sequential repeated nucleotides within a sequence. For example, the sequence *ATTGCT***CCCCC***AAAGTC* has a Run/Repeat of 5 (5 *C* bases appear sequentially in the sequence).

In some particular cases, there is the necessity to reject probes that have more than a specific Run/Repeat value. These regions are more likely to cause the formation of secondary structures [30].

2.3.7. BLAST

BLAST (Basic Local Alignment Search Tool) is an algorithm for rapid sequence comparison directly approximating alignments that optimize a measure of local similarity between two sequences [31]. This tool is useful for probe design once it allows testing for a possible hybridization between the found probes and

avoidable genes. The BLAST algorithm is available online on NCBI web site[1] or it can be run in a local computer with the executable file "*bl2seq.exe*" available for download on NCBI FTP site[2]. This algorithm will be discussed further in section 4.4.2.

2.3.8. Issue of overlapping probes

When designing probes for mutation detection, it can happen that mutations are next to each other and the probe overlap a mutations other than the one it was designed to detect. This can be a problem since if there are multiple mutations being overlapped by a given probe, the DNA sequence that it is supposed to hybridize to is not the expected and the probe may not detect properly the mutation for which it was designed.
This is an important issue that can be solved by designing enough extra probes (Formula 2.5) to cover all possible mutation arrangements.

$$\langle \text{Number of ExtraProbes} \rangle = 2^{\langle \text{number of overlaped probes} \rangle} - 1$$

Formula 2.5 – Number of necessary extra probes for each probe where mutation overlap occurs.

Take, for example, a probe of 15 nucleotides length centered in a given mutation, the nucleotide 10 of Figure 2.14. The mutation to detect is the "*g.10del*" (deletion of nucleotide 10 – "T"). The probe for the mutation *g.10del* overlaps the mutations *g.15C>T* and *g.17del*. As those mutations can be also present, or only one of them, or none, it is crucial to have some more extra probes to cover that possibilities as shown in Figure 2.14 at right side. The first probe is the "original" one, the one when assuming that the other two mutations are not present. The second one is when assuming that only the mutation *g.15C>T* is present being a *T* in the place of *C* in position 15. The third one is when considering that only mutation *g.17del* is present, and then the nucleotide 17 is not present (has been deleted) and all the following sequence slides one nucleotide to the left. The last probe is the one that considers the presence of both mutations g.15C>T and g.17del together with g.10del.

[1] http://www.ncbi.nlm.nih.gov/blast/bl2seq/wblast2.cgi

[2] ftp://ftp.ncbi.nlm.nih.gov/blast/executables

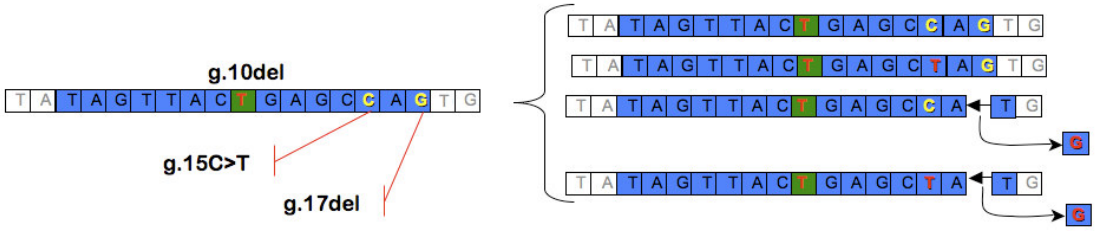


Figure 2.14 – Probe overlap over neighbor mutations.

This way, by designing the necessary probes for all arrangements (Formula 2.5) no matter the other two mutations occur or not, if the mutation *g.10del* is present, it will have more chance to be detected.

2.4. User requirements

The Biology Department of University of Aveiro in association with Biocant, are developing a DNA microchip for predicting the genetic predisposition for the familiar hypertrophic myocardiopathy disease by detecting mutations associated to the disease.

This and other similar projects have in common the need for bioinformatics tools to assist them in the probe design process.

Functional requirements are most described with formal methodologies such as the well known "*use cases*" [32, 33]. However, in this project we decided to use a non-formal specification. Considering the multidisciplinary characteristics of this area, and aiming to bring the users closer to the development process, the requirements were rather specified declaratively.

The following specifications were given by biologists working in microarray probe design for mutations detection when asked about what they would need and expected from a software application to help them in this process.

The desired software should have as input a list of mutations to be detected per gene and the respective genes sequence. As output, the software would present the probes that would be used in the microarray to identify the mutations present in the submitted sample.

The software should allow the adjustment of a set of parameters, defined by the user so that all designed probes may have the expected behavior and can be successfully used together in the same microarray.

2.4.1. User parameters

There are several parameters that must be satisfied by all designed probes in order to have similar hybridization conditions. These parameters along with some preferences were identified as being:

- Minimum and maximum nucleotide length of probes.
- Tm interval that probes must have.
- Maximum Run/Repeat length accepted on each probe.
- Room temperature, so that the probes do not form any secondary structures like hairpins or self-annealing.
- Possibility to design sense or anti-sense probes.
- Possibility to force the software to design probes with the mutations precisely at the centre of the oligo.
- Possibility to force the software to find the best probes even out of the predefined parameters.
- Possibility to design extra probes when there is an overlap of any probe with other mutations.

2.4.2. Input data

The software should be able to accept as input CSV excel-compatible files with lists of mutations (one per line) described in a standard nomenclature [11] (Table 2.2).

For the sequence of target genes, it must accept simple text files in FASTA [34] and GENBANK [35] formats.

2.4.3. Data processing

For each mutation of the input list, the software must design the best two probes, one to identify the wild type and another to identify the mutated type, according to the following conditions:

- The nucleotides number must be between specified boundaries.
- Probes should all fall within a defined Tm interval.
- Probes must have less nucleotides repeats than the maximum specified.
- It must be guaranteed that no secondary structures will be formed at room temperature.

- The probes should not hybridize with any other part of the same gene or with any other gene screened in the project.
- If a probe overlaps other mutations, the software should design extra probes, considering every possible mutations arrangement (Formula 2.5).

2.4.4. Output data

The output data (i.e. set of designed probes) should be visualized in a list which might be exported as a CSV Excel compatible file.

The list must show the probes and some relevant properties such as:

- Probe sequence in 5' → 3' direction.
- Start position in respective gene sequence.
- Length of probe (in number of nucleotides).
- Melting temperature according to the Formula 2.1 or Formula 2.2.
- Percentage of GC's bases (%GC) in the probe.
- Maximum Run Length found in the probe
- Estimated molecular weight of the probe
- Mutation description in HGVS format
- General information about the design process

For each probe it should be possible to view some other detailed information, such as:

- Detailed steps taken during each probe design.
- Thermodynamics details of each probe: visual details of self-annealing and hairpin predicted structures.
- BLAST details of each probe against each gene in the project: visual details of BLAST results.

There should also be a visual interface showing the position of the designed probes on the respective gene sequence.

2.5. Summary

This chapter has introduced the genetic principles behind microarray technology and the particularities to consider when designing probes for microarrays.

These probes placed on microarray substrate have to obey to particular conditions in order to successfully and specifically bind to the target DNA as, for

instance, having the same melting temperature, not forming secondary structures and not binding to genes other than the targeted one.

A description of the user needs and requirements in a software that would automatically design probes for using with microarrays for the specific task of mutations detection was also presented. The user requirements should be taken into account in the automatic design of probes, given a list of mutations to detect, the genes sequences with the targeted mutations and a set of parameters to consider in the design process.

3. Evaluation of existing tools for probe design

As there is a wide variety of bioinformatics tools, we decided to evaluate existing software related to probes design in order to understand, identify and confirm the missing functionalities pointed by the biologists.

We analyzed software referenced on the internet with the propose of identifying its ability to perform specific tasks such as sequence alignment, design of probes for species specific detection, check the probability of the hybridization of the designed the probes against a set of genes using BLAST algorithm, and design probes for mutation detection.

The tools were chosen based on publications in which they were referenced or based on its own publications.

In the following sections, we present briefly some features of the selected set: AlleleID, Oligowiz, Gencheck, Oligo Design, ROSO, Picky, Sarani and Visual OMP. We also present a cross comparison between them.

3.1. AlleleID

AlleleID is a commercial software developed by Premier Biosoft International[3], but can be tested through an available demo version. It makes alignments of multiple sequences and designs probes for species-specific detection. It then uses BLAST to screen possible alignments within any database server specified, as the NCBI database[4], for instance. (Figure 3.1).

This software is intuitive to use and has many configurable parameters for probes design. However, it is very poor in designing probes for detection of single nucleotide mutations in a high throughput manner, since it allows only to specify and design probes for SNP's one at a time, and it doesn't permit any action when there are probes overlapping multiple mutations.

[3] http://www.premierbiosoft.com

[4] http://www.ncbi.nlm.nih.gov/Database/

3.2. OligoWiz 2.0

OligoWiz 2.0 is a free software developed in Biocentrum of Technical University of Denmark[5] [6]. This is a Client-Server Java application with an easy-to-use GUI (Figure 3.2). It allows the design of probes for multiple proposes, and also supports gene sequence annotations [36].
However it possesses some limitations on the input sequences, since it only allows choosing the ones in the server, and it does not have any parameterization for the design of probes for mutations detection.

3.3. Genchek

Ocimum Biosolutions[6] has a set of bioinformatics tools for Biotech/Parma industry, which includes the Genchek software for sequence analysis (Figure 3.3).
This software is not available for download but, taking into account the internet site information, it is a very complete solution for sequence analysis, including data management, primers design and many features for BLAST alignment and sequence comparison. However it is not able to design probes for the detection of mutations.

3.4. Oligo Design

This is a free software developed in Maryland University[7] [37]. This software (Figure 3.4) automates some design aspects that improve the selection of probes for use in microarrays. This is a tool for assisting in probe design rather than a software dedicated to probe design.
It allows the alignment of multiple sequences and it designs probes for specific locations. It has many options for analysis of a given probe, including hybridization with a target sequence, information about hairpins and self-

[5] http://www.cbs.dtu.dk/services/OligoWiz2

[6] http://www.ocimumbio.com

[7] http://www.enme.umd.edu/bioengineering/

annealing structures formation and also about thermodynamics properties, but they are presented as individual tools, with poor integration between them.
There is also an option for designing probes for mutation detection but, once more, it only allows to specify SNP's and to design one single probe at a time.

3.5. ROSO

ROSO (Recherche et Optimisation de Sondes Oligonucléotidiques) is a free web service (Figure 3.5) provided by INSA – Lyon University and INRA – "Institut National de la Recherche Agronomique"[8], in France [38].
This software designs a specific set of probes to identify a given set of sequences for use in microarrays. It hands a set of constraints such as avoiding probes that could form stable secondary structures, guarantee no significant cross-hybridization and minimize the Tm variability of the probe set. It does not have any features for design of probes for mutation detection and is not intended for sequences alignment analysis.

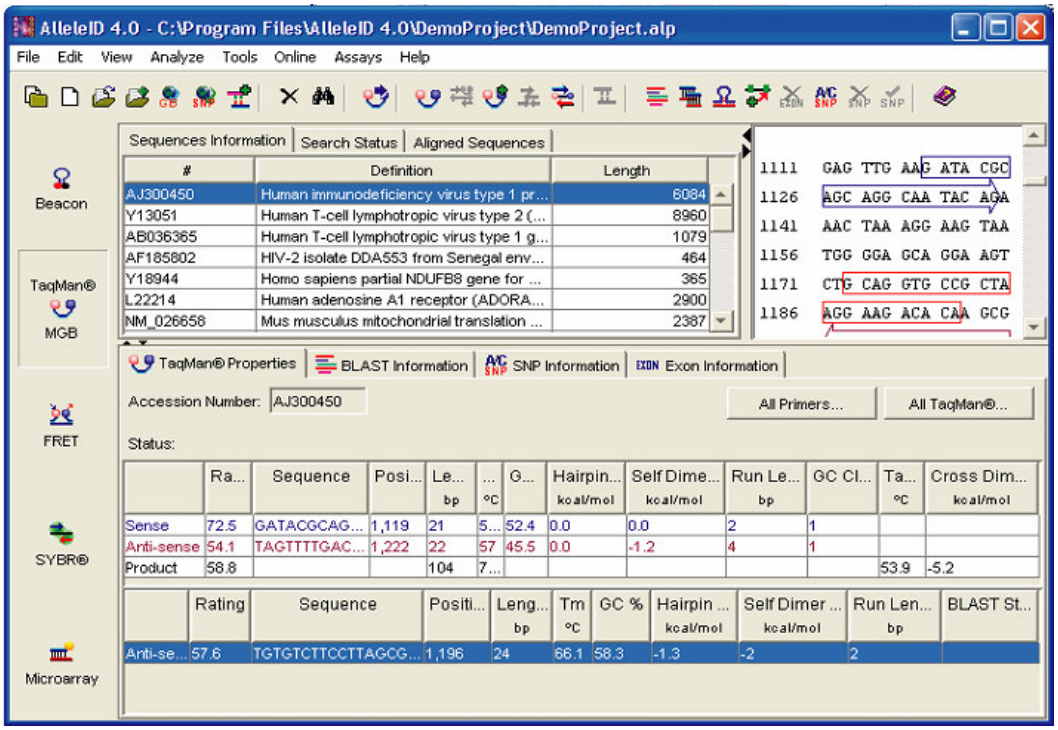


Figure 3.1 - AlleleID software.

[8] http://pbil.univ-lyon1.fr/roso

Figure 3.2- OligoWiz software.

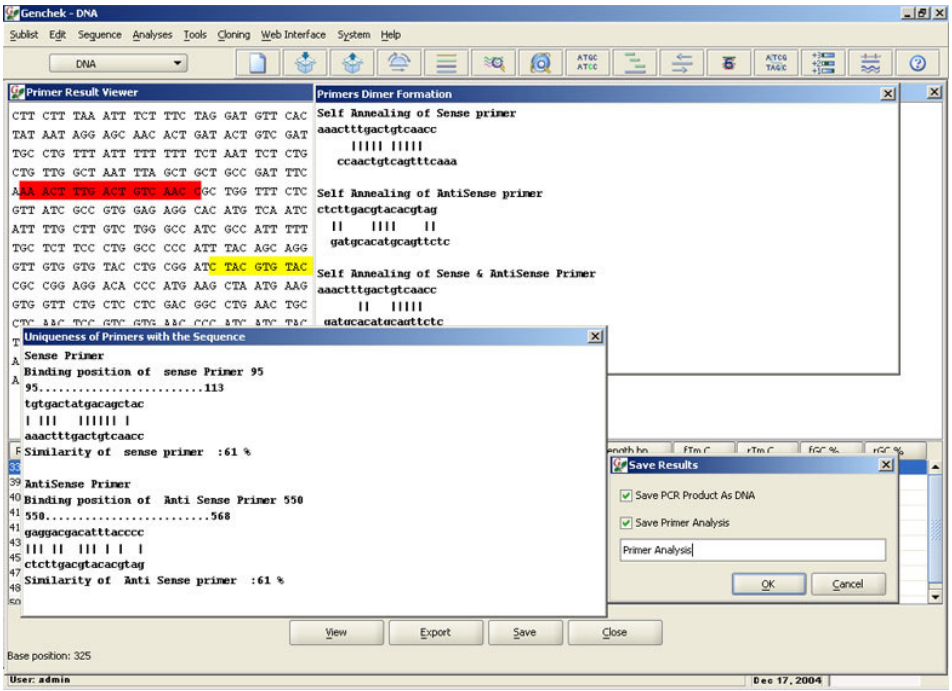


Figure 3.3 - Genchek software.

3.6. Picky

This is a software developed by the Iowa State University[9] [39]. It is a tool mainly intended to design microarray probes for large genomes. It is optimized to guarantee high specificity, sensitivity and uniformity for a given set of parameters. It does not support any kind of probe design for mutations detection. The GUI (Figure 3.6) is very appellative at the first sight, but it is complicated to understand the visual information associated with the results.

3.7. Sarani

Sarani is a commercial software for large-scale design of probes for microarrays developed by Strand Life Sciences[10] (Figure 3.7). It is not available for demo, but from their internet site information it is intended, as the Picky software, to design microarrays probes for large genomes only.
It does not refer to any kind of probe design for detection of mutations.

3.8. Visual OMP

Visual OMP is provided by DNASoftware and is a Nucleic Acid Structure and Primer/Probe Design commercial software[11].
According to the internet site information, it designs primers and probes for sequence identification using microarrays, it provides an intuitive graphical view (Figure 3.8) of secondary structures formation, and is optimized for reducing cross hybridization. There is no reference to the design of probes for mutation or SNP's detection.

---

[9] http://www.complex.iastate.edu/download/Picky/index.html

[10] http://www.strandgenomics.com/saranitour.html

[11] http://www.dnasoftware.com

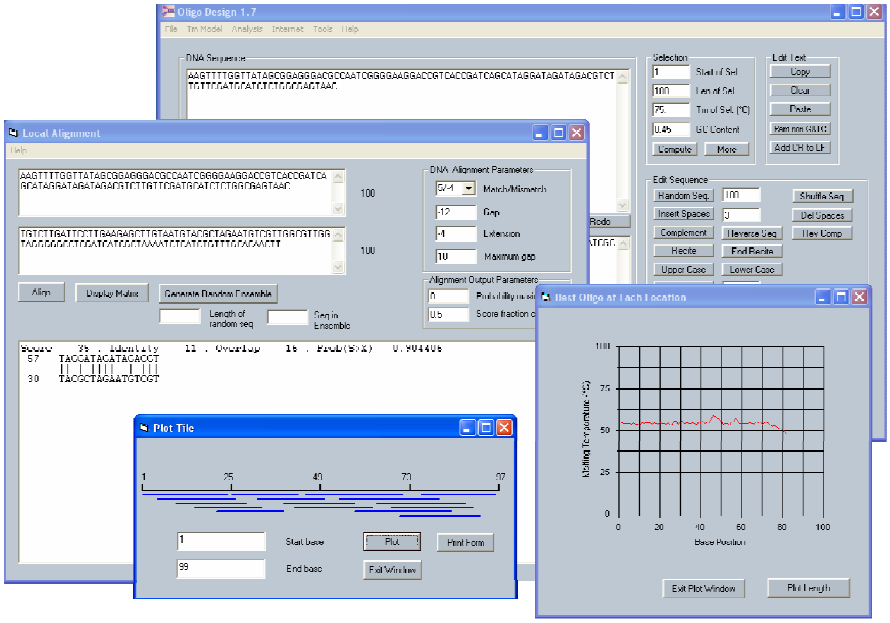


Figure 3.4 - OligoDesign software.

Figure 3.5 - ROSO software.

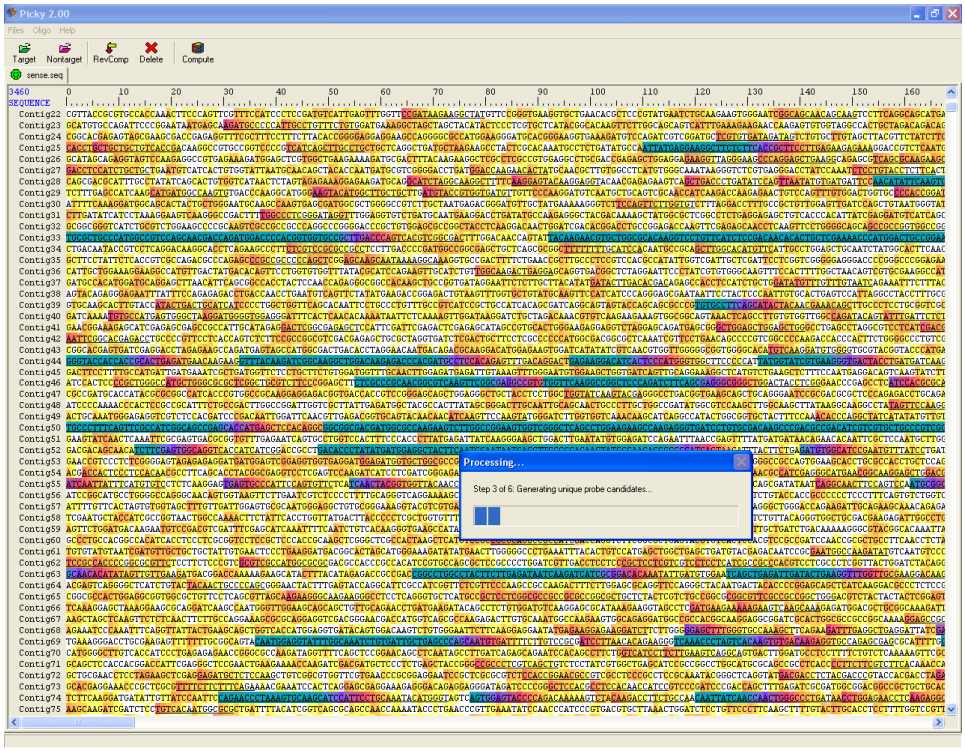


Figure 3.6 - Picky software.

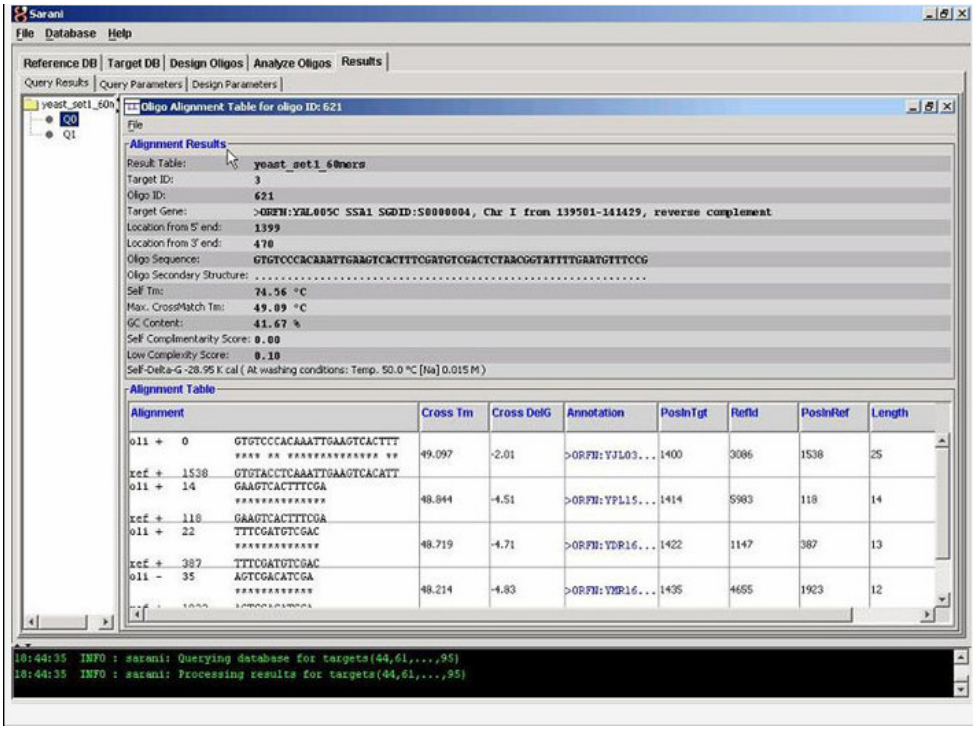


Figure 3.7 - Sarani software.

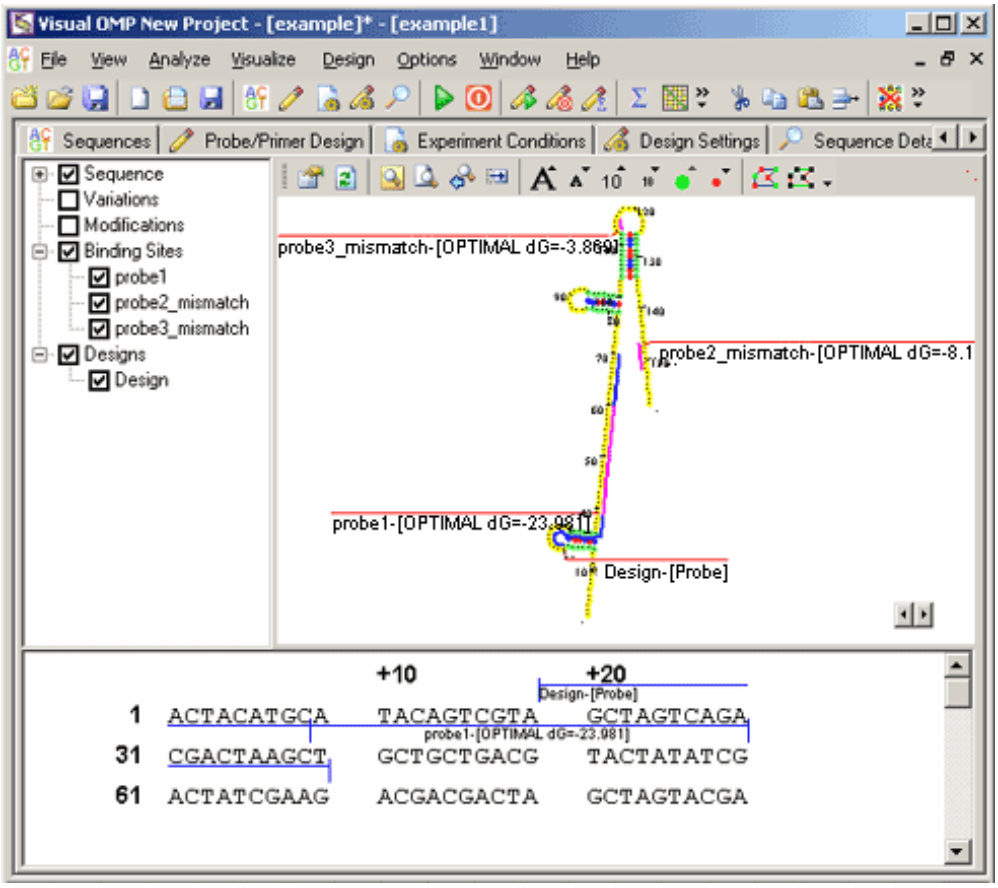


Figure 3.8 - Visual OMP software.

3.9. Cross comparison of the analyzed software tools

Table 3.1 presents an overview of the analyzed software in respect to the relevant features we have previously identified (see text above).

After analyzing the existing software, it was clear that a specific software to design probes for detection of mutations and SNP's was needed.

In fact, the applications that handled some kind of SNP probes design could only make one probe for a specific SNP, and no one could reach the following specific needs:

- Design of probes for other mutations more complex than SNP's
- Design simultaneously the complete set of probes for a set of mutations in order to accomplish the same probes specifications described in section 2.3.

- Check each probe for possible unspecific sequence alignments, insuring that it would not hybridize with other genes present in the analyzed sample.
- Possibility of designing extra probes to cover all arrangements when a probe overlaps other mutations in an automated manner.

Table 3.1 – Analyzed probes design software.

Software	License	Sequence Align	Probes Design	BLAST	Design probes for mutations
AlleleID	Commercial (DEMO available)	Multiple			Only for SNP's and 1 at a time
OligoWiz 2.0	Freeware				
Genchek	Commercial	Multiple	Primers Only		
Oligo Design	Freeware				Only for SNP's and 1 at a time
ROSO	Freeware				
Picky	Freeware		Confuse results		
Sarani	Commercial	Multiple (Not tested)	(Not tested)	(Not tested)	No reference
Visual OMP	Commercial	(Not tested)	(Not tested)	(Not tested)	No reference

Unavailable or unfavorable feature | Partially available or partially favorable feature | Available or favorable feature | Missing feature information

3.10. Summary

In this chapter, the results from the evaluation of existing probe-design software were presented, together with the missing functionalities found regarding the specific task of designing probes for mutation detection.

This research showed the necessity of a specific bioinformatic tool to assist such task, especially when the goal is to automate the design of large number of probes.

4. Mutation probe design software

After evaluating the existing software based on the user requirements and identifying the specific needs in mutation probe design, a bioinformatics tool was developed to fulfill the user requirements specified on section 2.4.

The software is meant to work upon the concept of individual projects, where the genes, mutations, options and all output data are organized together in a tree like scheme.

Each project information will be independently stored in a specified folder along with its files, containing the lists of mutations, the genes sequences, the results and a set of options including global parameters used for the probes design process and all project properties.

From the program workflow, presented in Figure 4.1, we can see that the process that leads the software to design a set of probes consists in creating a new project, adding genes and defining the probes design parameters, and, finally, specifying in these genes the location of the respective mutations. After all the data is inserted, the software will try to obtain all the probes that can be used in a microarray to identify mutations that are present in a particular sample. The software will also design the probes for detecting the respective wild species for control proposes.

This chapter presents the software structure, functionalities and algorithms. The GUI capabilities will also be explained.

4.1. Data structures

All the data are organized in XML and text files, following the common formats used on genomes (FASTA or GenBank).

As such, there is no special need for a database. However, to keep trace of experiments and to help organizing the data, it was decided to follow the project metaphor commonly adopted in programming frameworks.

When creating a project, the user specifies a name and a folder location for the project. The software will then create a new folder with a name equal to the project name inside the specified folder. All the data related with this project will be stored inside this new folder, which simplifies backup procedures. For

instance, to replicate a project or to copy it to another computer it is only necessary to copy the corresponding folder.

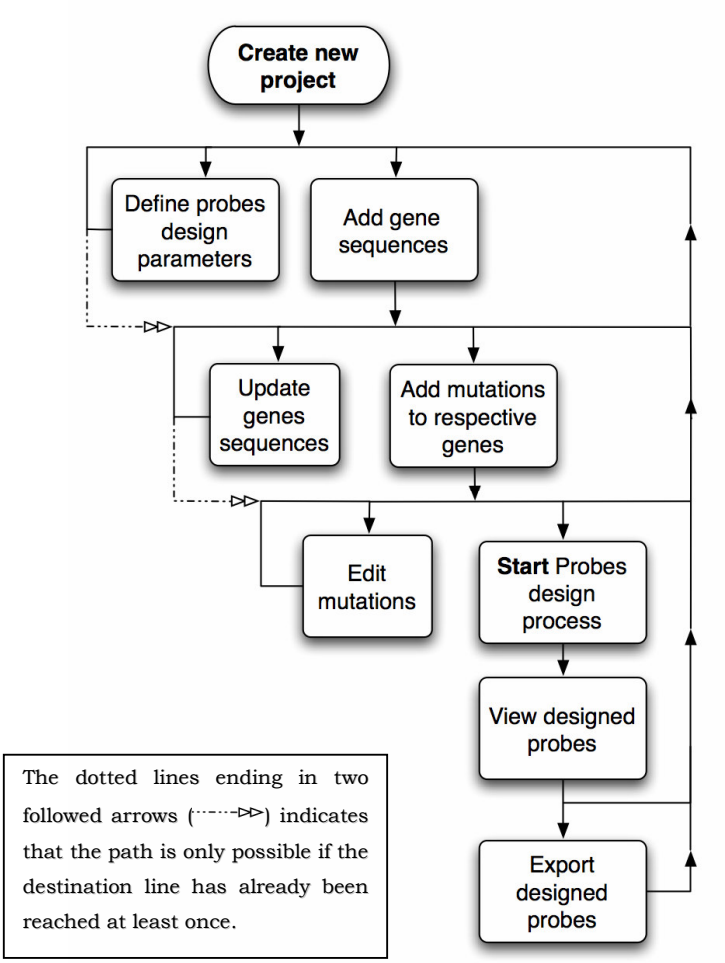


Figure 4.1 – Workflow from project creation to designed probes.

In Figure 4.2 is represented the basic data structure of a generic project named "<*ProjName*>" which was saved in a specific folder "<*SelectedFolder*>" selected by the user.

Inside the "<ProjName>" folder is a file with the same name as the project with extension "spd" ("<ProjName.spd>" on Figure 4.2) which is the file that exclusively represents the Project, the one to be selected when opening this project from the software.

The folder "Data" under the folder "<ProjName>" is where the entire project related data is stored.

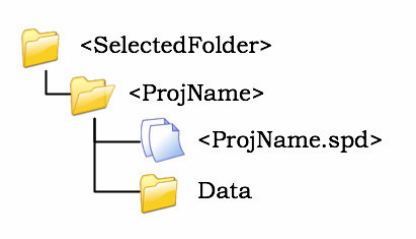


Figure 4.2 - Project file structure.

An example of a project structure is represented in Figure 4.3. Inside the data folder, we can find:

- Gene sequence related files
  - Sequences imported by the user as an example, for gene TSPYP5, the file: “TSPYP5.txt”
  - Temporary sequences generated by the software in case of mutation overlap: “OverlapGene1.txt”, ”OverlapGene2.txt”, ”OverlapGene3.txt”
- Mutations related files
  - Files with the mutation lists, either imported by the user or created by the software: “TSPYP5.csv”
  - Files with mutations already decoded: “TSPYP5.dml”
  - Temporary files of decoded files created during mutation list modification: “TSPYP5.tmp”
- Global project related files
  - One file with project properties such as dates, responsible persons, notes, default paths, etc.: “<ProjName.xml>”
  - One temporary file containing the above project properties created when exiting without saving: “proj.tmp”
  - One file containing the global options that will be used when designing the probes: “Global.opt”
  - One file with rules to decode the mutations input-file: “syntax.csv”
- Probes related files
  - Files with simple probes for each decoding file (“.dml”): “TSPYP5.mml”
  - One file with all the results obtained from the probes design process: “probes.glb”

- One auto-backup file with results of previous probe design processes: “probes.old”

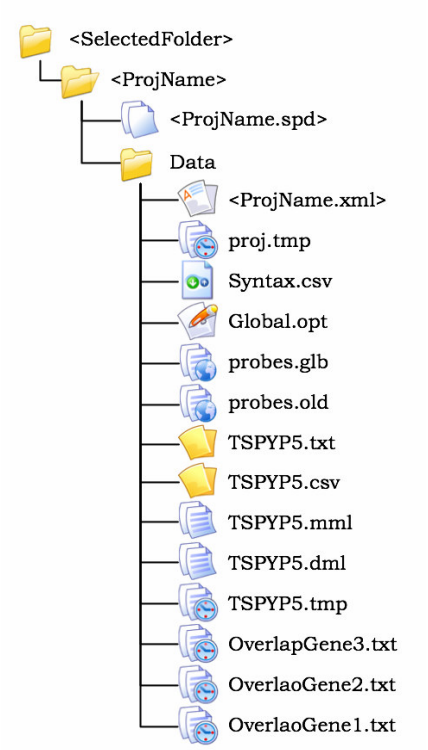


Figure 4.3 – Example of files under “Data” folder.

4.2. File types and data flux

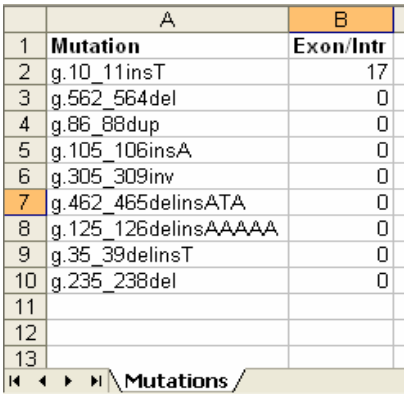
The software uses different file types along the process until it obtains the final probe list. The file types used are:

- Text files (.txt) – Simple text files containing gene sequence in FASTA or Genbank format.
- Comma Separated files (.csv) – Files compatible with spreadsheet software as Excel. They are mainly used for input and output data as described in sections 2.4.2 and 2.4.4.

  For inputting mutation data, these files must be formatted in the following manner (Figure 4.4):

  - The first line can be used as column header although the software will ignore it.

- From the second line and forward, the software will read one mutation per line with first column containing the mutation described in the standard nomenclature and the second column containing the exon number. This value will be merely informative.



	A	B
1	**Mutation**	**Exon/Intr**
2	g.10_11insT	17
3	g.562_564del	0
4	g.86_88dup	0
5	g.105_106insA	0
6	g.305_309inv	0
7	g.462_465delinsATA	0
8	g.125_126delinsAAAAA	0
9	g.35_39delinsT	0
10	g.235_238del	0
11		
12		
13		

Mutations

Figure 4.4 – Mutations list.

- Decoded Mutation List (.dml) – These are files in "csv" format (Figure 4.5).

```
Mutation;Position;Wild State; Mutated State;Exon/Intr
g.10_11insT;10_11;-;T;17
g.22_23delinsT;22_23;GG;T;11
g.35_39delinsT;35_39;TCACC;T;0
g.56_59del;56_59;AAAG;-;0
g.86_88insATT;86_88;-;ATT;0
g.86_88dup;86_88;TGT;TGTTGT;0
g.105_106insA;105_106;-;A;0
g.125_126delinsAAAAA;125_126;AT;AAAAA;0
g.235_238del;235_238;TCGA;-;0
g.305_309inv;305_309;CCAGC;CGACC;0
g.400_401insCAGT;400_401;-;CAGT;0
g.400_402insCAGT;400_402;-;CAGT;0
g.462_465delinsATA;462_465;TCCT;ATA;5
g.562_564del;562_564;GCC;-;0
g.2000_2001insATGG;2000_2001;-;ATGG;8
g.1a>t;;;;0
g.11g>a;;;;10
g.16t>A;16;T;A;0
```

Figure 4.5 – Decoded mutation list file (.dml).

- Merged Mutation List (.mml) – Files in XML format which handle a list of probes directly obtained by merging a mutation list with respective gene file (Figure 4.6).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ProbesFile GENEFILE="C:\SampleProj\Data\TNNT2.txt" PROJFILE="C:\SampleProj\Data\ProjectSample.xml"
            STATISTICS="" MUTSPROBESNUM="2" CHANGED="0" PROBESSTATE="2">
  <Options VALID="1">
    <Search MinLen="15" MaxLen="17" TargetTm="55" TmTol="5"></Search>
    <Basic CenterSNP="1" TFPOP="1" OVRLP="1" Sense="0"></Basic>
    <Advanced MaxDinLen="4" RoomTemp="16"></Advanced>
  </Options>
  <Mutations>
    <MUT0 MutDesc="g.10_11inst" Pos="10_11" Wild="-" Mut="T" Exon="17" Valid="1"></MUT0>
    <MUT1 MutDesc="g.22_23delinst" Pos="22_23" Wild="GG" Mut="T" Exon="11" Valid="1"></MUT1>
  </Mutations>
  <Probes>
    <PROBE0 Valid="1">
      <Wild Rating="0" Sequence="ACAACTTTAACCTGT" Posotion="4" Length="15" Tm="40" GC="33.3333" RunLength="3"
            MW="4510.9699" Valid="1" SNPPos="11" NextNtPos="19"></Wild>
      <Mut Rating="0" Sequence="ACAACTTTTAACCTG" Posotion="4" Length="15" Tm="40" GC="33.3333" RunLength="4"
           MW="4510.9699" Valid="1" SNPPos="11" NextNtPos="18"></Mut>
    </PROBE0>
    <PROBE1 Valid="1">
      <Wild Rating="1" Sequence="TGTGGAGGTCACGTA" Posotion="16" Length="15" Tm="46" GC="53.3333" RunLength="2"
            MW="4648.399" Valid="1" SNPPos="23" NextNtPos="31"></Wild>
      <Mut Rating="1" Sequence="CTGTGGATTCACGTA" Posotion="16" Length="15" Tm="44" GC="46.6666" RunLength="2"
           MW="4583.99" Valid="1" SNPPos="23" NextNtPos="33"></Mut>
    </PROBE1>
  </Probes>
</ProbesFile>
```

Figure 4.6 – Merged mutation list file (.mml).

- Global (.glb) – The files with extension "glb" are XML files containing a list of all probes designed by the software, and therefore global to the project (Figure 4.7).

```
<GlobalProbes PROJFILE="ProjectSample.xml" STATISTICS="9672:1381" PROBESNUM="7"
              COMBPROBES="1">
  <Options>
    <Search MinLen="15" MaxLen="17" TargetTm="55" TmTol="5"></Search>
    <Basic CenterSNP="1" TFPOP="1" OVRLP="1" Sense="0"></Basic>
    <Advanced MaxDinLen="4" RoomTemp="16"></Advanced>
  </Options>
  <Probes>
    <PROBE0 Valid="1" MutDesc="g.10_11inst" SNPPosDesc="10_11" OrigMut=""
            ProcessTime="2047" Type="4" Gene="TNNT2" GeneFile="TNNT2.txt">
      <Wild Rating="1" Sequence="AAACAACTTTAACCTGTGG" Posotion="2" Length="19"
            Tm="52" GC="36.8421" RunLength="3" MW="5795.7899" Valid="1"
            State="55" SNPPos="11" NextNtPos="21"
            StatusDesc="Probe Search Started...
            -&gt;TM of Probe was out of parameters: 40°C
            + Length increased by 2bp (one each side)
            -&gt;TM of Probe was out of parameters: 46°C
            =&gt;Probe has not been found inside Parameters!"
            VisualBlastDetails="
            ====== Gene: TNNT2 (No relevant BLAST Matches) ======

            ====== Gene: TSPYP5 (No relevant BLAST Matches) ======
            " VisualTdDetails="========== Self-Annealing Formation ==========
            5'-AAACAACTTTAACCTGTGG-3'
                     :  ||||  :
              3'-GGTGTCCAATTTCAACAAA-5'
                 Tm = 8°C    Score = 12       dG = 0.368 kcal/mol">
      </Wild>
      <Mut Rating="1" Sequence="AAACAACTTTTAACCTGTG" Posotion="2" Length="19"
           Tm="50" GC="31.5789" RunLength="4" MW="5770.7799" Valid="1"
           State="55" SNPPos="11" NextNtPos="20"
           StatusDesc="Probe Search Started...
```

Figure 4.7 - Global probes file (.glb).

- Options (.opt) – An XML formatted file holding the user defined options (Figure 4.8).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<GlobalOptions>
  <DefaultSearchProbesOptions>
    <Search MinLen="15" MaxLen="17" TargetTm="55" TmTol="5"></Search>
    <Basic CenterSNP="1" TFPOP="1" OVRLP="1" Sense="0"></Basic>
    <Advanced MaxDinLen="4" RoomTemp="16"></Advanced>
  </DefaultSearchProbesOptions>
</GlobalOptions>
```

Figure 4.8 - Options file (.opt).

The data will flow from the input files of mutation lists to one XML file containing all designed probes, passing by the different file types as shown in Figure 4.9 and described below:

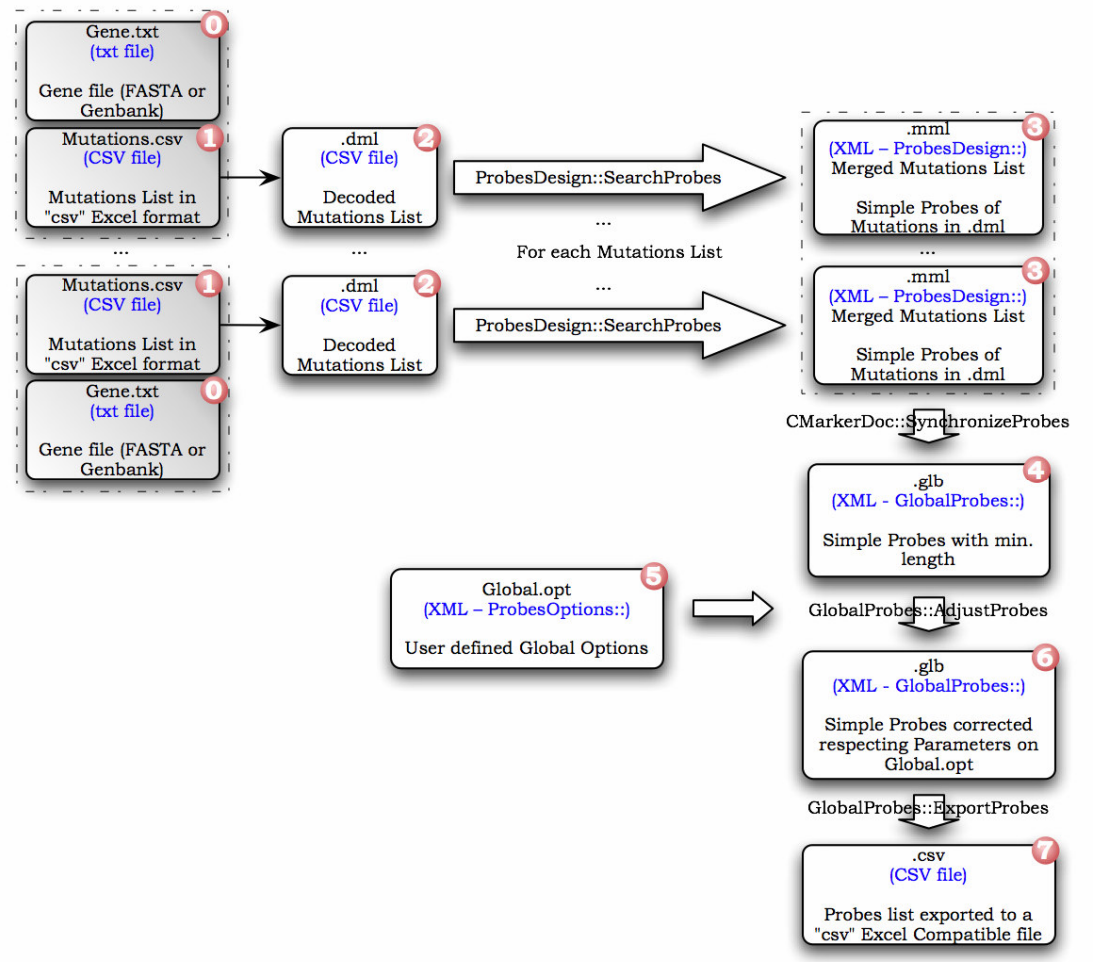


Figure 4.9 – Data flow within file types.

For each mutation file inserted in the project (Figure 4.9-1), a new file with the respective decoded mutations will be created (Figure 4.9-2). This step will validate the input mutations according to the nomenclature described in Table 2.2. The decoding process consists in reading all the mutations described in mutations list (Figure 4.9-1) and validating each one by looking for syntax errors and consulting the respective gene sequence file (Figure 4.9-0) to check if the nucleotides in the mutation position correspond to the ones described in the mutation. For each mutation for which these conditions are accomplished, the decoded file (Figure 4.9-2) will contain one more line with the following information:

```
Mutation;Position;Wild State;Mutated State;Exon/Intr
```

If the decoding process for a specific mutation has not succeeded, the line added to the decoded file (Figure 4.9-2) will have only the mutation description and all the other fields will be empty:

```
Mutation;;;;
```

When the "*Search Probes*" process is initiated, for each decoded file it will be created a merged mutations list in XML format (Figure 4.9-3) containing a simple probe for every valid and selected mutation. These probes are called "simple" since each one is a first attempt based only on the selection of a certain number of consecutive nucleotides from the original sequence centered on the mutation.
In this file (Figure 4.9-3), there will be also other relevant information needed for posterior processing.
When the software invokes the "*SynchronizeProbes*" method, all probes of the merged mutation list files (Figure 4.9-3) will be placed into one single Global Probes File named "Probes.glb" (Figure 4.9-4).
The probes in the Global Probes file (Figure 4.9-4) will be adjusted by applying project options represented in the file "Global.opt" (Figure 4.9-5). After this adjustment process (described in 2.4.3) the adjusted file (Figure 4.9-6) will now have all probes information.
The designed probes can then be exported to a "csv" Excel compatible file (Figure 4.9-7). The output file will be as shown in Table 4.1.

Table 4.1 - Output probes list exported to a “csv” file.

Gene	Num	State	Sequence	Position	Length	Tm	GC	RunLength	MW	Mutation
TNNT2	1	Wild	CCACAGGTTAAAGTTGTTT	2	19	52.00	36.84	3	5795.79	g.10_11inst
		Mutated	CACAGGTTAAAAGTTGTTT	2	19	50.00	31.58	4	5770.78	
	1.1	Wild	CCACTGGTTAAAGTTGTTT	2	19	52.00	36.84	3	5804.79	g.10_11inst
		Mutated	CACTGGTTAAAAGTTGTTTA	1	20	50.00	30.00	3	6083.98	
	2	Wild	ATACGTGACCTCCACAG	15	17	52.00	52.94	2	5241.43	g.22_23delinst
		Mutated	GTACGTGAATCCACAGG	15	17	52.00	52.94	2	5161.39	
	2.1	Wild	ATACGTGACCTCCACTG	15	17	52.00	52.94	2	5250.43	g.22_23delinst
		Mutated	GTACGTGAATCCACTGG	15	17	50.00	52.94	2	5170.39	
	3	Wild	GACTGTGGTGATGGATA	29	17	50.00	47.59	2	5074.34	g.35_39delinst
		Mutated	GAGACTGTATGGATACTA	29	18	50.00	38.89	2	5433.57	
	4	Wild	CTGAACAGGCAATACTG	555	17	50.00	47.59	2	5176.40	g.562_564del
		Mutated	CCCTGAACAAATACTGC	554	17	50.00	47.59	3	5256.44	
	5	Wild	CAGGCAGCAAGAGAAGA	1993	17	52.00	52.94	2	5094.37	g.2000_2001insatgg
		Mutated	CGCAGCACCATAGAGAA	1995	17	52.00	52.94	2	5183.50	
	6	Wild	ACCTCCACAGGTTAAAG	8	17	50.00	47.59	3	5216.42	g.16t>a
		Mutated	ACCTCCACTGGTTAAAG	8	17	50.00	47.59	3	5225.42	
	6.1	Wild	GAATCCACAGGTTAAAGT	7	18	50.00	38.89	3	5464.59	g.16t>a
		Mutated	TGAATCCACTGGTTAAAGT	7	19	50.00	36.84	3	5786.79	
	6.2	Wild	CCTCCACAGGTTAAAAG	8	17	50.00	47.59	4	5216.42	g.17t>a
		Mutated	ACCTCCACTGGTTAAAAG	8	18	50.00	44.44	4	5529.62	
	6.3	Wild	GAATCCACAGGTTAAAAG	8	18	50.00	38.89	4	5455.59	g.17t>a
		Mutated	TGAATCCACTGGTTAAAAG	8	19	50.00	36.84	4	5777.79	
TSPYP5	7	Wild	AAGGGCGCTGGGAGC	92	15	52.00	73.33	3	4464.95	g.99del
		Mutated	AAAGGGCGTGGGAGC	92	15	50.00	66.67	3	4439.94	

4.3. User interface

The software was made with an interface as intuitive as possible by designing a friendly GUI. The workspace was divided in tree views as shown in Figure 4.10.

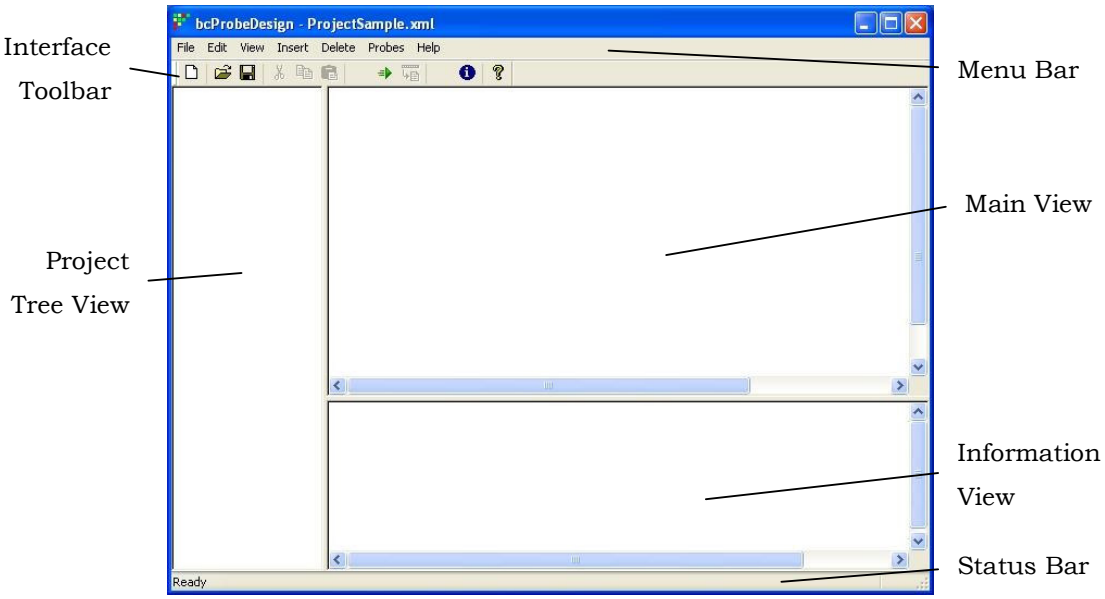


Figure 4.10 – Application workspace.

On the menu bar, the user can find the commands that allow the managing of the projects. The status bar indicates more specific information about the

commands to be performed. More frequent actions are also accessible as icon buttons in the toolbar (Figure 4.11).

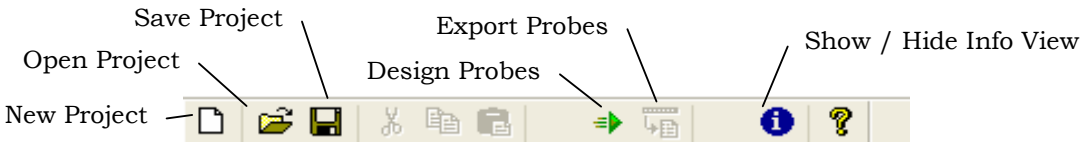


Figure 4.11 – Interface toolbar.

4.3.1. Project tree view

The Project Tree View (Figure 4.10) shows an organized view of all items present in the project and allows opening them directly by a mouse click.

The Project Tree appears when opening an existing project or when creating a new one. These options are accessible from the File Menu (Figure 4.12) and from the toolbar (Figure 4.11). The Project tree is filled with items when an existing project is opened, genes or mutations are inserted in the project or when new results came from the processing algorithm.

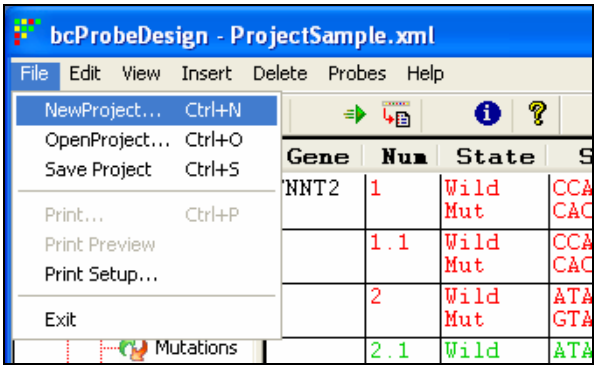


Figure 4.12 - File menu.

When creating a new project, a dialog box (Figure 4.13) is presented in order to enter information about project name, storage folder and creator name.

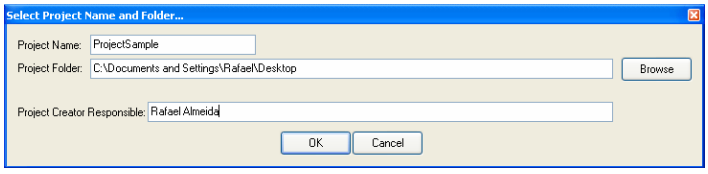


Figure 4.13 - New Project dialog box.

The project is represented by the tree structure which defines the relation between items. As shown in Figure 4.14, one project has associated options that are global to the project, different genes and a list with probes designed for current specified genes and mutations.

Each gene has also associated a list of mutations, a graphical view of target mutations placement within the gene and a graphical view of designed probes.

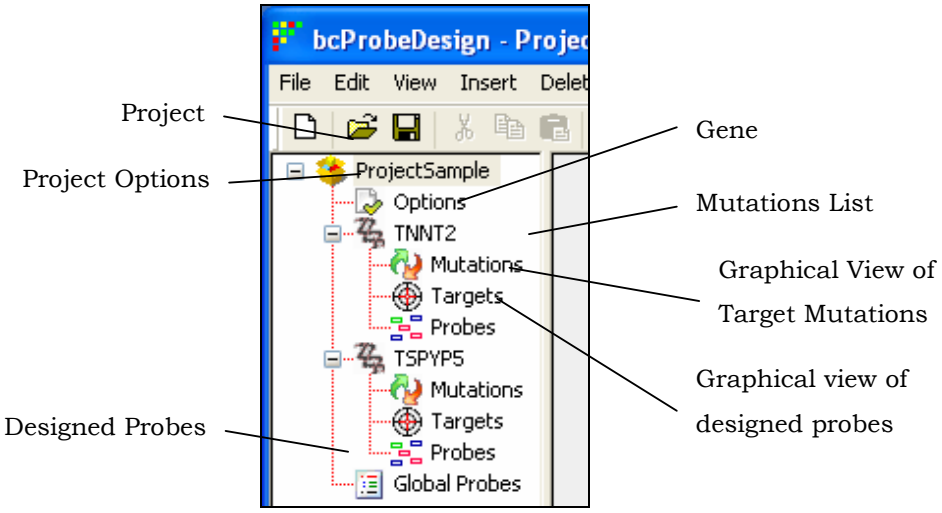


Figure 4.14 - Project tree view and item types.

By clicking on an item in the tree view, the respective data and options are displayed in the Main View.

The symbols (+) and (-) that appears on some nodes allows, respectively, to expand and collapse the sub-tree of that node.

From the View menu, it is also possible to collapse and expand all Project Tree and hide or show the toolbar, status bar and info view (Figure 4.15).

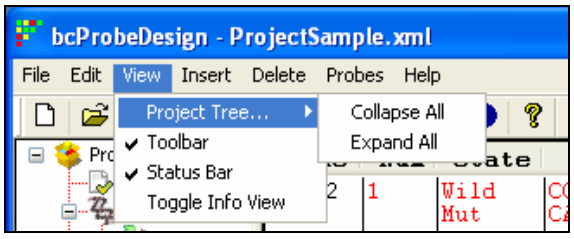


Figure 4.15 - View menu.

Genes or mutations can be imported or removed through the Insert and Delete menu, respectively (Figure 4.16)

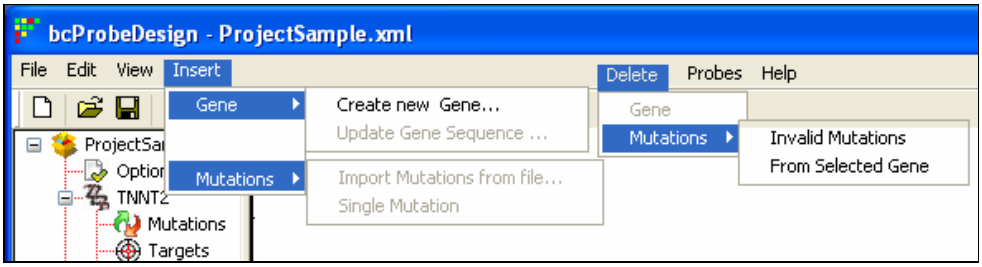


Figure 4.16 – Insert and Delete menus.

There is also the possibility to Insert and Delete a gene from the context menu of Tree View as shown in Figure 4.17.

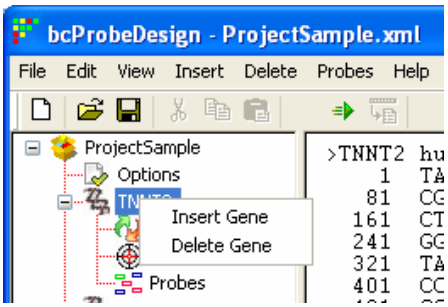


Figure 4.17 - Gene context menu.

4.3.2. Information view

The information view (Figure 4.10) is directly related to the main view, and it displays information associated with the data presented or selected in the main view.

The next section will describe the contents of this frame according to the different types of data presented in the main view.

4.3.3. Main view

The Main View (Figure 4.10) is a multipurpose view since it shows the information related to a selected item in tree view. Each type of item (Figure 4.14) (Project Root, Option, Genes, Mutations List, Targets, Probes and Global Probes) is associated with a specific type of view which is displayed in this view accordingly.

On next points of this section are described each possible types of main view.

4.3.3.1.Project general information

After opening a project, or by selecting the root project item in the Tree View, the main view will display the project general information (Figure 4.18):

- Project path – Local computer path of project location.
- Responsible personnel
- History summary, such as creation date, last saved date and last probe design
- Editable text notes

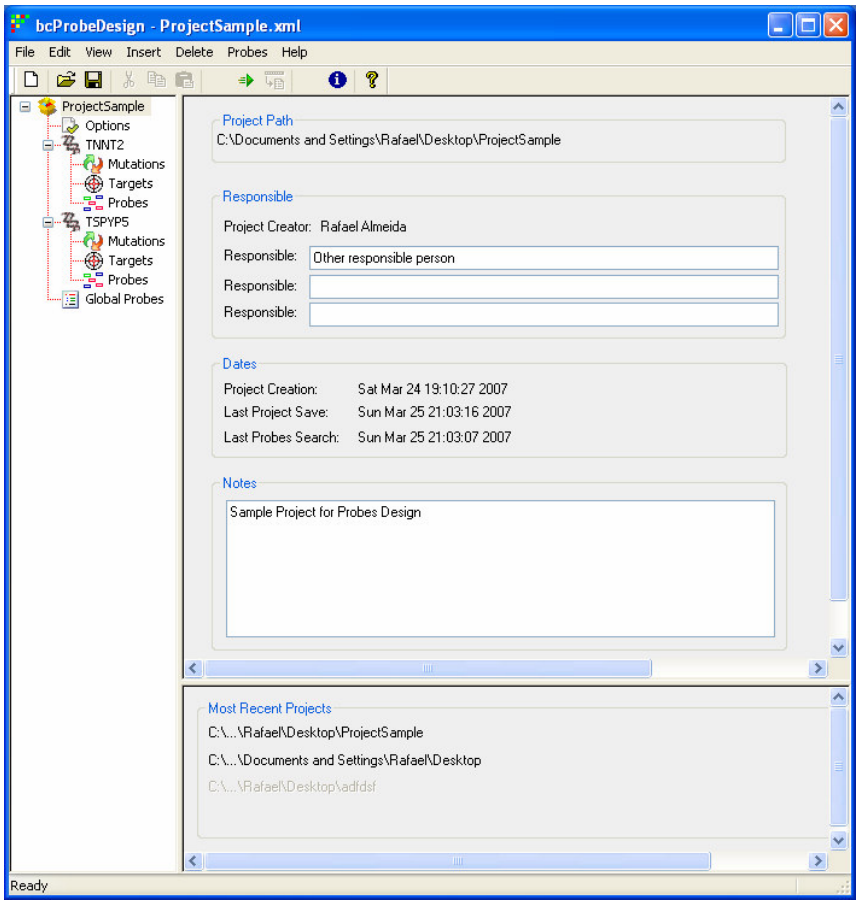


Figure 4.18 - General project information.

The info view shows the four most recent projects, which simplifies the user task to edit one of the last projects.

4.3.3.2.Probes design options

When selecting the Options item in Tree View, a form is displayed in main view, allowing the user to define the parameters required for the design of the probes (Figure 4.19).

These parameters are global to the active project and will be the guidelines for the software algorithms when designing probes.

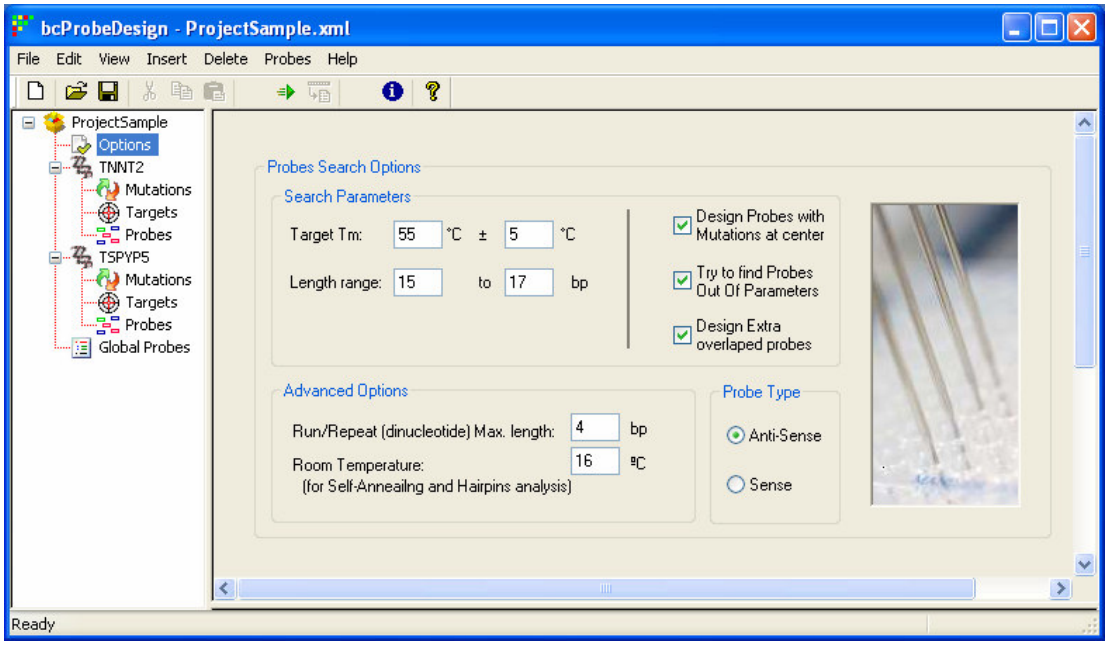


Figure 4.19 - Options view.

The project options are divided into three groups:

1. Search Parameters
    - **Target Tm** and allowed deviation (ΔTm) – The range of melting temperature that all designed probes must verify.
    - **Length** – The range of nucleotides length that all probes must accomplish.
    - **Design probes with mutations at center** – Indicates, when designing probes, if the mutation must be placed at the center of the probe. This implies that the probes must have an odd number of nucleotides.
    - **Try to find probes out of parameters** – Indicates that even if the software could not design a probe with the specified parameters, it will try to design one as close to the parameters as possible. In the output list, these probes will be specially marked with red color.

- **Design extra overlapped probes** – Indicates that when a probe intercepts one or more neighbor mutations, extra probes will be designed in order to have all possible arrangements of gene sequence regarding the mutations present (see section 2.3.8).

2. Probe Type
   - **Sense/Anti-Sense** – Specifies the directionality of probes to present at the output (see 2.1.2).
3. Advanced Options
   - **Run/Repeat (dinucleotide) max length** – Specifies the maximum number of consecutive repeated nucleotides allowed in a probe (see 2.3.6).
   - **Room Temperature** – Temperature at which the hybridization reaction takes place. This value will be used to predict the formation of secondary structures as Self-Annealing and Hairpin Loops (see 2.3.4 and 2.3.5).

4.3.3.3.Gene sequence

When a gene is selected, the main view shows the original content of the imported file containing the gene sequence (Figure 4.20).

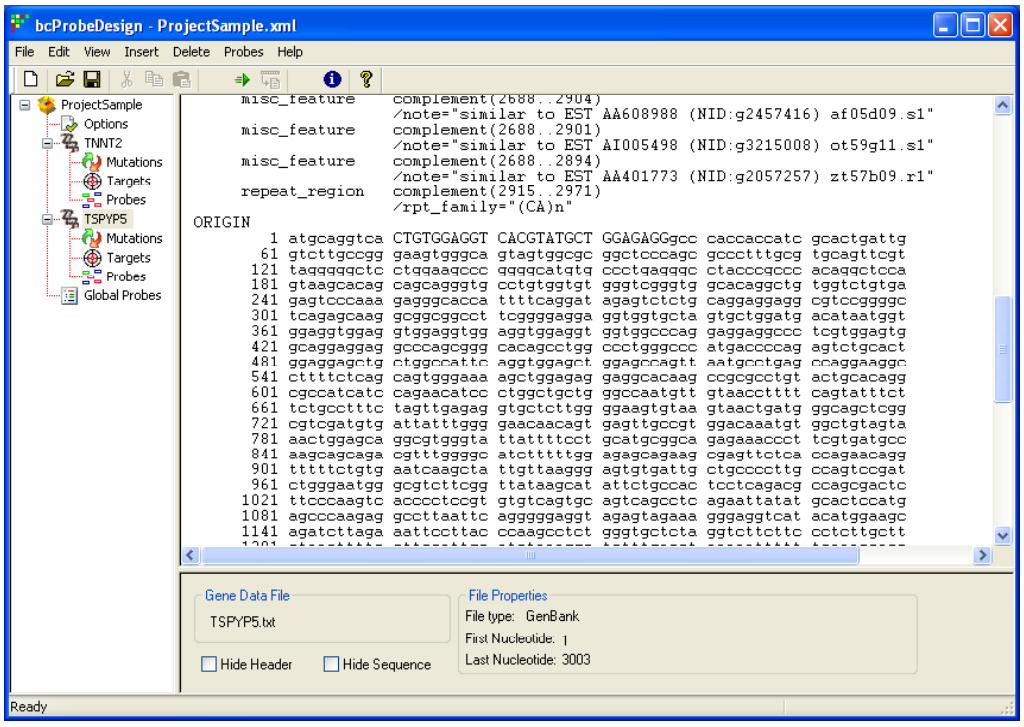


Figure 4.20 - Gene view.

The information view (on the bottom of the workspace) shows the file name from where the sequence was imported, the original file type (FASTA or GenBank) and the number of the first and last nucleotide of the sequence.
There are also options to hide/show the header and the sequence.

4.3.3.4.List of mutations

Genes can be inserted without any associated mutation list. In this case, the sub-tree below the gene does not exist. This can be useful if we have genes that will be present in the assay and we want to avoid the designed probes to hybridize with them.
On the other hand, if we specify any mutation (imported from a file or inserted individually), the sub-tree with items Mutations and Targets will appear below the Gene.
When the Mutations item is selected, the main view displays the list of mutations (Figure 4.21) already decoded.
For each mutation it is shown:

- The nucleotides number where the mutation occurs (the format X_Y indicated a range of nucleotides between positions X and Y).
- The wild version of nucleotides sequence on that position (the symbol "-" indicates that there will occur an insertion of nucleotides in the specified position).
- The Mutated version of the nucleotide sequence on the specified position (the symbol "-" indicates that the nucleotides between specified positions have been deleted).
- The Exon / Intron of the mutation. This is purely informative for the user since the software never uses this information during probe design.

When mutations cannot be decoded for some reason (for example wrong syntax or out of sequence range), they are marked with a red line over them and made unavailable.
The selection box on every valid mutation allows the indication of which ones the probes will be designed.

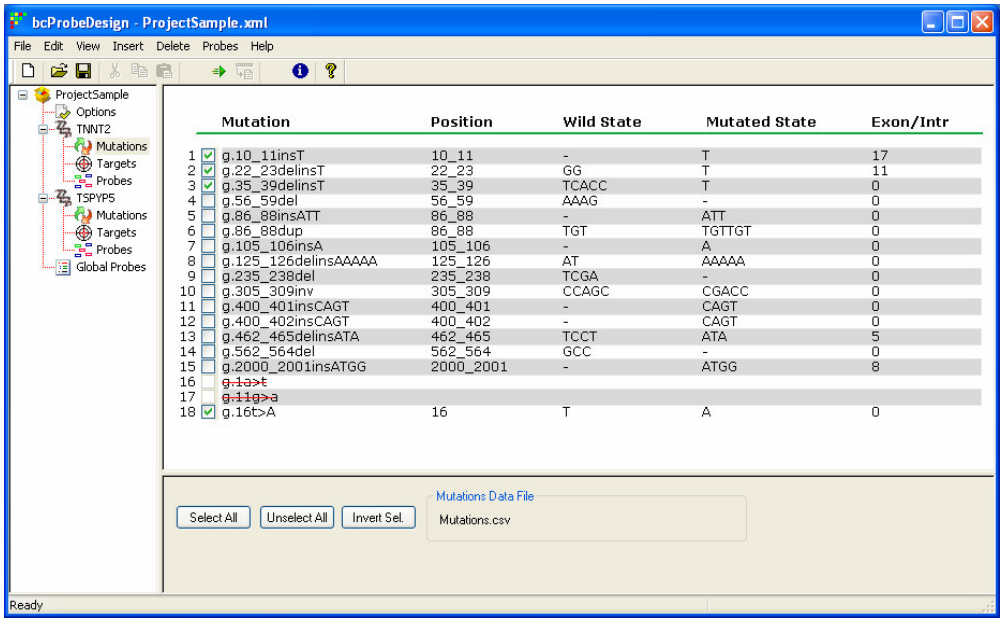


Figure 4.21 - Mutations view.

The information view (see Figure 4.22) shows the file that handles the mutation list and has three buttons, allowing to:

- Select all mutations.
- Unselect all mutations.
- Invert the selection.

There is a context menu for this view by right clicking on the mouse pad, as shown on Figure 4.22.

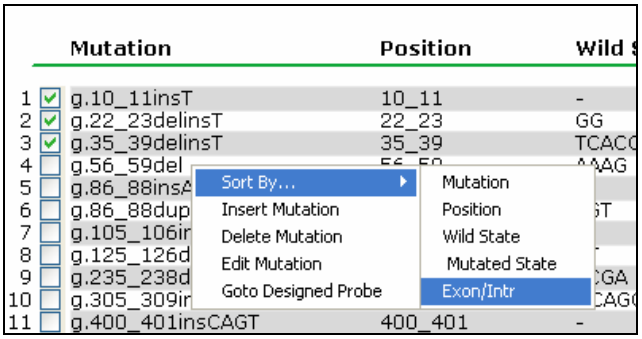


Figure 4.22 - Mutations context menu.

Through this context menu, the following operations are available:

- **Sort By...** - Allows sorting the list by any selected field on submenu.
- **Insert Mutation** – Shows the dialog box of Figure 4.23 in order to add a single mutation to the list.

- **Delete Mutation** – Deletes the mutation which was right clicked
- **Edit Mutation** – Opens the dialog box shown on Figure 4.23, containing information from the right clicked mutation, and allowing changes to it.
- **Goto designed Probe** – If probes were already designed, the Main View will display the Global Probes View and the probe designed for the right clicked mutation will be selected.

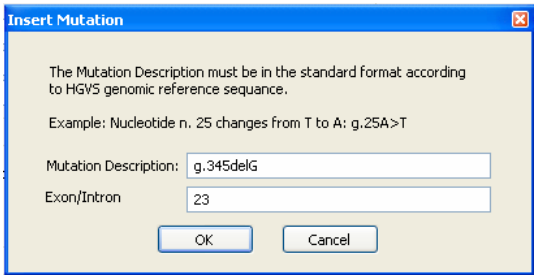


Figure 4.23 – Dialog box for single mutations insertion.

The dialog box for single mutations insertion asks for the mutation description according to HGVS standard format and the exon/intron number, which is optional and set to "0" if left empty.

4.3.3.5.Representation of targeted mutations in the gene sequence

When mutations exist for a specific gene, the targets item is available and, by clicking on it, the main view shows the gene sequence with the specified mutations highlighted (Figure 4.24) by a color code representing the type of mutations according to the legend on information view.
The Information view placed on the side of the legend of mutations colors, shows the options defined in Options View.

4.3.3.6.Location of the probes in the gene sequence

After invoking the command to design probes (Figure 4.11), the item Probes appears on the tree view below the genes with the designed probes associated to it. When Probes item is selected, the main view shows the same information as in the mutations target view (Figure 4.24), and also the placement of the designed probes (Figure 4.25).
Due to the complexity that would result from it, the overlapped probes are not shown by graphical means. This view is intended to be simply a quick view of the

general placement of the probes within the gene sequence, and therefore it is shown only one probe per mutation.

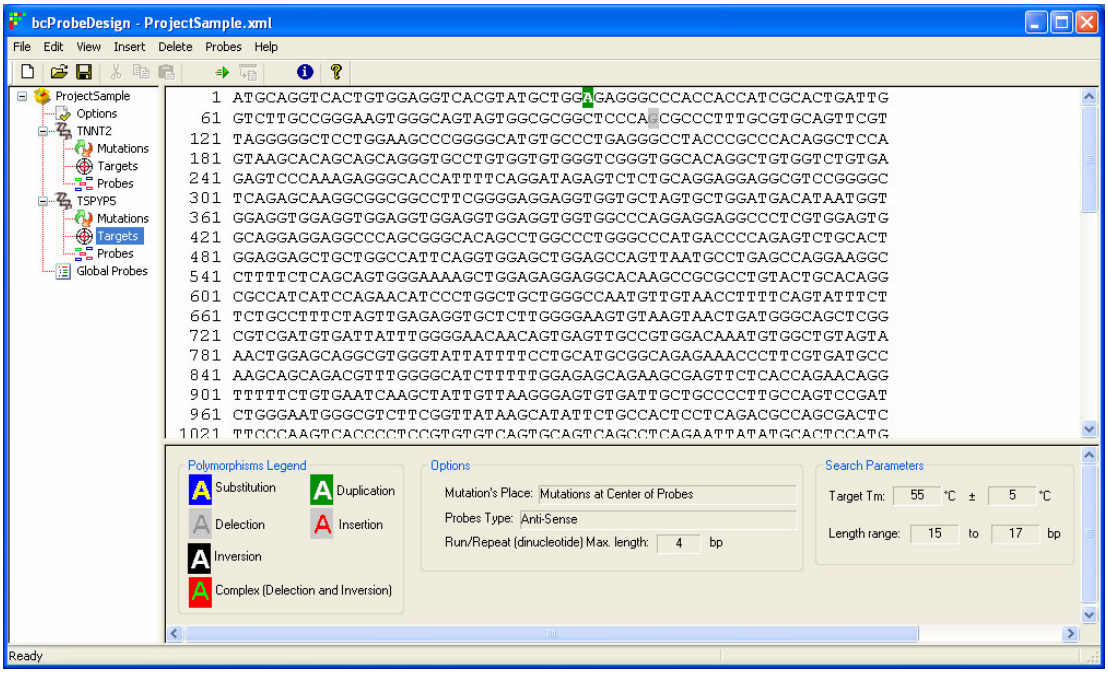


Figure 4.24 - Mutations target view.

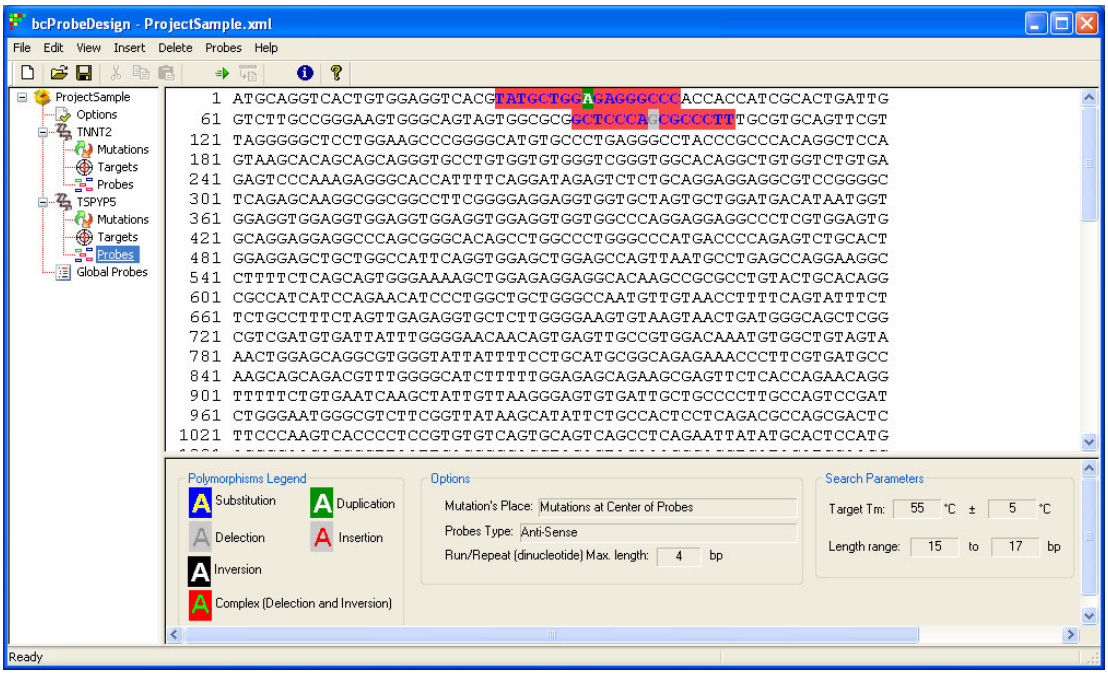


Figure 4.25 - Probes placement view.

4.3.3.7.Global probes list

After invoking the design probes command (Figure 4.11 or Figure 4.26), the software starts to design the probes while showing a progress dialog box (Figure 4.27) indicating the overall process and the current probe progress. On this dialog box it is also possible to cancel the operation.

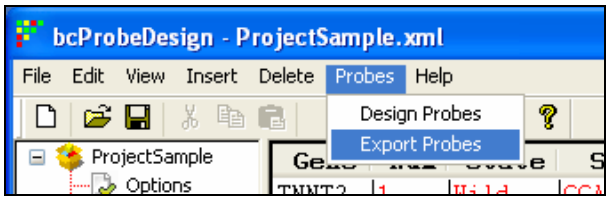


Figure 4.26 - Probes menu.

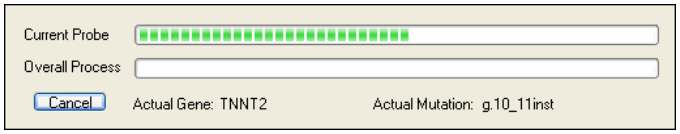


Figure 4.27 - Probes design progress.

After the designing process is finished, the item Global Probes appears on the tree view below the project item. When selecting it, the main view will display a list with all the designed probes (Figure 4.28).

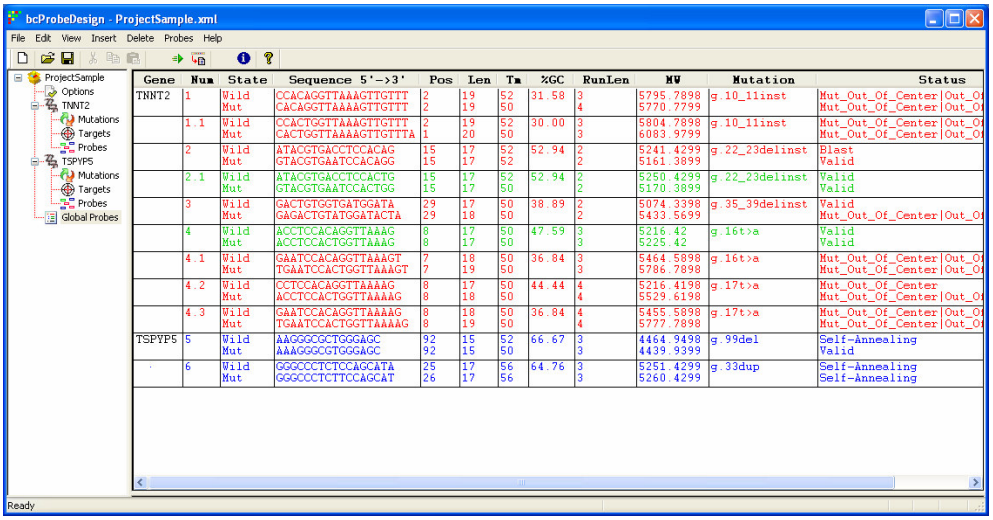


Figure 4.28 - Global probes view.

Each row contains two probes and the respective information, one to detect the wild type and the other to detect the mutated sequence.

Probes appear grouped by gene and are numbered from 1 to the number of the selected mutations. The first probe of each gene is identified by the gene name on the first column. If extra overlapped probes were designed, they are sub-numbered with first level having the same number as the original one (without any mutations overlapped) and the second level numbered from 1 to the number of extra probes designed (i.e. 1.4 indicates that is the 4th probe resultant from overlap of probe number 1 with neighbor mutations).

The text color of the rows indicates the quality of the probes. There are four possible colors:

- **Green** – Both probes for detection of wild and mutated type are valid and are inside the specified parameters on options view.
- **Blue** – At least one of the two probes couldn't be designed within the advanced parameters defined in options view. The reasons why it failed are summarized in the last column and details are described in the information view.
- **Red** – At least one of the two probes couldn't be designed within the basic search parameters specified in options view.
- **Grey** – At least one of the two probes were not in conditions to be accepted for adjustment by the respective algorithm. This could happen, for example, if it was not possible to create a simple probe regarding only the minimum probe length. The simple probe concept will be described ahead in section 4.4.1.

On this view (Figure 4.28) the information view shows the detailed information about the two probes of the selected row on the global probes view, as well as the time spent designing the probes. There are tree types of information that can be chosen by a set of three option-buttons (see details on Figure 4.29, Figure 4.30 and Figure 4.31):

- **Search Details (**Figure 4.29**)** – Detailed information about all steps taken by the probe adjustment algorithm, including the reasons of design failure, if any.

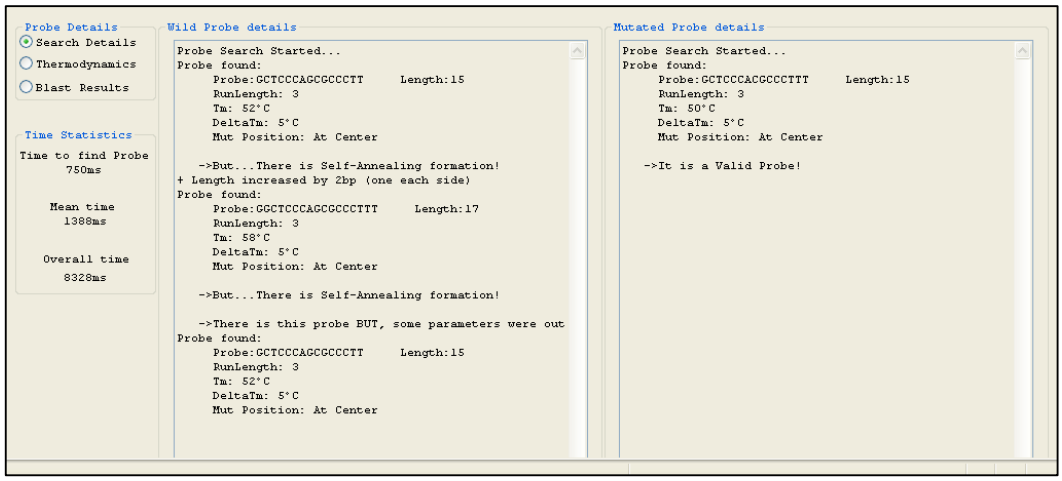


Figure 4.29 – Info view – Probe search details.

- **Thermodynamics (**Figure 4.30**)** – Comprehensive results view of prediction algorithms for hairpin loops and self-annealing formation for the two probes of the selected row on the global probes list.

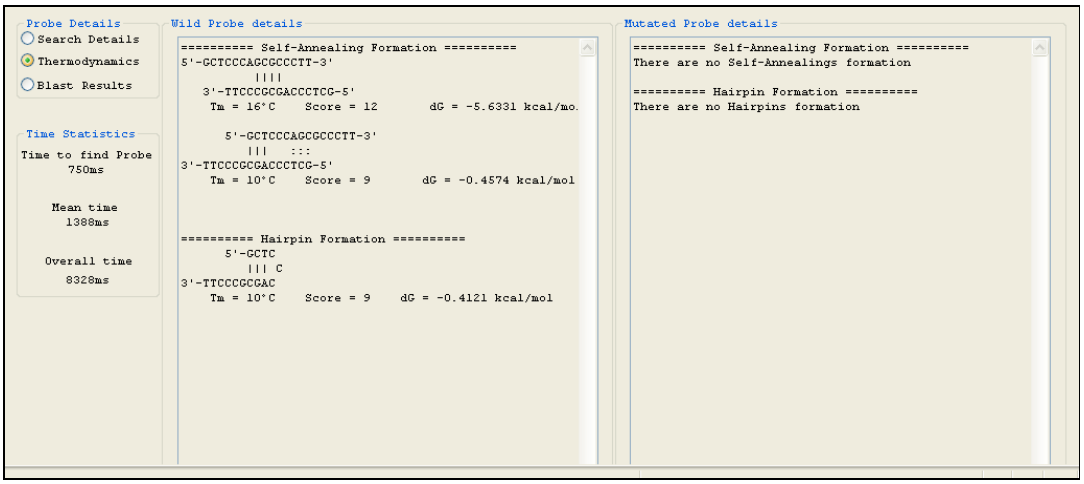


Figure 4.30 - Info View - Thermodynamics: hairpin loops and self-annealing prediction view.

BLAST Results (Figure 4.31**)** – BLAST Results view indicating the possible alignments predicted by the algorithm for each of the selected probes with all genes present in the project. The BLAST algorithm will be discussed further in section 4.4.2.

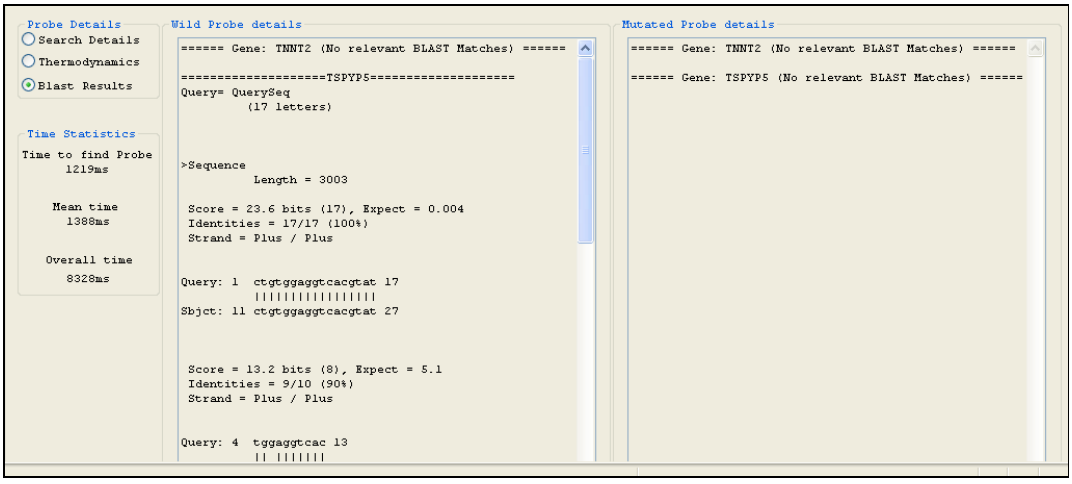


Figure 4.31 - Info view - BLAST Results.

The list of probes successfully designed can be exported to a CSV file which can be opened by Excel or any other program that accepts CSV files as input.

4.4. Algorithms

Probes Design Software has four main algorithms for discussion. The algorithm for probes adjustment is the principal one and was implemented in order to fulfill the specific needs described in section 2.4. This algorithm makes use of three other algorithms publicly available. These algorithms are BLAST [31], which was obtained from NCBI web site, and the algorithms for prediction of secondary structures (hairpin loops and self-annealing formation) which were already described in section 2.3. Although these two last algorithms were previously developed for other programmers, it was not possible to find a source code with the desired results. Therefore, due to the relative simplicity of the algorithms, they were implemented here according to the description made on sections 2.3.4 and 2.3.5.

4.4.1. Probe adjustment

The probe adjustment algorithm was the main and most heavy data processing algorithm developed for this software. It was designed to individually adjust each simple probe.

Simple probes are probes extracted directly from the sequence by considering it centered in the mutation and with length equal to the minimum specified for probes design.

The objective of constructing this algorithm was to adjust every simple probe so that it matches simultaneously the specified parameters of Projects Options which are:

- Nucleotide length of the probes between the specified minimum and maximum.
- Melting Temperature (Tm) of every probe within the specified interval.
- Each probe having an allowed maximum number of sequential repeated nucleotides (Run/Repeat).
- Probes cannot form any secondary structures like hairpins or self-annealing at the specified Room Temperature.
- If required, placement of the probes in a manner that the mutation is located exactly at the center of the probe.
- If specified, the design of extra probes when there is an overlap of the probe with neighbor mutations.
- If specified, the design of a probe as near as possible to the defined parameters, in the case that the user-defined parameters could not be met by the algorithm.

For a better understanding of the algorithm, it will be presented as a flowchart diagram in a high level of abstraction. This diagram (Figure 4.32) resumes ideally the adjustments to be made on each simple probe (Figure 4.9-4) in order to satisfy the specified parameters.

According to the diagram, the main test condition (Figure 4.32-0) checks all the probe parameters and terminates the algorithm with a valid probe found when all the conditions are met. To reach this state, either the simple probe was already inside the defined parameters or it was adjusted during the algorithm run.

The adjustment, if needed, starts by changing the probe position. Since the simple probe has been designed to have the mutation at the center, we can try to slide the probe by one nucleotide to one side maintaining the probe length (Figure 4.32-1) and then test again the properties in order to establish if they are within the parameters. This cycle is repeated until all possible positions of the probe have been tested.

The original simple probe starts with a length equal to the minimum specified. If within all possibilities from the last algorithm cycle a valid probe was not found, the probe length will be increased by one nucleotide (Figure 4.32-2) and again tested against the parameters, starting the algorithm cycle once more.

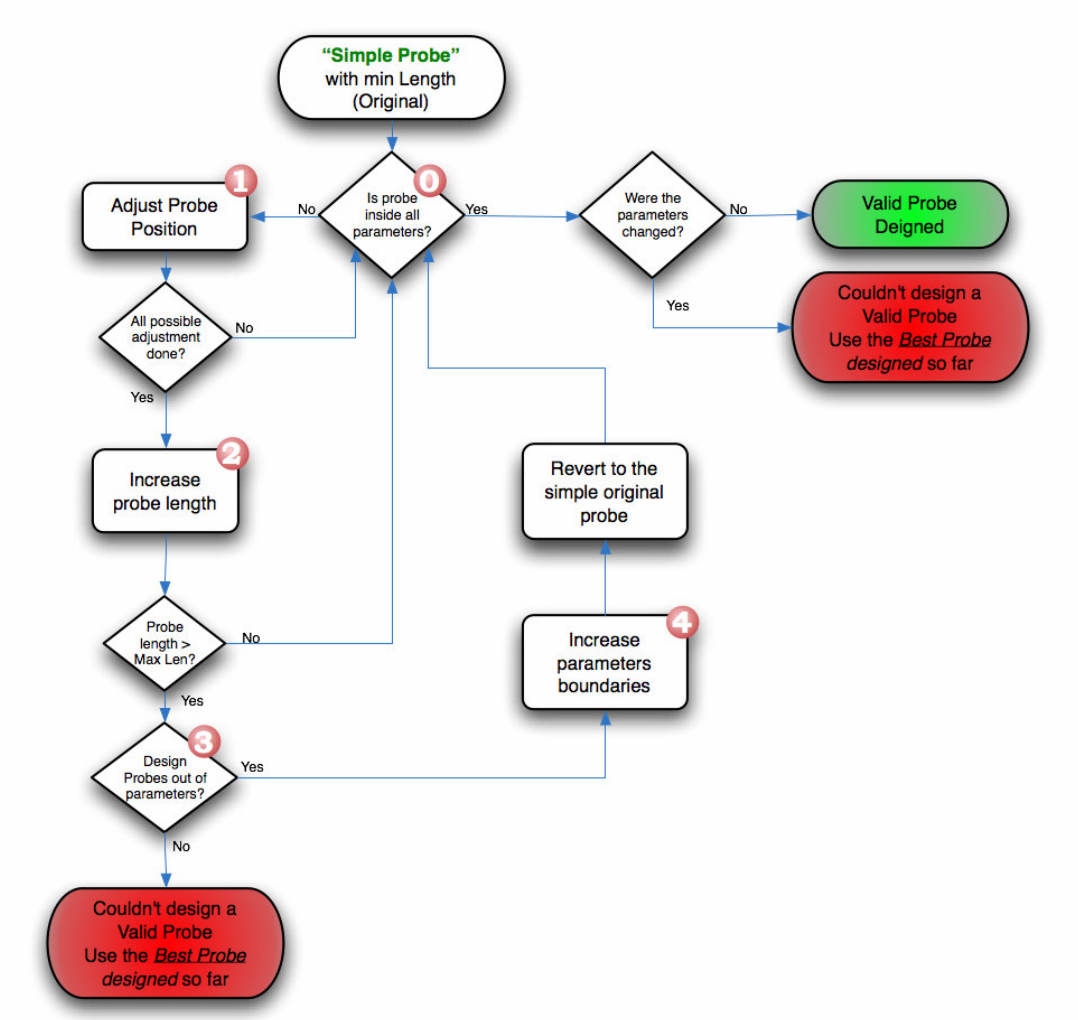


Figure 4.32 - Simple diagram of probes adjustment algorithm.

When the probe length reaches the maximum probe length specified and a valid probe has not been found, all efforts to find a valid probe for the specific mutation were done.

At this time, if the algorithm was not configured to find probes out of defined parameters (Figure 4.32-3), the algorithm stops and presents the best probe found so far, marking it however as "out of parameters".

If defined to design probes out of parameters, the algorithm goes further and increases the parameter boundaries (Figure 4.32-4), reverts to the original simple probe and start all over again. From this point ahead, if the main test conditions are accomplished (Figure 4.32-0), the designed probe will be invariantly classified as "out of parameters" since the original parameters have been changed and all the possible design options were tested.

The diagram in Figure 4.32 is a very simplified view of the probes adjustment algorithm since here are shown only the primary steps in which the algorithm is based. As can be seen from the interpretation of this diagram, the algorithm will not end until it finds a probe that accomplishes the parameters of the main test (Figure 4.32-0). The implemented algorithm is, however, much more complete and assumes a set of limits when designing probes out of the initial user-defined parameters so that the algorithm stops when the length of the probe reaches two times the maximum length specified, or when the ΔTm reaches three times the originally specified value.

In Figure A1.1 (Appendix 1), the reader can find a more detailed diagram of the probes adjustment algorithm.

4.4.2. BLAST

The BLAST algorithm is today the most common choice when sequence alignments are needed. By aligning sequences, it is possible to predict the source of an hybridization between them. For example, if we want to test if the probe with sequence "*5'-AGCGCGATAT-3'*" will hybridize with a specific gene, we run the BLAST algorithm with the reverse sequence probe (*5'-ATATCGCGCT-3'*) against the gene sequence and if we obtain the result of Figure 4.33, we can see that there are no nucleotide mismatches (called gaps), and therefore the probe will theoretically hybridize with the gene.

```
Gene: 5'-…A G C G C G A T A T C G C G C T A T A G G C…-3'
                      | | | | | | | | | |
Probe: 5'- A T A T C G C G C T -3'
```

Figure 4.33 – Ungapped BLAST match sequence.

However, depending on the statistical significance, the BLAST results of Figure 4.34 can also indicate a possible percentage of hybridization between the probe and the gene.

```
Gene: 5'-…A G C G C G A T A T A G T G C T A T A G G C…-3'
                      | | | |   |   | | |
Probe: 5'- A T A T C G C G C T -3'
```

Figure 4.34 - Gapped BLAST match sequence.

The BLAST algorithm originally described by Altschul in 1990 [31] only finds ungapped alignments like the one in Figure 4.33. BLAST was, however, improved and it now assigns statistical significance values to produced alignments using Karlin-Altschul statistics [40] and it was improved to allow gapped alignments (Figure 4.34) that are constructed using a variety of different types of gap costs. By these gap costs, the algorithm generates local alignments that are more accurate and statistically significant [41].

Since BLAST algorithm is freely available from NCBI in various types of applications, it was decided not to implement it but, instead, integrate one executable file for local BLAST that implements the tool "BLAST 2 sequences" available on NCBI web site[12]. This tool produces the alignment of two sequences present in separated FASTA files using BLAST engine for local alignment along with several parameters. This tool is also available in a stand-alone executable file (bl2seq.exe) that can be run from command line with parameters and returns a text file, which allows integrating it in the probe design software.

The executable file "bl2seq.exe" is placed in the same folder as the probe design software executable file, and is called from the design software with the appropriate parameters.

From the long list of arguments accepted by the "bl2seq.exe", we use and specify the following ones:

-i First sequence: Seq.txt

-j Second sequence: Query.txt

-p Program name: blastn

-W Word size: 4

-G Cost to open a gap: 0

[12] http://www.ncbi.nlm.nih.gov/blast/bl2seq/wblast2.cgi

-q Penalty for a nucleotide mismatch (blastn only): -1

-D Output format: 0 - traditional, 1 – tabular: 1

-o alignment output file: Out.txt

The remaining parameters are left by default.

For testing the hybridization of designed probes with undesired genes, the software stores in a file (Seq.txt) the sequence of the gene being analyzed and in another (Query.txt) the sequence of the probe in FASTA format. Then it runs the command: "*bl2seq -i Seq.txt -j Query.txt -p blastn -W 4 -g T -G 0 -q -1 -o Out.txt -D 1*" that outputs the BLAST result in the file "*Out.txt*". On Figure 4.35, there is an output sample of alignment of two sequences with the "*bl2seq.exe*".

```
# BLASTN 2.2.15 [Oct-15-2006]
# Query: QuerySeq
# Fields: Query id, Subject id, % identity, alignment length, mismatches, gap openings, q.
start, q. end, s. start, s. end, e-value, bit score
QuerySeq    Sequence    94.12    17    0    1    1    17    26    41    0.042    20.1
QuerySeq    Sequence    86.67    15    1    1    1    14    1528    1514    1.0    15.5
QuerySeq    Sequence    100.00    9    0    0    7    15    1642    1634    2.3    14.4
QuerySeq    Sequence    100.00    9    0    0    6    14    2851    2859    2.3    14.4
QuerySeq    Sequence    90.00    10    1    0    6    15    70    79    5.1    13.2
QuerySeq    Sequence    100.00    8    0    0    8    15    249    256    5.1    13.2
QuerySeq    Sequence    83.33    12    2    0    2    13    279    290    5.1    13.2
QuerySeq    Sequence    90.00    10    1    0    2    11    342    351    5.1    13.2
QuerySeq    Sequence    78.57    14    3    0    3    16    421    434    5.1    13.2
QuerySeq    Sequence    83.33    12    2    0    4    15    613    602    5.1    13.2
QuerySeq    Sequence    90.00    10    1    0    1    10    1586    1577    5.1    13.2
QuerySeq    Sequence    100.00    8    0    0    7    14    1975    1982    5.1    13.2
```

Figure 4.35 – Sample of "bl2seq.exe" output file.

Once the alignments that were found (one in each row) are ordered by "bit score" value, it is only needed to analyze the first row since if this one corresponds to no hybridization conditions, none of the others will either.

The conditions taken to consider a possible hybridization between a query sequence and a given probe were: a) to have less than two mismatches; b) to have less than two gaps; c) to have a percentage of identity equal or higher than 90%, and d) the length of the alignment as to be at least 90% of the probe length.

4.4.3. Prediction of hairpin loop formation

The prediction of hairpin loop formation is done by sliding the probe with itself (Figure 4.36) calculating for each position the melting temperature (Tm) and the stability of DNA duplex structure (ΔG) of sequential base-pairs bounded groups

with length equal or greater than tree nucleotides (red connections in the Figure 4.36).

Figure 4.36 - Hairpin loop prediction.

The hairpin loop formation is to be considered in a specific position if the temperature of the group with the higher melting temperature calculated is above the Room Temperature specified in the project options as already explained in section 2.3.4.

4.4.4. Prediction of self-annealing formation

The algorithm fused or prediction of self-annealing is similar to the one used for hairpin loop prediction. However, instead of sliding the probe within itself, it is slided with other equal probe, as shown in Figure 4.37.

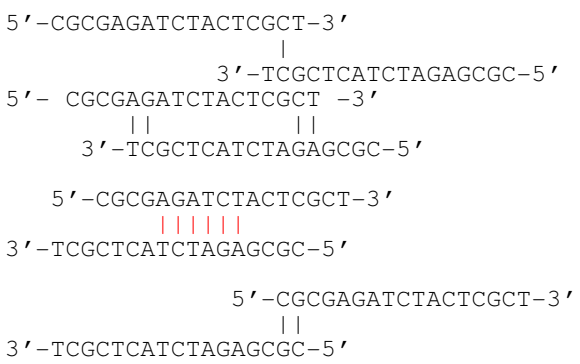


Figure 4.37 - Self-annealing prediction.

Similarly to the hairpin prediction, the self-annealing formation at each nucleotide position of the probe must be considered if the melting temperature of the group with the higher energy of association is above the room temperature specified in the project options (section 2.3.4).

4.5. Development environment

The Probes Design Software was developed in Microsoft Visual Studio 2005 .NET IDE/SDK (Integrated Development Environment/Software Development Kit) using MFC (Microsoft Foundation Classes).
The software and libraries were developed in C++ language with .NET Framework 2.0 (also included in the installation package).
The diagram of Figure 4.38 describes the relation between the main packages of the software, which are:

- Graphical User Interface – This package is responsible for the graphical interface and the control of actions regarding the probes design.
- ProbesDesign – Contains classes with the algorithms responsible for the design of the probes.
- DataBanks – This package contains classes that allow the interface with gene sequences and mutation data.

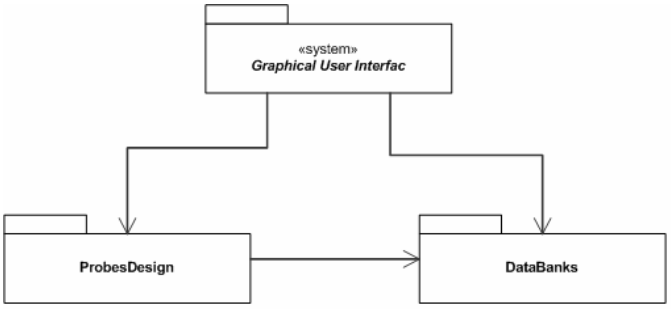


Figure 4.38 – Main packages relationship.

The UML diagrams [42] of Figure 4.39 and Figure 4.40 explain the relationship between the main classes of the probe-design package and the data-banks package, respectively.
The GUI package (Figure 4.39) interacts with the classes of probe design package. The interface with this package is mainly made by the *ProbesDesign* class. This class contains methods to load and save the project data that can be accessed by its class members. It is also responsible for the design of simple probes that become objects of class *Probe* and uses the *GlobalProbes* class for the adjustment of all probes so that they fulfill the design parameters specified on *ProbesOptions* class. The access of this package to the DataBank package allows

for the interfacing with gene sequence files and mutation list files. The mutations are objects of class *Mutation* defined in DataBank package as well.

The package DataBanks (Figure 4.40) contains the class *FastaBank* and *GenBank* which in turns contains the methods for accessing the genome sequence present in FASTA and GenBank formatted files, respectively. The *MutBank* class has the methods for reading and decoding mutations described in HGVS format from "csv" files (Figure 4.4), storing it's decoded data in objects of class *Mutation*. This package also includes the class *Utils* that implements several utility methods for general proposes and is used by almost all classes.

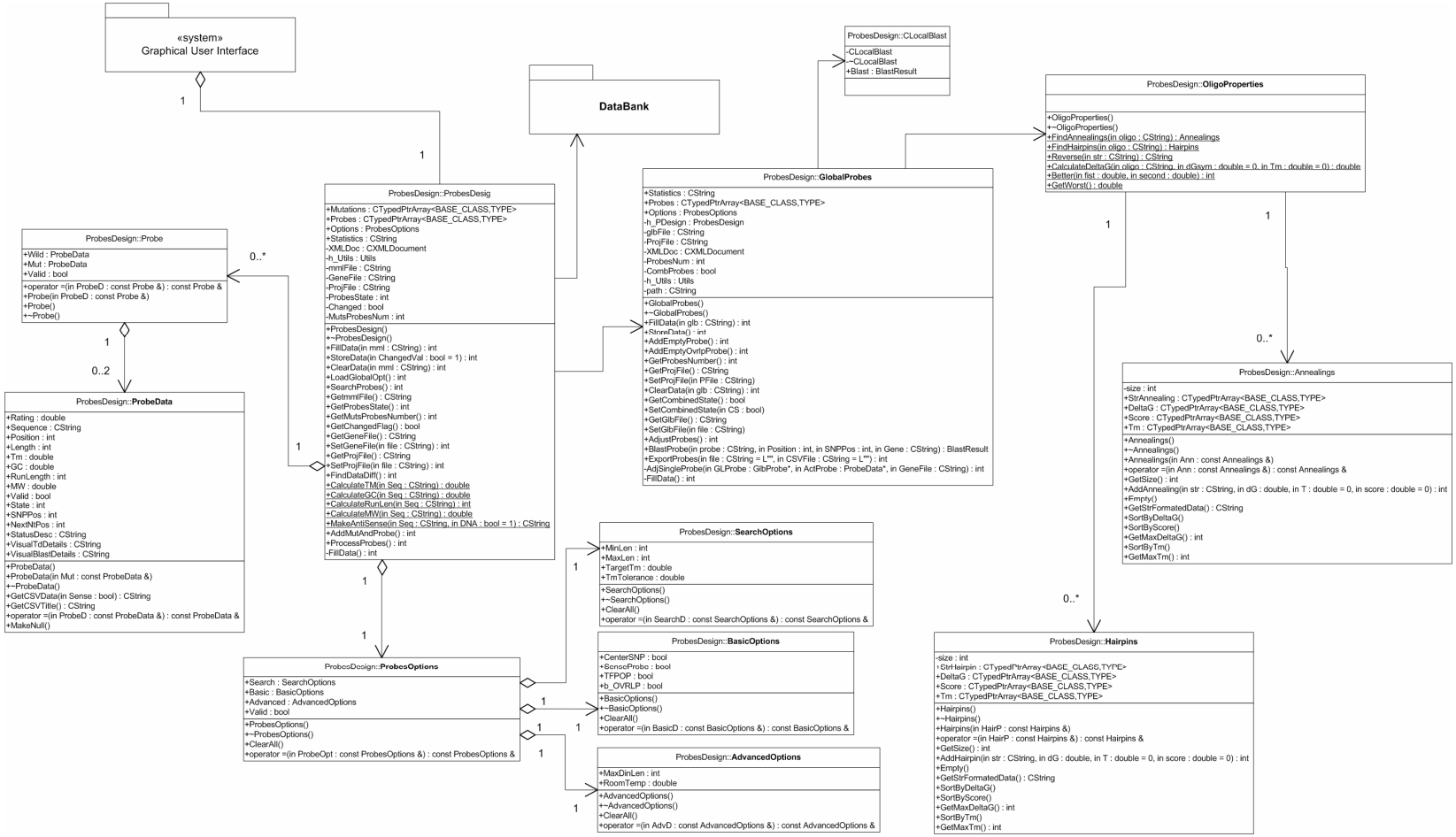


Figure 4.39 – Class diagram and relations of ProbesDesign package.

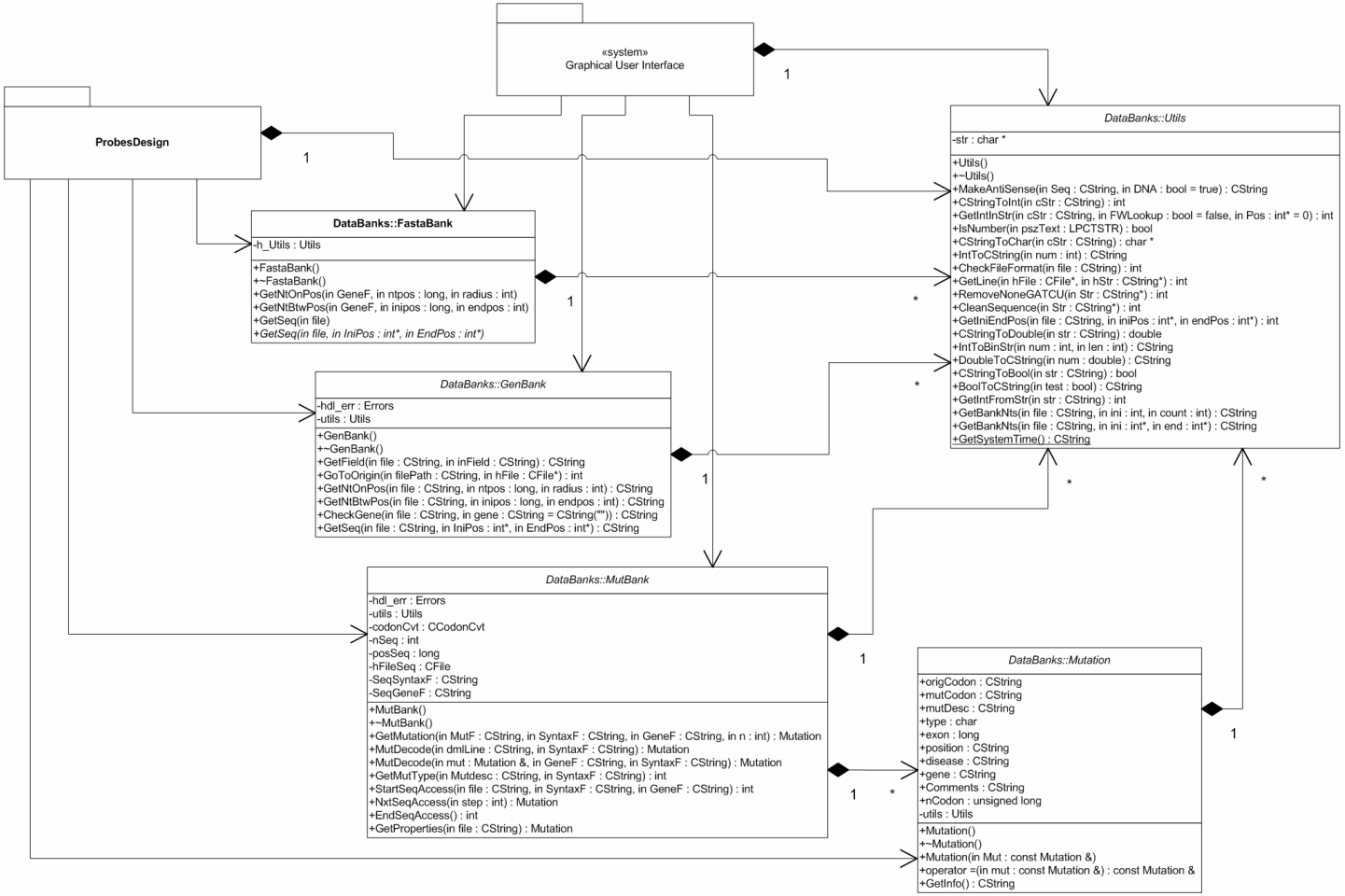


Figure 4.40 - Class diagram and relations of DataBanks package.

4.6. Summary

This chapter includes a description of the implemented software, developed with the objective to fill a gap found in the existing probe-design software for mutation detection.

This software works upon the concept of projects. To each project is associated a set of options and probe parameters that are applied and constrain the designing the probes. Also associated with the project, there are gene sequences and lists of the mutations to be detected in each gene.

The software designs the best suitable probes for detecting the specified mutations, as well as the control probes for detecting the non-mutated sequence.

This chapter also presented the algorithms used in the developed software for BLAST alignment and prediction of secondary structures formation, along with the steps given to design and implement the algorithm used for probe adjustment according to user defined parameters.

The relationships between the software packages themselves as well as between the classes were made clear using UML diagrams.

5. Results

In order to evaluate the features of the developed software, it was tested with some fictitious data that simulated all types of mutations and input data. The software was also tested for some specific scenarios such as overlapped probes, cross-hybridization problems, secondary structures formation, the use of very stringent parameters, wrong input data and so on. The software performed well on these conditions, as expected.

To be able to compare the accuracy of the design of probes with and without using this software, it was tested for a set of mutations in a specific gene, for which probes had already been designed manually.

The set of mutations was chosen from a set of sarcomere protein gene mutations that cause familial hypertrophic cardiomyopathy. The gene in analysis was MYBP-C, which sequence can be downloaded in GENBANK format from NCBI by searching for *locus* HSU91629.

The mutations to be detected in this gene are listed in Table 5.1.

Table 5.1 – Mutations for gene MYBP-C.

Mutation	Exon
g.5137A>C	6
g.5139G>C	6
g.5166G>A	6
g.5190A>G	6
g.5194A>C	6
g.5250G>A	6
g.5254A>C	6
g.5256G>A	6
g.5257G>A	6

The manual process of probes design is typically performed by choosing a specific probe length in order to have a melting temperature around the desired value, and getting the probe sequence to include the mutation locus, while trying to keep the mutation at the centre of the probe sequence.

Table 5.2 is a representation of the procedure taken by biologists to design the probes for the mutations mentioned in Table 5.1. The designed probes will have

as reference a sequence of a certain number of nucleotides (15 in this case) centered on a mutation or located around a group of mutations close together (in Table 5.2, these sequences are marked in yellow and the mutated nucleotide positions in red). For each reference sequence there is the corresponding wild type probe, which corresponds to its anti-sense sequence (the blue lines in Table 5.2), and one probe for detecting each possible arrangement of mutations within the reference sequence. As can be seen, this can easily become a tedious and time consuming task. Besides that, the occurrence of more than one mutation within the same probe region can reduce the success of hybridization, specially if the position of the mutated nucleotides is near the centre of the probe, which determines a higher disturbance on the hybridization dynamics. This may constitute a serious problem for detecting the hybridization signal in experimental work. Also, because the length of probes was not flexible (probes were all designed to have 15 nucleotides), the melting temperature across all probes varied between 46ºC and 54ºC, in a range of as much as 8ºC, which may pose a problem when choosing a temperature to conduct the experiment, since some probes may not hybridize.

By using the developed software, it just takes a few minutes to input the data in the form of one text file containing the gene sequence downloaded from NCBI and one CSV file (created with, for example, Microsoft Excel) containing the list of mutations (as shown in Table 5.1), and then configure the parameters and click on the RUN button. Within a few seconds, the user can obtain a complete set of probes that allows the identification of both the wild type and the mutated sequences, including the extra probes for multiple mutations. Moreover, all relevant properties of each probe, such as Tm, GC base content, molecular weight, length and position in gene sequence are automatically annotated.

In this particular case of probe design, the user might specify the parameters as show in Figure 5.1, allowing the probe length to be adjusted between 15 and 22 nucleotides, and forcing the melting temperature to be between 46ºC and 52ºC (range of 6ºC), designing the probes with the mutations at centre.

Table 5.2 - Manual probes design procedure – Sequences marked in yellow correspond to selected probes. Marked in red are the mutated nucleotide positions. *nt* - probe length, *Tm* - melting temperature, *GC%* - GC base content.

```
5101 ctggtgtccc ctgacgcccc gtccctccat gcacacaagt ctatctgttc gagctgcaca
5161 tcaccgatgc ccagcctgcc ttcactggca gctaccgctg tgaggtgtcc accaaggaca
5221 aatttgactg ctccaacttc aatctcactg tccacggtga gggggccctg gtgtctgtcc
```

Mutation	Nucleotide change	Probe sequence (5'->3')	nt	Tm (°C)	GC %
Wt	Wt: A5137, G5139	GATAGACCTGTGTGC	15	46	53.3
IVS6-2A>C	Mut:A5137C	GATAGACCGGTGTGC	15	48	60.0
Val219Leu	Mut:G5139C	GATAGACCTGTGTGC	15	46	53.3
IVS6-2A>C +Val219Leu	Mut:A5137C +Mut:G5139C	GATAGACCGGTGTGC	15	48	60.0

```
5101 ctggtgtccc ctgacgcccc gtccctccat gcacacaagt ctatctgttc gagctgcaca
5161 tcaccgatgc ccagcctgcc ttcactggca gctaccgctg tgaggtgtcc accaaggaca
5221 aatttgactg ctccaacttc aatctcactg tccacggtga gggggccctg gtgtctgtcc
```

Mutation	Nucleotide change	Probe sequence (5'->3')	nt	Tm (°C)	GC %
Wt	Wt: G5166	TGGGCATCGGTGATG	15	48	60.0
Asp228Asn	Mut:G5166A	TGGGCATTGGTGATG	15	46	53.3

```
5101 ctggtgtccc ctgacgcccc gtccctccat gcacacaagt ctatctgttc gagctgcaca
5161 tcaccgatgc ccagcctgcc ttcactggca gctaccgctg tgaggtgtcc accaaggaca
5221 aatttgactg ctccaacttc aatctcactg tccacggtga gggggccctg gtgtctgtcc
```

Mutation	Nucleotide change	Probe sequence (5'->3')	nt	Tm (°C)	GC %
Wt	Wt: A5190, A5194	AGCGGTAGCTGCCAG	15	50	66.7
(AGC)Ser236Gly(GGC)	Mut:A5190G	AGCGGTAGCCGCCAG	15	52	73.3
Tyr237Ser	Mut:A5194C	AGCGGGAGCTGCCAG	15	52	73.3
(AGC)Ser236Gly(GGC) +Tyr237Ser	Mut:A5190G +Mut:A5194C	AGCGGGAGCCGCCAG	15	54	80.0

```
5101 ctggtgtccc ctgacgcccc gtccctccat gcacacaagt ctatctgttc gagctgcaca
5161 tcaccgatgc ccagcctgcc ttcactggca gctaccgctg tgaggtgtcc accaaggaca
5221 aatttgactg ctccaacttc aatctcactg tccacggtga gggggccctg gtgtctgtcc
```

Mutation	Nucleotide change	Probe sequence (5'->3')	nt	Tm (°C)	GC %
Wt	Wt: G5250, A5254	CACCGTGGACAGTGA	15	48	60.0
Val256Ile	Mut:G5250A	CACCGTGGAIAGTGA	15	46	53.3
His257Pro	Mut:A5254C	CACCGGGGACAGTGA	15	50	66.7
Val256Ile +His257Pro	Mut:G5250A +Mut:A5254C	CACCGGGGAIAGTGA	15	48	60.0
Glu258Lys	Mut:G5256A	CACTGTGGACAGTGA	15	46	53.3
Glu258Lys + Val256Ile	Mut:G5256A +Mut:G5250A	CACTGTGGAIAGTGA	15	44	46.7
Glu258Lys +His257Pro	Mut:G5256A +Mut:A5254C	CACTGGGGACAGTGA	15	48	60.0
Glu258Lys +His257Pro +Val256Ile	Mut:G5256A +Mut:A5254C +Mut:G5250A	CACTGGGGAIAGTGA	15	46	53.3
IVS6+1G>A	Mut:G5257A	CAICGTGGACAGTGA	15	46	53.3
IVS6+1G>A +Glu258Lys	Mut:G5257A +Mut:G5256A	CAITGTGGACAGTGA	15	44	46.7
IVS6+1G>A +His257Pro	Mut:G5257A +Mut:A5254C	CAICGGGGACAGTGA	15	48	60.0
IVS6+1G>A +Val256Ile	Mut:G5257A +Mut:G5250A	CAICGTGGAIAGTGA	15	44	46.7
IVS6+1G>A +Glu258Lys +His257Pro	Mut:G5257A +Mut:G5256A +Mut:A5254C	CAITGGGGACAGTGA	15	46	53.3
IVS6+1G>A +Glu258Lys +Val256Ile	Mut:G5257A +Mut:G5256A +Mut:G5250A	CAITGTGGAIAGTGA	15	40	42.0
IVS6+1G>A +Glu258Lys +His257Pro +Val256Ile	Mut:G5257A +Mut:G5256A +Mut:A5254C +Mut:G5250A	CAITGGGGAIAGTGA	15	44	46.7

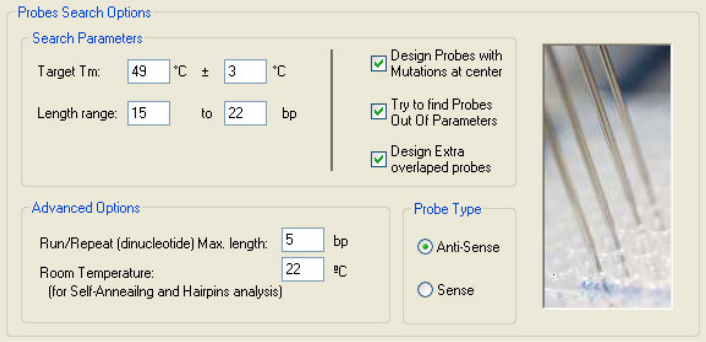


Figure 5.1 - Project options for probe design in a case study presented in the text.

As can be seen on the probe list of Table A2.1 (Appendix2), the software successfully designed a full set of probes for wild type and mutated sequences), considering all possible arrangements for probes overlapping neighbor mutations. The automated design of probes allows a better accuracy in Tm matching and in centering the mutations on the probe sequences.

In order to better understand the difference between the probes designed by the software and the ones designed in a non automated way, we analyzed both methods of probe design for two mutations where each probe overlaps neighbor mutations.

In Figure 5.2, there is part of a gene sequence containing two nearby mutations used for comparing the results using an automated and non automated approach for probe design. The nucleotides highlighted in red are the ones that will be tested for mutation. The mutations for which the probes will be designed will be the "*A5190G*", where the nucleotide on position 5190 is mutated from an "A" to a "G", and the "*A5194C*", where the nucleotide on position 5194 is mutated from an "A" to a "C". We have to take into account that the designed probes must be anti-sense, and therefore the mutations found in the probes will correspond respectively to a "T" mutated to a "C", and a "T" mutated to a "G".

```
5101 ctggtgtccc ctgacgcccc gtccctccat gcacacaggt ctatctgttc gagctgcaca
5161 tcaccgatgc ccagcctgcc ttcactggca gctaccgctg tgaggtgtcc accaaggaca
5221 aatttgactg ctccaacttc aatctcactg tccacggtga gggggccctg gtgtctgtcc
```

Figure 5.2 - Sequence used for comparing results of automated and non-automated probe design.

Analyzing the probes designed by hand, in Table 5.3 (the column named "*Probes designed by hand*") we can see that four probes can be designed. As we have two mutations near to each other, four probes ($2^2 = 4$) would be sufficient to cover all arrangements, one being the probe to detect the wild type sequence (probe number 1 of Table 5.3), two for detecting the mutations individually (probes number 2 and 6 of Table 5.3) and one more to detect the presence of both mutations simultaneously (probe number 4 of Table 5.3).

Analyzing now the probes designed by the software (the column named "*Probes designed by the software*" of Table 5.3) we can see that more probes than supposed enough were designed. This happens because the software will design all possible arrangements but this time with one more constraint: the mutation in analysis must be at the centre of the probe. Additionally, the software will also design extra "wild" probes. We call "wild" (between double quotes) to a sequence that does not have the mutation in analysis, but it has any other predicted neighboring mutation. In this way we will have for '*n*' mutations close together, which the respective probes will overlap, as many probes as calculated with the Formula 5.1.

$$\langle \text{Number of Probes} \rangle = 2^n \times n$$

Formula 5.1 - Number of probes for '*n*' mutations close together.

It may seam that probes will be repeated, like the probes number 4 and 8 that have the same mutated bases, but actually, each one is centered to each mutation. The same happens between the probes 1 and 5 and between the probes 2 and 7.

The redundancy created by designing different probes for detecting the same mutations will allow higher accuracy when interpreting the results from the hybridization, since we will have probes that will work as controls for each possible arrangement of mutations.

On Table A2.2, there is a list of the probes resulting from running the software with the same data and parameters used for creating the list of Table A2.1, except that now the Tm was set to be 50ºC±2ºC (range of 4ºC). This adjustment in parameters is expected to produce probes with even better melting temperature homology. However, this restriction on Tm parameters produced three probes with mutations slightly deviated from the centre (probes 8.1, 8.4

and 9.5 in Table A2.1) which could be interpreted as a less good probe design strategy. Nevertheless, since the deviation is of only one bp (base pair) from the centre, it will probably not cause a problem for hybridization dynamics. The absence of complementarity at this less central position is still enough perturbing to the stability of the hybridization so that non-specific hybrids may not be formed at the optimized hybridization temperature.

Table 5.3 - Comparison between probes designed by the software and by hand.

Mutation in Analysis	Num	Nucleotide Change	Probes designed by the software			Probes designed by hand		
			Sequence (5'->3')	Length	Tm	Sequence (5'->3')	Length	Tm
Ser236Gly (A5190G)	1	Wild	CGGTAGCTGCCAGTG	15	50.00	AGCGGTAGCTGCCAG	15	50
	2	Mut:A5190G	CGGTAGCCGCCAGTG	15	52.00	AGCGGTAGCCGCCAG	15	52
	3	"Wild" with mutation A5194C	CGGGAGCTGCCAGTG	15	48.00			
	4	Mut:A5190G +Mut:A5194C	CGGGAGCCGCCAGTG	15	46.00	AGCGGCAGCCGCCAG	15	54
Tyr237Ser (A5194C)	5	Wild	ACAGCGGTAGCTGCC	15	50.00			
	6	Mut:A5194C	ACAGCGGCAGCTGCC	15	52.00	AGCGGCAGCTGCCAG	15	52
	7	"Wild" with mutation A5190G	ACAGCGGTAGCCGCC	15	48.00			
	8	Mut:A5194C + Mut:A5190G	CACAGCGGCAGCCGCCA	17	52.00			

Legend: Nucleotides in red: mutation where the probe is centered; Nucleotides in yellow: mutation also accounted in the probe sequence; Nucleotides in grey: place of other mutations not considered for probe design.

6. Conclusions and future work

To be able to evaluate the existing software involving the design of probes for mutations detection using microarrays, as well as to conclude about the missing functionalities, it was required the understanding of the biological and technical concepts behind the microarrays technology as well as the requirements by the biologists about this matter.

Some bioinformatics tools were tested and evaluated for this specific task. After this analysis, it was evident the lack of tools that could perform this specific task according to the particular requirements.

In order to overcome this need, a novel software tool was developed, taking into care the accomplishment of all user requirements. For that, a set of algorithms have been studied, as was the case of BLAST and the prediction of secondary structures formation. Moreover, an algorithm for the adjustment of simple probes for mutations detection was idealized and implemented. The standard recommended nomenclature for description of sequence variants described by the Human Genome Variation Society (HGVS) was used throughout the software development, for standardization purposes.

In respect to the software architecture, the development with Microsoft operative systems was preferred to others due to the fact that the majority of the end users would be accustomed to use this system, and also due to the simplicity and experience in programming with the Microsoft .NET Framework. Some new knowledge had to be acquired for using MFC libraries, regular expressions and RTF syntax.

After analyzing the results reached by using the developed software to design sets of probes for detecting mutations, it can be concluded that the developed software reduces significantly the time spent designing probes and does it more efficiently and rigorously as well.

Regarding the resultant designed probes, it can be said that, they are more specific for each mutation, since more probes are designed to detect all possible arrangements not only of mutated sequences, but also for all possible “wild” sequences in respect to the possible neighboring mutations.

The hybridization success tends to be higher given that the algorithm always tries to place the probes with the respective mutations as centered as possible

and designs them with the main objective to match the desired melting temperature interval, increasing the Tm homology.

The software also verifies the possibility of secondary structures formation and undesired hybridization of probes with other genes included in the project or with any part of the gene other than the one it was designed for, contributing in this way also to the success of the hybridization experiment.

Given all these aspects together, it can be said that scientists will probably get higher accuracy probes by using the ones designed by this software to detect mutations using microarrays.

Nevertheless, there are some aspects that can be improved in the future, as the already mentioned performance in the probes adjustment algorithm and some other features dealing with data, such as printing options, text search and alternative presentation of the results. Future developments should also consider the possibility to view overlapped probes in the graphical interface.

The software would also benefit from options such as retrieving the gene sequences directly from NCBI given, for example, the accession number, and the possibility to design primers and other types of probes, useful in other experimental applications.

The hybridization success rate of the probes designed by this software could be in practice evaluated by actually manufacturing the probes suggested and the ones designed by hand and then comparing the hybridization results.

References

[1] G. Mendel, "Experiments in Plant Hybridization," *Verh Naturf Ver Brunn*, vol. 4, pp. 1-47, 1865.

[2] A. Griffiths, J. Miller, D. Suzuki, R. Lewontin, and WMGelbart, *An Introduction to Genetic Analysis*: W.H. Freeman and Company, 2000.

[3] R. B. Stoughton, "Applications of DNA microarrays in biology," *Annu Rev Biochem*, vol. 74, pp. 53-82, 2005.

[4] A. Jung, "DNA chip technology," *Anal Bioanal Chem*, vol. 372, pp. 41-2, 2002.

[5] M. Cuzin, "DNA chips: a new tool for genetic analysis and diagnostics," *Transfus Clin Biol*, vol. 8, pp. 291-6, 2001.

[6] H. B. Nielsen, R. Wernersson, and S. Knudsen, "Design of oligonucleotides for microarrays and perspectives for design of multi-transcriptome arrays," *Nucleic Acids Res*, vol. 31, pp. 3491-6, 2003.

[7] F. Li and G. D. Stormo, "Selection of optimal DNA oligos for gene expression arrays," *Bioinformatics*, vol. 17, pp. 1067-76, 2001.

[8] I. Lee, A. A. Dombkowski, and B. D. Athey, "Guidelines for incorporating non-perfectly matched oligonucleotides into target-specific hybridization probes for a DNA microarray," *Nucleic Acids Res*, vol. 32, pp. 681-90, 2004.

[9] H. Lodish, A. Berk, L. S. Zipursky, P. Matsudaira, D. Baltimore, and J. Darnell, *Molecular Cell Biology*, 4 ed: W. H. Freeman and Company., 2000.

[10] R. Twyman, "*Mutation or polymorphism?,*" Mutation or polymorphism?, 2003. [Online]. Available: http://genome.wellcome.ac.uk/doc_WTD020780.html. [Accessed: Sep. 16, 2007].

[11] H. G. V. Society, "*Nomenclature for the description of sequence variants,*" Nomenclature for the description of sequence variants; Discussions, 2005. [Online]. Available: http://www.hgvs.org/mutnomen/disc.html. [Accessed: Sep. 16, 2007].

[12] M. J. Heller, "DNA microarray technology: devices, systems, and applications," *Annu Rev Biomed Eng*, vol. 4, pp. 129-53, 2002.

[13] M. C. Pirrung, "How to Make a DNA Chip," *ANGEWANDTE CHEMIE*, vol. 41, pp. 1276-89, 2002.

[14] K. K. Wilgenbus and P. Lichter, "DNA chip technology ante portas," *J Mol Med*, vol. 77, pp. 761-8, 1999.

[15] D. Shalon, S. J. Smith, and P. O. Brown, "A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization," *Genome Res*, vol. 6, pp. 639-45, 1996.

[16] A. Relogio, C. Schwager, A. Richter, W. Ansorge, and J. Valcarcel, "Optimization of oligonucleotide-based DNA microarrays," *Nucleic Acids Res*, vol. 30, pp. e51, 2002.

[17] J. Letowski, R. Brousseau, and L. Masson, "Designing better probes: effect of probe size, mismatch position and number on hybridization in DNA oligonucleotide microarrays," *J Microbiol Methods*, vol. 57, pp. 269-78, 2004.

[18] C. C. Chou, C. H. Chen, T. T. Lee, and K. Peck, "Optimization of probe length and the number of probes per gene for optimal microarray analysis of gene expression," *Nucleic Acids Res*, vol. 32, pp. e99, 2004.

[19] M. W. Karaman, S. Groshen, C. C. Lee, B. L. Pike, and J. G. Hacia, "Comparisons of substitution, insertion and deletion probes for resequencing and mutational analysis using oligonucleotide microarrays," *Nucleic Acids Res*, vol. 33, pp. e33, 2005.

[20] K. L. Gunderson, X. C. Huang, M. S. Morris, R. J. Lipshutz, D. J. Lockhart, and M. S. Chee, "Mutation detection by ligation to complete n-mer DNA arrays," *Genome Res*, vol. 8, pp. 1142-53, 1998.

[21] E. F. Nuwaysir, W. Huang, T. J. Albert, J. Singh, K. Nuwaysir, A. Pitas, T. Richmond, T. Gorski, J. P. Berg, J. Ballin, M. McCormick, J. Norton, T. Pollock, T. Sumwalt, L. Butcher, D. Porter, M. Molla, C. Hall, F. Blattner, M. R. Sussman, R. L. Wallace, F. Cerrina, and R. D. Green, "Gene expression analysis using oligonucleotide arrays produced by maskless photolithography," *Genome Res*, vol. 12, pp. 1749-55, 2002.

[22] A. Panjkovich and F. Melo, "Comparison of different melting temperature calculation methods for short DNA sequences," *BIOINFORMATICS*, vol. 21, pp. 711-722, 2005.

[23] J. Marmur and P. Doty, "Determination of the base composition of deoxyribonucleic acid from its thermal denaturation temperature," *J Mol Biol*, vol. 5, pp. 109-18, 1962.

[24] P. M. Howley, M. A. Israel, M. F. Law, and M. A. Martin, "A rapid method for detecting and mapping homology between heterologous DNAs. Evaluation of polyomavirus genomes," *J Biol Chem*, vol. 254, pp. 4876-83, 1979.

[25] D. G. Wilkinson, *In Situ Hybridization A Practical Approach*: IRLPRESS: Oxford University Press, 1992.

[26] K. J. Breslauer, L. A. Marky, R. Frank, and H. Blöcker, "Predicting DNA duplex stability from the base sequence," *Proc Natl Acad Sci U S A*, vol. 83, pp. 3746-50, 1986.

[27] S. A. Kushon, J. P. Jordan, J. L. Seifert, H. Nielsen, P. E. Nielsen, and B. A. Armitage, "Effect of secondary structure on the thermodynamics and kinetics of PNA hybridization to DNA hairpins," *J Am Chem Soc*, vol. 123, pp. 10805-13, 2001.

[28] P. M. Vallone, T. M. Paner, J. Hilario, M. J. Lane, B. D. Faldasz, and A. S. Benight, "Melting studies of short DNA hairpins: influence of loop sequence and adjoining base pair identity on hairpin thermodynamic stability," *Biopolymers*, vol. 50, pp. 425-42, 1999.

[29] H. Hyyro, M. Juhola, and M. Vihinen, "Genome-wide selection of unique and valid oligonucleotides," *Nucleic Acids Res*, vol. 33, pp. e115, 2005.

[30] "*Generation of Template DNA,*" Brain Gene Expression Map (stjudebgem.org), 2004. [Online]. Available: http://www.stjudebgem.org/web/html/Generation.php. [Accessed: Sep. 16, 2007].

[31] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J Mol Biol*, vol. 215, pp. 403-10, 1990.

[32] J. Lee and X. Nien-Lin, "Analyzing user requirements by use cases: a goal-driven approach," *Software, IEEE*, vol. 16, pp. 92-101, 1999.

[33] N. Maiden and S. Robertson, "Developing use cases and scenarios in the requirements process," presented at Software Engineering, 2005. ICSE 2005. Proceedings. 27th International Conference on, 2005.

[34] "*FASTA format description,*" FASTA format description. [Online]. Available: http://www.ncbi.nlm.nih.gov/blast/fasta.shtml. [Accessed: Sep. 16, 2007].

[35] "*GenBank Flat File Format,*" GenBank Sample Record, 2006. [Online]. Available: http://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html. [Accessed: Sep. 16, 2007].

[36] R. Wernersson and H. B. Nielsen, "OligoWiz 2.0--integrating sequence feature annotation into the design of microarray probes," *Nucleic Acids Res*, vol. 33, pp. W611-5, 2005.

[37] K. E. Herold and A. Rasooly, "Oligo Design: a computer program for development of probes for oligonucleotide microarrays," *Biotechniques*, vol. 35, pp. 1216-21, 2003.

[38] N. Reymond, H. Charles, L. Duret, F. Calevro, G. Beslon, and J. M. Fayard, "ROSO: optimizing oligonucleotide probes for microarrays," *Bioinformatics*, vol. 20, pp. 271-3, 2004.

[39] H. H. Chou, A. P. Hsia, D. L. Mooney, and P. S. Schnable, "Picky: oligo microarray design for large genomes," *Bioinformatics*, vol. 20, pp. 2893-902, 2004.

[40] S. Karlin and S. F. Altschul, "Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes," *Proc Natl Acad Sci U S A*, vol. 87, pp. 2264-8, 1990.

[41] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Res*, vol. 25, pp. 3389-402, 1997.

[42] M. Szlenk, "Formal Semantics and Reasoning about UML Class Diagram," presented at Dependability of Computer Systems, 2006. DepCos-RELCOMEX '06. International Conference on, 2006.

Appendix 1

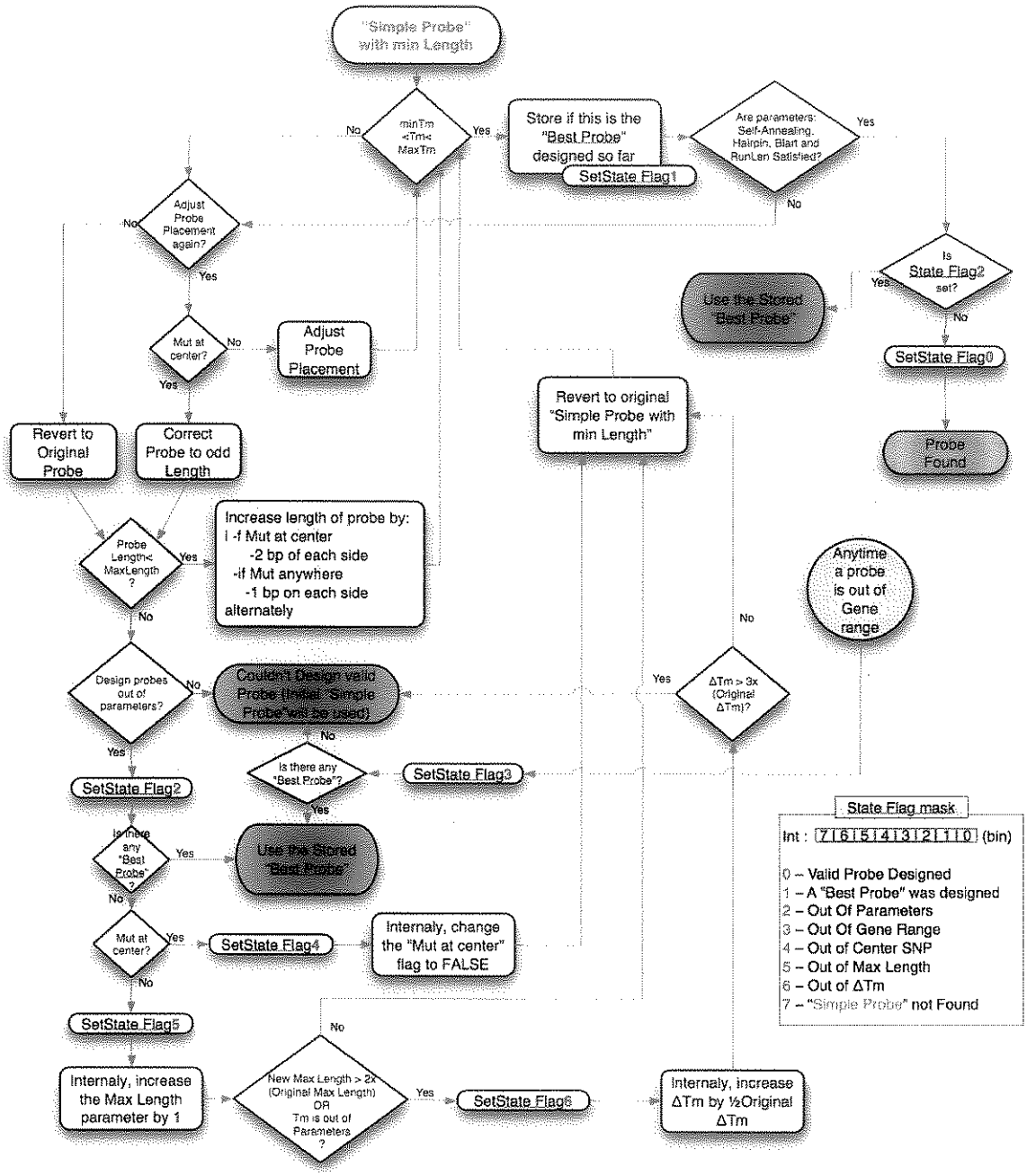


Figure A1.1Implementation of probes adjustment - Detailed diagram.

Appendix 2

Table A2.1 – Probes for mutations listed on Table 5.1 – Mutations for gene MYBP-C with Tm = 49ºC ±3ºC.

Gene	Num	State	Sequence	Position	Length	Tm	GC	RunLength	MW	Mutation
MYBPC3	1	Wild	GATAGACCTGTGTGCAT	5129	17	50.00	47.59	2	5154.38	g.5137a>c
		Mutated	ATAGACCGGTGTGCA	5130	15	46.00	53.33	2	4527.98	
	1.1	Wild	GATAGAGCTGTGTGCAT	5129	17	46.00	41.18	1	5114.36	g.5137a>c
		Mutated	AGATAGAGCGGTGTGCATG	5128	19	50.00	42.15	1	5683.74	
	2	Wild	CAGATAGACCTGTGTGC	5131	17	52.00	52.94	2	5170.39	g.5139g>c
		Mutated	CAGATAGAGCTGTGTGC	5131	17	52.00	52.94	1	5130.37	
	2.1	Wild	CAGATAGACCGGTGTGC	5131	17	50.00	52.94	2	5146.38	g.5139g>c
		Mutated	CAGATAGAGCGGTGTGC	5131	17	46.00	47.59	1	5106.36	
	3	Wild	TGGGCATCGGTGATG	5159	15	48.00	60.00	3	4481.95	g.5166g>a
		Mutated	TGGGCATTGGTGATG	5159	15	46.00	53.33	3	4465.94	
	4	Wild	CGGTAGCTGCCAGTG	5183	15	50.00	66.67	2	4537.98	g.5190a>g
		Mutated	CGGTAGCCGCCAGTG	5183	15	52.00	73.33	2	4553.99	
	4.1	Wild	CGGGAGCTGCCAGTG	5183	15	48.00	66.67	2	4513.97	g.5190a>g
		Mutated	CGGGAGCCGCCAGTG	5183	15	46.00	66.67	2	4529.98	
	5	Wild	ACAGCGGTAGCTGCC	5187	15	50.00	66.67	2	4569.00	g.5194a>c
		Mutated	ACAGCGGGAGCTGCC	5187	15	52.00	73.33	3	4544.99	
	5.1	Wild	ACAGCGGTAGCCGCC	5187	15	48.00	66.67	2	4585.99	g.5194a>c
		Mutated	CACAGCGGGAGCCGCCA	5186	17	52.00	64.76	2	5194.50	
	6	Wild	CCGTGGACAGTGAGA	5243	15	48.00	60.00	2	4503.97	g.5250g>a
		Mutated	CCGTGGATAGTGAGA	5243	15	46.00	53.33	2	4487.96	
	6.1	Wild	ATCGTGGACAGTGAGAT	5242	17	48.00	47.59	2	5105.36	g.5250g>a
		Mutated	CATCGTGGATAGTGAGATT	5241	19	50.00	42.15	2	5731.76	
	6.2	Wild	ACTGTGGACAGTGAGAT	5242	17	48.00	47.59	2	5105.36	g.5250g>a
		Mutated	CACTGTGGATAGTGAGATT	5241	19	50.00	42.15	2	5731.76	
	6.3	Wild	CATTGTGGACAGTGAGATT	5241	19	50.00	42.15	2	5731.76	g.5250g>a
		Mutated	CATTGTGGATAGTGAGATT	5241	19	46.00	36.84	2	5715.75	
	6.4	Wild	CCGGGGACAGTGAGA	5243	15	46.00	60.00	2	4479.96	g.5250g>a
		Mutated	ACCGGGGATAGTGAGAT	5242	17	46.00	47.59	2	5081.35	
	6.5	Wild	ATCGGGGACAGTGAGAT	5242	17	46.00	47.59	2	5081.35	g.5250g>a
		Mutated	CATCGGGGATAGTGAGATT	5241	19	48.00	42.15	2	5707.75	
	6.6	Wild	ACTGGGGACAGTGAGAT	5242	17	46.00	47.59	2	5081.35	g.5250g>a
		Mutated	CACTGGGGATAGTGAGATT	5241	19	48.00	42.15	2	5707.75	
	6.7	Wild	CATTGGGGACAGTGAGATT	5241	19	48.00	42.15	2	5707.75	g.5250g>a
		Mutated	TCATTGGGGATAGTGAGATTG	5240	21	50.00	38.95	2	6294.13	
	7	Wild	CTCACCGTGGACAGT	5247	15	48.00	60.00	2	4593.10	g.5254a>c
		Mutated	CTCACCGGGGACAGT	5247	15	50.00	66.67	4	4569.00	
	7.1	Wild	CCTCATCGTGGACAGTG	5246	17	52.00	58.82	2	5195.40	g.5254a>c
		Mutated	CCTCATCGGGGACAGTG	5246	17	50.00	58.82	2	5171.39	
	7.2	Wild	CCTCACTGTGGACAGTG	5246	17	52.00	58.82	2	5195.40	g.5254a>c
		Mutated	CCTCACTGGGGACAGTG	5246	17	50.00	58.82	2	5171.39	
	7.3	Wild	CCTCATTGTGGACAGTG	5246	17	48.00	52.94	2	5179.39	g.5254a>c
		Mutated	CCTCATTGGGGACAGTG	5246	17	46.00	52.94	2	5155.38	
	7.4	Wild	CCTCACCGTGGATAGTG	5246	17	52.00	58.82	2	5195.40	g.5254a>c
		Mutated	CCTCACCGGGGATAGTG	5246	17	50.00	58.82	2	5171.39	
	7.5	Wild	CCTCATCGTGGATAGTG	5246	17	48.00	52.94	2	5179.39	g.5254a>c

		Mutated	CCTCATCGGGGATAGTG	5246	17	46.00	52.94	2	5155.38	
	7.6	Wild	CCTCACTGTGGATAGTG	5246	17	48.00	52.94	2	5179.39	g.5254a>c
		Mutated	CCTCACTGGGGATAGTG	5246	17	46.00	52.94	2	5155.38	
	7.7	Wild	CCCTCATTGTGGATAGTGA	5245	19	50.00	47.37	3	5796.79	g.5254a>c
		Mutated	CCCTCATTGGGGATAGTGA	5245	19	48.00	47.37	3	5772.78	
	8	Wild	CCCTCACCGTGGACA	5249	15	50.00	66.67	3	4649.40	g.5256g>a
		Mutated	CCCTCACTGTGGACA	5249	15	48.00	60.00	3	4633.30	
	8.1	Wild	CCCTCATCGTGGACA	5249	15	46.00	60.00	3	4633.30	g.5256g>a
		Mutated	CCCCTCATTGTGGACAG	5248	17	50.00	58.82	4	5235.42	
	8.2	Wild	CCCTCACCGGGGACA	5249	15	48.00	66.67	3	4625.30	g.5256g>a
		Mutated	CCCCTCACTGGGGACAG	5248	17	52.00	64.76	4	5227.42	
	8.3	Wild	CCCCTCATCGGGGACAG	5248	17	52.00	64.76	4	5227.42	g.5256g>a
		Mutated	CCCCTCATTGGGGACAG	5248	17	48.00	58.82	4	5211.50	
	8.4	Wild	CCCTCACCGTGGATA	5249	15	46.00	60.00	3	4633.30	g.5256g>a
		Mutated	CCCCTCACTGTGGATAG	5248	17	50.00	58.82	4	5235.42	
	8.5	Wild	CCCCTCATCGTGGATAG	5248	17	50.00	58.82	4	5235.42	g.5256g>a
		Mutated	CCCCTCATTGTGGATAG	5248	17	46.00	52.94	4	5219.50	
	8.6	Wild	CCCCTCACCGGGGATAG	5248	17	52.00	64.76	4	5227.42	g.5256g>a
		Mutated	CCCCTCACTGGGGATAG	5248	17	48.00	58.82	4	5211.50	
	8.7	Wild	CCCCTCATCGGGGATAG	5248	17	48.00	58.82	4	5211.50	g.5256g>a
		Mutated	CCCCCTCATTGGGGATAGT	5247	19	50.00	52.63	5	5837.90	
	9	Wild	CCCCTCACCGTGGAC	5250	15	52.00	73.33	4	4674.50	g.5257g>a
		Mutated	CCCCTCATCGTGGAC	5250	15	50.00	66.67	4	4658.40	
	9.1	Wild	CCCCTCACTGTGGAC	5250	15	48.00	66.67	4	4658.40	g.5257g>a
		Mutated	CCCCCTCATTGTGGACA	5249	17	50.00	58.82	5	5275.44	
	9.2	Wild	CCCCTCACCGGGGAC	5250	15	50.00	73.33	4	4650.40	g.5257g>a
		Mutated	CCCCTCATCGGGGAC	5250	15	46.00	66.67	4	4634.30	
	9.3	Wild	CCCCTCACTGGGGAC	5250	15	46.00	66.67	4	4634.30	g.5257g>a
		Mutated	CCCCCTCATTGGGGACA	5249	17	48.00	58.82	5	5251.43	
	9.4	Wild	CCCCTCACCGTGGAT	5250	15	48.00	66.67	4	4658.40	g.5257g>a
		Mutated	CCCCCTCATCGTGGATA	5249	17	50.00	58.82	5	5275.44	
	9.5	Wild	CCCCCTCACTGTGGATA	5249	17	50.00	58.82	5	5275.44	g.5257g>a
		Mutated	CCCCCTCATTGTGGATA	5249	17	46.00	52.94	5	5259.43	
	9.6	Wild	CCCCTCACCGGGGAT	5250	15	46.00	66.67	4	4634.30	g.5257g>a
		Mutated	CCCCCTCATCGGGGATA	5249	17	48.00	58.82	5	5251.43	
	9.7	Wild	CCCCCTCACTGGGGATA	5249	17	48.00	58.82	5	5251.43	g.5257g>a
		Mutated	GCCCCCTCATTGGGGATAG	5248	19	52.00	57.89	5	5813.80	
	10	Wild	GGGGCCGCCACTTGA	17698	15	52.00	73.33	4	4553.99	g.17705g>a
		Mutated	GGGGCCGTCACTTGA	17698	15	50.00	66.67	4	4537.98	

Table A2.2 – Probes for mutations listed on Table 5.1 – Mutations for gene MYBP-C with Tm = 50ºC ±2ºC.

Gene	Num	State	Sequence	Position	Length	Tm	GC	RunLength	MW	Mutation
MYBPC3	1	Wild	GATAGACCTGTGTGCAT	5129	17	50.00	47.59	2	5154.38	g.5137a>c
		Mutated	GATAGACCGGTGTGCAT	5129	17	52.00	52.94	2	5130.37	
	1.1	Wild	AGATAGAGCTGTGTGCATG	5128	19	52.00	42.15	1	5707.75	g.5137a>c
		Mutated	AGATAGAGCGGTGTGCATG	5128	19	50.00	42.15	1	5683.74	
	2	Wild	CAGATAGACCTGTGTGC	5131	17	52.00	52.94	2	5170.39	g.5139g>c
		Mutated	CAGATAGAGCTGTGTGC	5131	17	52.00	52.94	1	5130.37	
	2.1	Wild	CAGATAGACCGGTGTGC	5131	17	50.00	52.94	2	5146.38	g.5139g>c
		Mutated	ACAGATAGAGCGGTGTGCA	5130	19	50.00	42.15	1	5714.76	
	3	Wild	TGGGCATCGGTGATG	5159	15	48.00	60.00	3	4481.95	g.5166g>a
		Mutated	CTGGGCATTGGTGATGT	5158	17	52.00	52.94	3	5108.35	
	4	Wild	CGGTAGCTGCCAGTG	5183	15	50.00	66.67	2	4537.98	g.5190a>g
		Mutated	CGGTAGCCGCCAGTG	5183	15	52.00	73.33	2	4553.99	
	4.1	Wild	CGGGAGCTGCCAGTG	5183	15	48.00	66.67	2	4513.97	g.5190a>g
		Mutated	GCGGGAGCCGCCAGTGA	5182	17	52.00	64.76	2	5123.37	
	5	Wild	ACAGCGGTAGCTGCC	5187	15	50.00	66.67	2	4569.00	g.5194a>c
		Mutated	ACAGCGGGAGCTGCC	5187	15	52.00	73.33	3	4544.99	
	5.1	Wild	ACAGCGGTAGCCGCC	5187	15	48.00	66.67	2	4585.99	g.5194a>c
		Mutated	CACAGCGGGAGCCGCCA	5186	17	52.00	64.76	2	5194.50	
	6	Wild	CCGTGGACAGTGAGA	5243	15	48.00	60.00	2	4503.97	g.5250g>a
		Mutated	ACCGTGGATAGTGAGAT	5242	17	50.00	47.59	2	5105.36	
	6.1	Wild	ATCGTGGACAGTGAGAT	5242	17	48.00	47.59	2	5105.36	g.5250g>a
		Mutated	CATCGTGGATAGTGAGATT	5241	19	50.00	42.15	2	5731.76	
	6.2	Wild	ACTGTGGACAGTGAGAT	5242	17	48.00	47.59	2	5105.36	g.5250g>a
		Mutated	CACTGTGGATAGTGAGATT	5241	19	50.00	42.15	2	5731.76	
	6.3	Wild	CATTGTGGACAGTGAGATT	5241	19	50.00	42.15	2	5731.76	g.5250g>a
		Mutated	TCATTGTGGATAGTGAGATTG	5240	21	52.00	38.95	2	6318.14	
	6.4	Wild	ACCGGGGACAGTGAGAT	5242	17	50.00	52.94	2	5097.36	g.5250g>a
		Mutated	CACCGGGGATAGTGAGATT	5241	19	52.00	47.37	2	5723.76	
	6.5	Wild	CATCGGGGACAGTGAGATT	5241	19	52.00	47.37	2	5723.76	g.5250g>a
		Mutated	CATCGGGGATAGTGAGATT	5241	19	48.00	42.15	2	5707.75	
	6.6	Wild	CACTGGGGACAGTGAGATT	5241	19	52.00	47.37	2	5723.76	g.5250g>a
		Mutated	CACTGGGGATAGTGAGATT	5241	19	48.00	42.15	2	5707.75	
	6.7	Wild	CATTGGGGACAGTGAGATT	5241	19	48.00	42.15	2	5707.75	g.5250g>a
		Mutated	TCATTGGGGATAGTGAGATTG	5240	21	50.00	38.95	2	6294.13	
	7	Wild	CTCACCGTGGACAGT	5247	15	48.00	60.00	2	4593.10	g.5254a>c
		Mutated	CTCACCGGGGACAGT	5247	15	50.00	66.67	4	4569.00	
	7.1	Wild	CCTCATCGTGGACAGTG	5246	17	52.00	58.82	2	5195.40	g.5254a>c
		Mutated	CCTCATCGGGGACAGTG	5246	17	50.00	58.82	2	5171.39	
	7.2	Wild	CCTCACTGTGGACAGTG	5246	17	52.00	58.82	2	5195.40	g.5254a>c
		Mutated	CCTCACTGGGGACAGTG	5246	17	50.00	58.82	2	5171.39	
	7.3	Wild	CCTCATTGTGGACAGTG	5246	17	48.00	52.94	2	5179.39	g.5254a>c
		Mutated	CCCTCATTGGGGACAGTGA	5245	19	52.00	52.63	3	5788.79	
	7.4	Wild	CCTCACCGTGGATAGTG	5246	17	52.00	58.82	2	5195.40	g.5254a>c
		Mutated	CCTCACCGGGGATAGTG	5246	17	50.00	58.82	2	5171.39	
	7.5	Wild	CCTCATCGTGGATAGTG	5246	17	48.00	52.94	2	5179.39	g.5254a>c
		Mutated	CCCTCATCGGGGATAGTGA	5245	19	52.00	52.63	3	5788.79	
	7.6	Wild	CCTCACTGTGGATAGTG	5246	17	48.00	52.94	2	5179.39	g.5254a>c
		Mutated	CCCTCACTGGGGATAGTGA	5245	19	52.00	52.63	3	5788.79	

	7.7	Wild	CCCTCATTGTGGATAGTGA	5245	19	50.00	47.37	3	5796.79	g.5254a>c
		Mutated	CCCTCATTGGGGATAGTGA	5245	19	48.00	47.37	3	5772.78	
	8	Wild	CCCTCACCGTGGACA	5249	15	50.00	66.67	3	4649.40	g.5256g>a
		Mutated	CCCTCACTGTGGACA	5249	15	48.00	60.00	3	4633.30	
	8.1	Wild	CCCTCATCGTGGACAG	5248	16	50.00	62.50	3	4922.22	g.5256g>a
		Mutated	CCCCTCATTGTGGACAG	5248	17	50.00	58.82	4	5235.42	
	8.2	Wild	CCCTCACCGGGGACA	5249	15	48.00	66.67	3	4625.30	g.5256g>a
		Mutated	CCCCTCACTGGGGACAG	5248	17	52.00	64.76	4	5227.42	
	8.3	Wild	CCCCTCATCGGGGACAG	5248	17	52.00	64.76	4	5227.42	g.5256g>a
		Mutated	CCCCTCATTGGGGACAG	5248	17	48.00	58.82	4	5211.50	
	8.4	Wild	CCCTCACCGTGGATAG	5248	16	50.00	62.50	3	4922.22	g.5256g>a
		Mutated	CCCCTCACTGTGGATAG	5248	17	50.00	58.82	4	5235.42	
	8.5	Wild	CCCCTCATCGTGGATAG	5248	17	50.00	58.82	4	5235.42	g.5256g>a
		Mutated	CCCCCTCATTGTGGATAGT	5247	19	52.00	52.63	5	5861.82	
	8.6	Wild	CCCCTCACCGGGGATAG	5248	17	52.00	64.76	4	5227.42	g.5256g>a
		Mutated	CCCCTCACTGGGGATAG	5248	17	48.00	58.82	4	5211.50	
	8.7	Wild	CCCCTCATCGGGGATAG	5248	17	48.00	58.82	4	5211.50	g.5256g>a
		Mutated	CCCCCTCATTGGGGATAGT	5247	19	50.00	52.63	5	5837.90	
	9	Wild	CCCCTCACCGTGGAC	5250	15	52.00	73.33	4	4674.50	g.5257g>a
		Mutated	CCCCTCATCGTGGAC	5250	15	50.00	66.67	4	4658.40	
	9.1	Wild	CCCCTCACTGTGGAC	5250	15	48.00	66.67	4	4658.40	g.5257g>a
		Mutated	CCCCCTCATTGTGGACA	5249	17	50.00	58.82	5	5275.44	
	9.2	Wild	CCCCTCACCGGGGAC	5250	15	50.00	73.33	4	4650.40	g.5257g>a
		Mutated	CCCCCTCATCGGGGACA	5249	17	52.00	64.76	5	5267.44	
	9.3	Wild	CCCCCTCACTGGGGACA	5249	17	52.00	64.76	5	5267.44	g.5257g>a
		Mutated	CCCCCTCATTGGGGACA	5249	17	48.00	58.82	5	5251.43	
	9.4	Wild	CCCCTCACCGTGGAT	5250	15	48.00	66.67	4	4658.40	g.5257g>a
		Mutated	CCCCCTCATCGTGGATA	5249	17	50.00	58.82	5	5275.44	
	9.5	Wild	CCCCCTCACTGTGGATA	5249	17	50.00	58.82	5	5275.44	g.5257g>a
		Mutated	CCCCCTCATTGTGGATAG	5248	18	50.00	55.56	5	5548.62	
	9.6	Wild	CCCCCTCACCGGGGATA	5249	17	52.00	64.76	5	5267.44	g.5257g>a
		Mutated	CCCCCTCATCGGGGATA	5249	17	48.00	58.82	5	5251.43	
	9.7	Wild	CCCCCTCACTGGGGATA	5249	17	48.00	58.82	5	5251.43	g.5257g>a
		Mutated	GCCCCCTCATTGGGGATAG	5248	19	52.00	57.89	5	5813.80	
	10	Wild	GGGGCCGCCACTTGA	17698	15	52.00	73.33	4	4553.99	g.17705g>a
		Mutated	GGGGCCGTCACTTGA	17698	15	50.00	66.67	4	4537.98	