UNIVERSITA' DEGLI STUDI DI PADOVA

SCUOLA DI DOTTORATO DI RICERCA IN

SCIENZE DELLE PRODUZIONI VEGETALI

INDIRIZZO AGROBIOTECNOLOGIE - CICLO XXII

Dipartimento di

AGRONOMIA AMBIENTALE E PRODUZIONI VEGETALI

Gene prediction and functional annotation in the *Vitis vinifera* genome

Direttore della Scuola : Ch.mo Prof. Andrea Battisti

Supervisori : Ch.mo Prof. Angelo Ramina

Ch.mo Prof. Giorgio Valle

Dottorando : CLAUDIO FORCATO

DATA CONSEGNA TESI

01 febbraio 2010

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Padova, 14/01/2010 Claudio Forcato

A copy of the thesis will be available at *http://paduaresearch.cab.unipd.it/*

Dichiarazione

Con la presente affermo che questa tesi è frutto del mio lavoro e che, per quanto io ne sia a conoscenza, non contiene materiale precedentemente pubblicato o scritto da un'altra persona né materiale che è stato utilizzato per l'ottenimento di qualunque altro titolo o diploma dell'università o altro istituto di apprendimento, a eccezione del caso in cui ciò venga riconosciuto nel testo.

Padova, 14/01/2010 Claudio Forcato

Una copia della tesi sarà disponibile presso *http://paduaresearch.cab.unipd.it/*

ABSTRACT

In the last years the increasing number of sequencing projects and the availability of completely sequenced genomes pose the problem of searching for gene sequences in a rapid and reliable way. Bioinformatics is playing a fundamental role in this research field. In fact, many bioinformatic tools and software that consider multiple and heterogeneous evidence sources have been developed in order to improve the genome annotation.
Genome annotation can be divided in two distinct phases: gene prediction and functional annotation. The prediction phase is the process to identify the exact gene structure, delimiting the exon-intron boundaries and the localization of genes on the genome. Otherwise, the functional annotation is the action of characterizing predicted genes, assigning them a biological function, a metabolic role or describing structural features.
This PhD project focuses on the development of computational methods for the management of data coming from a genome sequencing project. The work consists on the implementation of a bioinformatic platform for gene prediction and functional annotation of the *Vitis vinifera* genome. This work has been carried out in collaboration with CRIBI bioinformatic group, that is member of the Grape sequencing project.
The annotation platform consists of two distinct modules. The first module regards gene prediction. Different computational methods showed a great reliability to discover molecular signals and to reconstruct gene boundaries, becoming fundamental in the annotation at genome-level. These methods are represented by ab-initio predictors, genome alignments of ESTs or proteins or comparative genomics.
Otherwise, in the second module of annotation platform, the predicted genes are functionally characterized, adopting mainly a similarity approach. This approach bases on the assumption that regions highly conserved maintain the original functions or roles also in different species.
This project includes also the development of databases and tools to store and retrieve genome data. In particular, the PhD work focused on the implementation of a XML-based query system that permits the information retrieval through web page access and, in the next future, also through web-services workflows.

SOMMARIO

Negli ultimi anni il crescente numero di progetti di sequenziamento e la disponibilità di genomi completamente sequenziati hanno posto il problema della ricerca di sequenze geniche in modo rapido e affidabile. La Bioinformatica sta giocando un ruolo fondamentale in questo campo di ricerca. Infatti, sono stati sviluppati molti strumenti informatici che utilizzano dati molteplici ed eterogenei al fine di migliorare l'annotazione genomica.
L'annotazione genomica può essere suddivisa in due fasi distinte: la predizione genica e l'annotazione funzionale. La predizione genica consiste nell'individuazione dell'esatta struttura del gene, determinando il confine esone-introne e la localizzazione dei geni sul genoma. Invece, l'annotazione funzionale è il processo di caratterizzazione dei geni, che assegna loro una funzione biologica, un ruolo metabolico o che descrive le loro caratteristiche strutturali.
Questo progetto di dottorato prevede lo sviluppo di metodi computazionali per la gestione dei dati provenienti da progetti di sequenziamento genomico. Il lavoro consiste nella realizzazione di una piattaforma bioinformatica per la predizione genica e l'annotazione funzionale del genoma di *Vitis vinifera*. Questo lavoro è stato svolto in collaborazione con il gruppo di bioinformatica del CRIBI, membro del progetto internazionale di sequenziamento del genoma di vite.
La piattaforma di annotazione è suddivisa in due moduli. Il primo modulo riguarda la predizione genica. Diverse metodiche computazionali hanno mostrato una grande affidabilità nella ricerca di segnali molecolari e nella ricostruzione della struttura genica, diventando strumenti fondamentali per l'annotazione genomica. Questi metodi sono rappresentati da predittori ab-initio, da allineamenti di EST o proteine sul genoma o dalla genomica comparata.
Invece, nel secondo modulo della piattaforma di annotazione, i geni predetti sono caratterizzati funzionalmente attraverso l'utilizzo di un approccio di similarità. Questo approccio si basa sul presupposto che le regioni altamente conservate mantengono le funzioni e i ruoli originali anche in specie diverse.
Questo progetto prevede anche lo sviluppo di banche dati e strumenti per immagazzinare e recuperare i dati di annotazione. In particolare, il lavoro di dottorato si è concentrato sulla realizzazione di un sistema di query basato su XML che permette il recupero delle informazioni attraverso pagine web e, nel prossimo futuro, anche attraverso l'utilizzo di workflow basati sui web services.

RINGRAZIAMENTI

Desidero innanzi tutto ringraziare il prof. Angelo Ramina e il prof. Giorgio Valle per avermi dato l'opportunità di frequentare la scuola di dottorato e di specializzarmi nella materia che spero diventerà il mio lavoro.

Ringrazio tutti i colleghi del CRIBI e del gruppo del prof. Valle, ma in particolar modo il laboratorio di Bioinformatica al completo: Davide Campagna, Lucas Stefanutti, Alessandra Bilardi, Svetlin Manavski, Elisa Caniato, Alessandro Vezzi, Riccardo Schiavon, Riccardo Rosselli, e molti altri che ho sicuramente dimenticato. Grazie per i vostri suggerimenti e la vostra professionalità.
Un grazie particolare va ad Erika Feltrin, per avermi dato preziosi consigli durante tutti i 3 anni di dottorato, e soprattutto ai miei due fraterni compagni di viaggio Alessandro Albiero e Nicola Vitulo: grazie per il vostro aiuto, per la vostra competenza ma soprattutto per la vostra amicizia.

Ringrazio il dott. Claudio Bonghi, il dott. Alessandro Botton e tutto il gruppo del prof. Ramina ad Agripolis.

Un ringraziamento particolare va a mio fratello Massimiliano, un continuo e costante esempio di professionalità, ma soprattutto di vita.

Un grazie di cuore a Gabriele e Carla, per la loro generosità, per il continuo supporto e per la loro infinita pazienza.

Infine un ringraziamento speciale va a Laura, per la sua capacità di essermi così meravigliosamente vicino.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Part I

INTRODUCTION

1

GRAPE GENOME PROJECT

CONTENTS

1.1 *vitis vinifera* GENOME

The sequencing of *Vitis vinifera* genome is a fundamental step in crop science because it allows to exploit information derived from DNA decoding to elucidate aspects of grapevine physiology, biochemistry, genetics and breeding. Moreover the genome sequencing allows to develop new theoretical concepts and molecular tools, to assess in detail the grape wide genetic variability and to preserve and exploit genetic natural resources for a modern viticulture.
In 2007 the French-Italian public consortium published the first draft of the sequence, correspondent to the 8.4x assembly [52]. The genetic source was the highly homozygous ($\approx$ 93%) Pinot Noir inbred line PN40024, produced by INRA-France. The grapevine genome, estimated of 485Mb, is three times higher than the *Arabidopsis thaliana* one (125Mb), the first plant genome sequenced, and more than six times smaller than human genome (3Gb). The *Vitis vinifera* genome is the fourth one produced for flowering plants, the second for woody species and the first for a fruit crop [1; 2; 104].

Grapevine is the first sequenced genome for fruit crop

Grapevine was selected because of the reduced size of the genome, the many biological properties and the important place in the cultural heritage of humanity. In particular, grapevine offers the possibility to study different pathways and biological aspects of particular interest, such as synthesis of anthocyanins, flavonoids, polyphenols and other secondary metabolites, berry quality, extreme susceptibility to pathogens, disease resistance and adaptation to different growing environments, biology of reproduction, etc. Moreover, the gene catalog determination allows to set up species-specific microarrays for gene expression studies and the availability of genome sequence makes possible the characterization of germoplasms present in various worldwide collections. Finally, a decisive aspect for the genome characterization is its

economic impact that fully justifies the financial effort being the grapevine an important crop.

485Mb organized in 19 chromosomes, containing 30 thousands genes

The 485Mb genome is organized in 19 chromosomes and, according to the 8.4x prediction, they contain about 30,000 protein-coding genes. This value is noticeably lower than that for *Populus trichocarpa* (45,555 in a 485Mb genome) and for *Oryza sativa* (37,544 in a 389Mb genome). In grapevine, the gene density is not homogeneous, with large regions that alternate low and high gene densities. The density pattern is shared by poplar but not by *Arabidopsis* and rice.

About 40% of the genome is made up of repetitive/transposable elements (TEs). A significative part of TEs and retrotransposons localize within introns.

Focusing on the proteomic aspects, grapevine shows an expansion of gene families with aromatic features. In particular, *stilbene synthase* and *terpene synthase* (more specifically *monoterpene syntase*) families show an higher gene copy number than in other species. Stilbene synthase drives the synthesis of resveratrol, known for the beneficial effects on human health. Terpenes are components of resins and aromas and are essential for plant growth and development and for the interaction with the environment.

Furthermore, a high number of disease-resistance genes have been identified. The resulting proteins contain a nucleotide binding site (NBS) and a leucine-rich repeat (LRR) responsible for recognition specificity. They are organized in clusters and their heterogeneity seems to function in genome evolution as the basic material for the generation of new resistance specificities [107].

From a phylogenetic point of view, the grapevine haploid genome seems to derive from three ancestral genomes. This palaeo-hexaploidation (true hexaploidation event or subsequent genome duplications) is shared with poplar and *Arabidopsis* (dicotyledons) but is absent in rice (monocotyledon). However, these species have recently experienced whole genome duplication events (WGD), not present in grape. In particular, poplar underwent one recent WGD event and *Arabidopsis* two.

An alternative scenario assumes three genome duplications for dicotyledons, one shared by all dicots, one by *Arabidopsis* and poplar, but not *Vitis*, and one specific for *Arabidopsis* and poplar [107]. Finally, *Vitis* underwent two genome duplications through an hybridization event subsequent to the separation from the *Arabidopsis* and poplar lineage.

1.2 PHD PROJECT

My PhD project has been focused on the implementation of a bioinformatic platform for gene prediction and functional annotation, applied to the *Vitis vinifera* genome.

During my PhD activity, I have been working in collaboration with CRIBI bioinformatic group that is an active member of

the VIGNA consortium. Together with the french counterpart Genoscope, VIGNA constitutes the international Grape genome project for the sequencing and annotation of *Vitis vinifera* genome. The 8x genome release was sequenced, annotated and published in 2007. Sequence assembly and annotation of the 12x release is in progress and will be published in the next months. The official 12x gene prediction, that in the following chapters is named V0, was accomplished by Genoscope and consists of 26,347 protein-coding genes. The CRIBI group was in charge of the functional annotation of the V0 predicted genes.

Gene prediction regarding the 12x assembly consists of 26,347 genes

Although the CRIBI contribution to the Grape project is officially limited to functional annotation, a more comprehensive project was developed, which comprises gene prediction step and tools for querying databases. Indeed, the increasing sequencing of new genomes requires appropiate computational methods in order to extract information from the raw nucleotide sequence. Thus, the main goal has been to develop a bioinformatic platform that greatly automates all the procedures required for a complete genome annotation, in a species-unspecific manner, to make the platform re-usable in other sequencing projects.

However, the genome research field is continously in progress, and newer, more efficient and reliable computational techniques are daily developed and released. The modularity embedded in the platform allows to easily extend and improve the annotation procedures, integrating any updated software.

Modularity was a key-word in the CRIBI annotation platform development

The resulting platform is developed in Java and Perl programming language and is composed of several phases, summarized in the figure 1.1:

1. *gene prediction*: the starting point is the raw nucleotide sequence. The genome is evaluated for the presence of gene evidence through different approaches: *ab-initio* gene finders, comparative genomics and EST or protein alignments. Each method produces its own results, describing chromosome positions and putative intron-exon structures of genes. Sometimes these results do not agree and can highlight conflicting situations, hard to solve. At this point, the platform makes use of an integrative software, called JIGSAW, that combines all the evidence sources to produce the final consensus predictions. In the analysis of conflicting regions or prediction quality, the Gbrowse genome browser represents a good utility, allowing to visually inspect any genome regions. Gene prediction ends with the production of the gene catalog.

   The output of gene prediction stage is the gene catalog for the genome under analysis

2. *functional annotation*: this stage assigns biological functions, metabolic roles or structural features to the predicted gene set. The functional annotation is mainly based on a similarity approach: information is collected from inter-species sequence similarity, assuming that regions highly conserved maintain the same functions or roles also in different species. In the world wide web a great amount of databases and re-

   Functional annotation is based on a similarity approach

sources are available to infer gene product properties including database of protein sequences (UniProt) or protein domains (PFAM, SMART, Prosite), metabolic maps (KEGG), software for predicting cellular localizations or structural features. Moreover, these associations between genes and proteins or domains are fundamental to classify genes with Gene Ontology terms.

3. *database storage*: a relational MySQL database is created to consistently store all the annotation data. This database has a gene-centered star topology.

4. *database interface*: in the final step, a query system, based on XML, facilitates access to the stored data in a rapid and interactive manner.

High-quality predictions are decisive for achieving solid functional annotations

The more important step is the gene prediction, because it determines the gene structures and consequently, the coding sequences. The following phases are strictly correlated to the quality of the gene catalog. For this reason, a strategy based on deep-sequencing of transcriptome data was studied to check the prediction quality. In the following chapters, an application of this quality check in a comparison between the official v0 prediction release and v1 (a testing prediction produced by CRIBI platform) will be shown.
In this PhD thesis, the main computational methods integrated in the first version of the annotation platform will be presented. The platform allows to fully annotate a genome starting from the raw nucleotide sequence.
However there are some aspects that need to be improved. In the next future, comparative genomics approaches will be enriched and enforced, the database structure will be re-modeled and the database interface will be strengthened and optimized for web service implementation.

1.2.1 Genome data management

Genome projects represent great computational challenges

An underestimated problem in genome annotation projects is the huge amount of data to analyze. These data are represented by the millions of nucleotides of the genome, the thousands of ESTs to align or the thousands of genes to functionally characterize. Any computational routine that has to process such data sets is extremely time and memory consuming, arising significative computational issues. Moreover, alignment or prediction algorithms not optimized to deal with large data set worsen the situation. Even the adoption of powerful servers is not sufficient to solve the problem, without a clever strategy to optimize the processor usage.
A possible solution is represented by the parallelization, that is the distribution of computational processes in different servers or processors. This strategy allows the subdivision of the process in several routines and their contemporary execution, decreasing

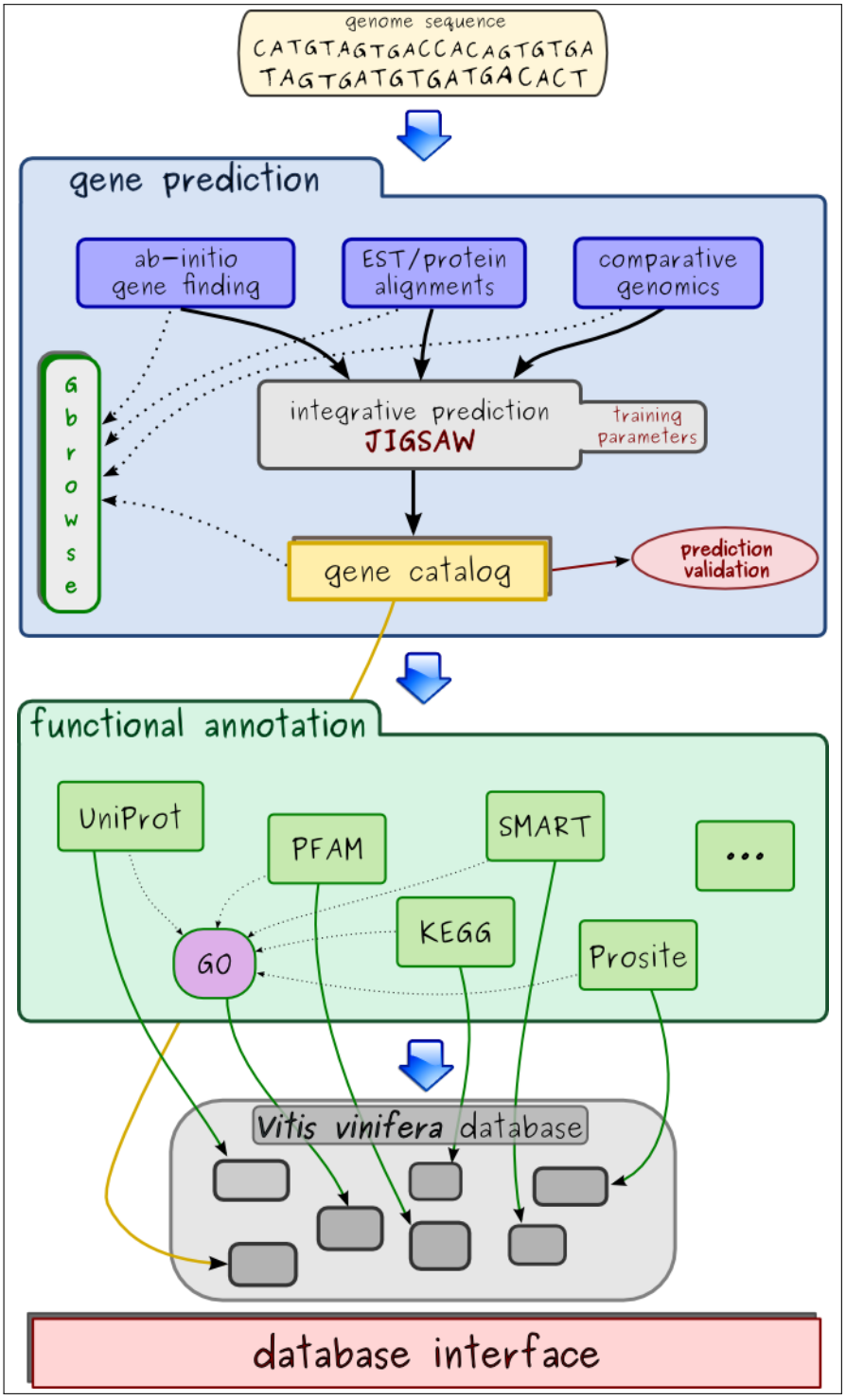


Figure 1.1: The platform is mainly divided in *gene prediction* and *functional annotation* step. All the predicted genes and related functional properties are stored in a MySQL database and can be collected and analyzed through a web-based interface.

The annotation platform adopts distributed computations

the execution time for the entire process.
In the developed platform, a parallel approach based on a client-server architecture was adopted (Fig. 1.2). The client-side is an user-interface to the services or resources provided by the server application. The latter collects the client requests, processes them and returns the results to the client. In our case, the client transfers the input data to the server for the subsequent input elaborations. The server-side is represented by a manager daemon that gathers the input data, subdivides them and distributes the smaller data sets to different servers. The manager daemon makes a continuous check of the process status and optimizes the processor occupation, maximizing the work mass at each moment and minimizing the entire execution time. Once all the processors have completed their tasks, the manager daemon collects and assembles the results and send them back to the client.
In particular, the platform uses this strategy in the prediction and functional annotation stages. In the former, the distribution regards the scaffolds making up the Grape genome; in the latter, the distribution of protein-coding genes.
From the hardware point of view, the computing distribution is realized thanks to 15 servers with a total of 30 processors present at the CRIBI bioinformatic laboratory, and to the usage of LICC[1] cluster system formed by 20 nodes corresponding to 80 processors.

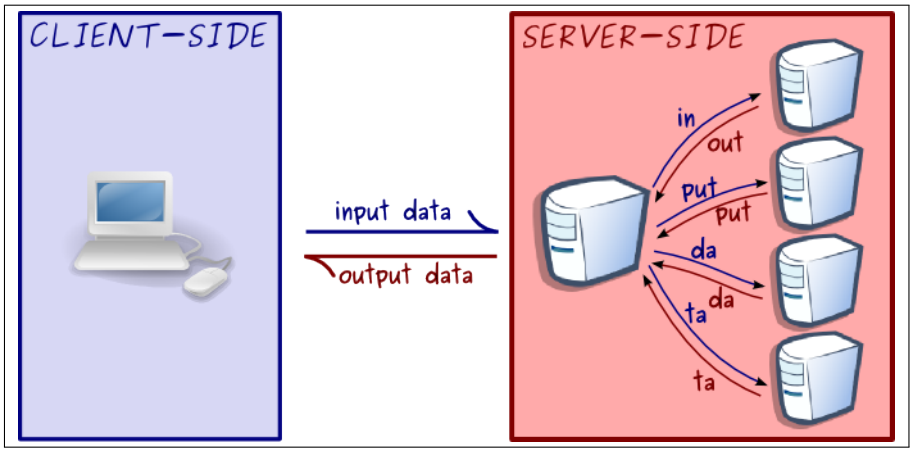


Figure 1.2: The client-server architecture needed to process the large amount of genome data. The server-side is characterized by a daemon that manages the process parallelization.

1 Laboratorio Interdipartimentale di Chimica Computazionale.

Part II

GENE PREDICTION

2

GENE EVIDENCE SOURCES

CONTENTS

The gene prediction is finalized to identify the gene catalog in a genome sequence. It includes several phases as identification of gene positions and localizations on the genome, determination of exon-intron boundaries and discover of molecular signals (e.g. start/stop codons, splicing sites, etc.). In genome projects, these tasks are accomplished through several computational methods that showed great reliability in discovering structure signals and reconstructing gene boundaries. The three major computational approaches are *ab-initio* gene finders, ESTs or protein alignments and comparative genomics.
In this chapter, methods used in the prediction phase and present in the developed platform will be described.

2.1 GENE MODEL

Given a new genome, the most important task is to determine the structure of protein-coding genes, representing the majority of the transcribed and the translated genes. They show incredible diversity in size and organization but have some conserved features. Thus, before detailing the gene finding techniques, it could be useful to define the concept of *gene structure*.
The core of the gene is the coding sequence (CDS), that contains the nucleotides translated in the protein amino acids. The coding region begins with the initiation codon, which is usually ATG and ends with one of three termination codons: TAA, TAG or TGA. On either side of the coding region are DNA sequences that are transcribed but are not translated. These untranslated regions (UTR) often contain regulatory elements that control protein synthesis. UTRs and CDS are called *exons* that represent the

transcribed portion of the gene. Exons may be interrupted by *introns*, DNA sequences that are cut in the mature transcripts by a *splicing* process to form the messanger RNA (mRNA). The splicing machinery is able to recognize the intron boundaries thanks to consenus sequences present in the exon-intron junctions: the *donor* site, represented by GT or GC bases at the beginning of the intron, and the *acceptor* site, represented by the AG bases at the end of the intron [22].
These features are shared by all protein-coding genes and are used by *in silico* methods to build hypothetical gene models based on the raw nucleotide sequence.

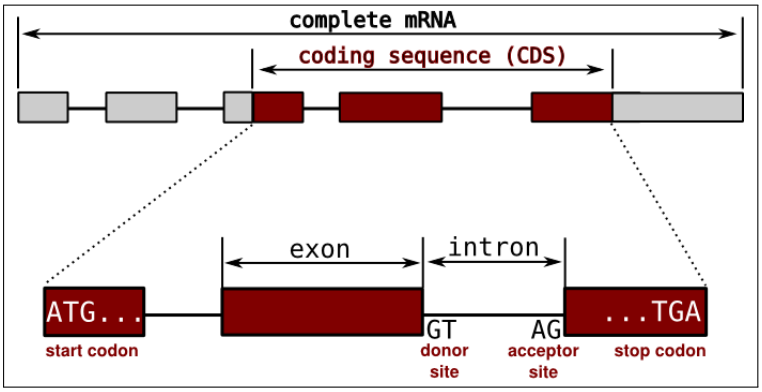


Figure 2.1: The typical gene structure detected by gene-finding programs, formed by CDS exons and introns. UTRs are usually not predicted.

2.1.1 Annotation file format

GFF is the standard file format used for annotation data exchange

The large increase of genome projects and gene-finding methods arise the need to develop a standard format for the computational description of the gene features. This would favor data exchange and development of modular gene-finding programs that could be extended with the integration of external information. A format called GFF ("General Feature Format") was proposed as a protocol for the transfer of feature information[1]. It is a tab-delimited file with a record-based structure, where each feature is described on a single line, and line order is not relevant. Each line is formed by 9 fields that describe:

1. *seqid*: the ID of the landmark used to establish the coordinate system for the current feature.
2. *source*: algorithm or software used to determine this feature.
3. *type*: the type of the feature, e.g. CDS, exon, gene, mRNA, etc.
4. *start*: the start 1-based coordinate of the feature. Start is always less than or equal to end.

1 http://gmod.org/wiki/GFF3

5. *end*: the end of the feature.

6. *score*: the score of the feature, represented by similarity values for alignment methods or p-value scores for *ab-initio* gene finders. This field can be set to "." in absence of score.

7. *strand*: the strand of the feature, "+" for plus strand and "-" for minus strand.

8. *phase*: for CDS features, it indicates how to read the codon series. The allowed values are 0, 1 or 2 and, together with strand, it determines the reading frame.

9. *attributes*: a list of feature attributes in the format tag=value, separated by semicolons. Possible attributes are "ID", an unique identifier, or "Name" and "Parent", necessary to groups exons into transcripts and transcripts into genes.

An example of GFF file is following:

```
chr1 JIGSAWGAZE gene 14402501 14405865 . + . ID=JGVv10.3
chr1 JIGSAWGAZE mRNA 14402501 14405865 . + . ID=JGVv10.3.t01;Parent=JGVv10.3
chr1 JIGSAWGAZE UTR  14402501 14402707 . + . Parent=JGVv10.3.t01
chr1 JIGSAWGAZE CDS  14402708 14403942 . + . Parent=JGVv10.3.t01
chr1 JIGSAWGAZE CDS  14405003 14405348 . + . Parent=JGVv10.3.t01
```

2.2 *ab-initio* PREDICTIONS

The *ab-initio* software is the first gene-finding method to be described [20; 4; 71; 28]. These systems are a great resource in gene prediction because they produce gene structures quickly and inexpensively. Otherwise, these positive traits are counterbalanced by the not very high levels of accuracy and reliability. However, a great advantage of *ab-initio* gene finders is the detection of gene evidences that can not be discovered through other means, as expression data.
Ab-initio gene finders try to identify the gene structure starting from the raw nucleotide sequence. They are usually based on Generalized Hidden Markov Models, a technique that uses matrices of stochastic parameters obtained from a previous training phase. This *training phase* is performed on a set of curated genes, with a known exon-intron structure, called *training data set*. In this way, the gene finder gains the ability of generalization, the capacity of inferring the general properties from a limited set of "example" genes. After the training step, the gene finder should be able to predict the gene structure in novel unseen sequences, based on the intrinsic sequence-based characteristics of the training data set.

Gene finders gains the generalization ability trough a training phase

An important aspect to be considered is that the training should be specific for the genome under analysis. In fact, sequence features as codon bias and splicing signals vary from organism to organism and the nearest phylogenetic neighbor does not necessarily possess compatible parameters. The risk to use a gene

finder trained with no species-specific genes is to obtain inaccurate predictions [62].
Moreover, big concerns regard the quality of the training set. The set of training genes has to be sufficiently representative of the full complement of genes in the genome, so that the gene finder is able to generalize traits of model genes and recognize novel genes in the DNA sequence. The problem that can arise is the *overtraining*, the opposite of generalization, that is the gene finder ability to detect only the model genes. Thus, the large quantity and the variety of the training genes are fundamental to obtain accurate predictions [73].

Training set has to be numerous, various and species-specific

2.2.1 SNAP and GeneID

The *ab-initio* gene finders that have been integrated in the platform are SNAP [62] and GeneID [83; 16]. SNAP is a typical GHMM finder but, compared to others, it is provided with a training module that makes it easily adaptable to different organisms. Thus, the GHMM parameters can be adjusted in a species-specific manner. Otherwise, GeneID is developed with a hierarchical structure, different from other common GHMM finders. At first, fixed-length signals (e.g. splicing sites, start/stop codons, etc.) are predicted and scored along the sequence using PWMs. Then, potential exons are constructed from these sites and scored as the sum of the defining sites plus the score of a Markov model for coding DNA. Finally, a dynamic programming algorithm defines the gene structure that maximizes the sum of the score of the assembled exons. Moreover, it is already supplied with grape parameters and it is not necessary to train the model.

2.3 EST ALIGNMENTS

After the transcription of protein-coding genes, the primary transcript is processed by the splicing machinery to remove introns and a 5′ cap and a 3′ poly-A tail are added forming the mature mRNA. Therefore, the mature mRNA embodies all the exon knowledge of the gene. Expressed Sequence Tags are short subsequences of mRNA (400-800 bases), obtained from the sequencing of ends of cDNA clones (DNA complementary to mRNA) coming from a cDNA library. Thus, ESTs represent portions of expressed genes.
The mapping and alignment of ESTs onto the genome represent a fundamental resource to genome research for localizing the genes and for reconstructing the intron-exon boundaries [44; 99; 109]. Moreover, these alignments could be useful to investigate the splicing mechanisms and alternative transcripts formation.
Two important issues arise in the alignment procedure: a) high sequencing error rate, resulting in low quality EST sequences and b) low availability of ESTs for the genome under analysis.

EST alignments represent the "gold standard" for gene prediction

Both problems heavily affect the alignment procedures, because sequencing errors and the use of EST coming from closely related organisms produce unperfect matches and uncorrect splicing site detection, complicating the exact identification of real exon-intron structure. To address these needs, several alignment algorithms have been developed to map ESTs on the genome: BLAT [61], Sim4 [42], EST_GENOME [78], Spidey [110], GMAP [113], etc. They envisage the EST opening, spreading adjacent EST portions in distant genome regions. They allow to finely model the alignments, offering the possibility to set different options as maximal intron length, minimal similarity or identity percentage, gap opening and extension penalties, etc.
In EST alignments, three different public EST libraries were used:

- a set of dicotyledons EST, formed by about 1 million of sequences
- a set of *Vitis* ESTs, excluding the sequences of *vinifera* variety
- a set of *Vitis vinifera* ESTs

In addition, the *Vitis vinifera* public ESTs were integrated with a private set coming from the sequencing of berry and leaf transcriptome of *Vitis vinifera* with 454 technology.
The three libraries were separated to model the different specificity level, with the intention to differently weight the EST alignments in the prediction stage.
To align both *Vitis* libraries, GMAP software was employed because it showed a great ability and precision with EST coming from the same or very close organisms. In particular, EST alignments were filtered according with identity (85%) and coverage (70%) criteria.
Otherwise, a different approach was adopted dealing with dicotyledon ESTs. In fact, some problems were encountered in finding a good alignment software for this kind of sequences. This library is different from the others because it consists of EST coming from organisms at different phylogenetic levels. In this case, a greater sensitivity was required by alignment algorithms. Another limiting problem was represented by the huge amount of dicotyledon sequences compared with *Vitis* ones. This large quantity becomes particularly problematic, since alignment sensitivity is directly proportional to execution time. To align dicotyledon ESTs decreasing the execution time, the solution was the partitioning of EST database. The dicotyledon sequences were aligned on the genome using a sensitive, fast but inaccurate algorithm, Spidey. In this way, there was the identification of *matching-islands*, that are chromosome regions with at least one EST match. Also in this case, the alignments were filtered with 60% identity and 60% coverage. A matching-island is composed by a set of matching ESTs, that represent a subset of the EST database. For each island, a more rigorous alignment was executed in parallel between subsets of EST database and

Dicotyledon ESTs are partitioned in smaller subsets to decrease the computation time

genome subregions using EST_GENOME algorithm with a identity threshold of 70%. The parallel execution of dicotyledon EST alignments was possible thanks to the usage of the LICC cluster system. By this way, a sensitive, accurate and fast (few days) alignment for an enormous EST library was obtained.

2.3.1 UTRs prediction

UTRs are usually not contemplated in the standard gene-finding methods

The computational methods for gene prediction usually focus on the modeling of coding sequence structure, neglecting the untranslated regions. This is because of the variable length and composition of UTRs and the lack of some shared features that could help in their identification. Moreover, not making part of the translated sequence, the holy grail of gene prediction, they are not a priority and their identification is postponed in subsequent phases.
However, the UTR annotation represents a valuable resource for studying promoters and regulative patterns.
Although, the standard procedure in the developed platform does not include the discovery of UTRs, a module that allows to annotate untranslated regions based on EST evidences was implemented. Since ESTs derive from mRNA, there should be EST sequences that correspond to UTR transcript portions.
The module develops in several steps:

- *selection of EST alignments*. A filter is applied such that EST alignments with introns longer than $15,000$ nucleotides are rejected, representing probably false alignments.

- *validation of the initial or final exon*. Only the exons with at least 50 (or less if the exon is smaller) bases covered by more than one EST are considered for the UTR extension. By this way, poorly-confirmed exons are blocked, avoiding unreliable UTR elongation.

- *determination of the elongation region*. A region can be elongated until the number of EST evidences are greater than a predefined threshold, avoiding extensions due to isolated mis-aligned ESTs. The percentage threshold is computed referring to the ESTs that overlap the first positions of the initial or terminal exon.

- *structure definition*. For each elongation region, the module tries to define a tentative structure consensus, based on quality values: in an elongation region $E[i,j]$, each nucleotide is scored for the dominant structure feature (exon/intron), given the EST evidences. The nucleotide i is considered of high quality if the percentage of exonic (or intronic) EST tracks for the i position is greater than a predefined q value. The total number of high quality nucleotides in the $E[i,j]$ determines if the elongation region is a valid UTR.

An example of an UTR extension is illustrated in the figure 2.2. In this case, there are 5 ESTs (red boxes) covering the first positions of an exon (blue box). Given an elongation threshold of 40%, the software tries to extend UTR until 3 EST evidences are present. In the figure the elongation region is highlighted with a gray color. After the elongation region identification, the UTR structure is defined according to the dominant exon/intron evidence in each interval. The quality for each interval is computed as the number of the exon (or intron) evidences over the EST coverage. The definition of a quality threshold q determines the number of high-quality nucleotides contained in the elongation region, and so, the acceptance of UTR consensus structure. In the figure, this is represented by yellow boxes.

UTR structure consensus is determined by coverage and quality of EST signal

This module represents the first attempt for UTR annotation and there is space for improvements. Future efforts should be directed to give a strong statistical support to the definition of consensus structures.

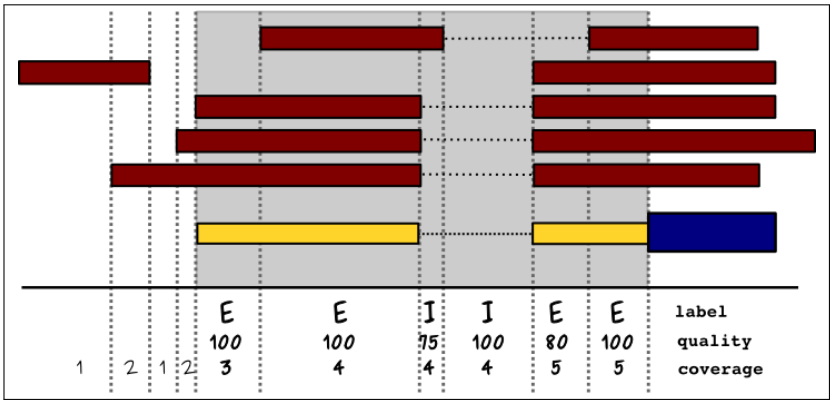


Figure 2.2: The red boxes represent EST alignments and the dotted lines the interposed introns. Blue box is the first exon of a predicted gene. The elongation region is colored in gray and it is subdivided in intervals determined by alignment boundaries (vertical dotted lines). The UTR consensus structure is described by yellow boxes. The E or I labels stand for exon or intron dominant evidence for each interval; coverage is the number of EST evidence for each interval; quality is the number of exon (or intron) ESTs over the total number of EST evidences.

2.4 PROTEIN ALIGNMENTS

The mapping of proteins to a genome sequence is very similar to mapping of ESTs. The main differences are that the sequence is composed of amino acids and that the protein does not contain UTRs. In fact, the mature protein-coding transcript is transformed in a protein through the translation process. The portion of the transcript that correspond to a protein is the coding sequence (or CDS), that is defined by a start codon and a stop codon.

For protein alignment, a matching-island approach, similar to that used for EST alignment, was adopted

In the alignment phase, the UniProt database was used as reference, representing an universal protein resource. However, the great amount of proteins provided by UniProt ($\approx$ 6 millions) requires the adoption of the same matching-island approach used for dicotyledon EST alignment. At first, a filter was applied to the database, keeping all the proteins belonging to plant species. Then, the protein set was roughly aligned onto the genome using BLAT, a fast but inaccurate algorithm, to isolate the genome regions that present protein matches. Only the alignments with 85% of protein coverage were selected. For each matching-island, the protein alignments were improved using a sophisticated aligner, GeneWise [15], that considers splicing sites and start/stop codons for alignment refinement. However, the GeneWise sensitivity and precision are balanced by tremendous execution time. Thus, a further filtering procedure was performed for each matching-island. The filter consisted to select the first 10 hits (if present) in a Blast [7] alignment. Therefore, for each region at most 10 proteins were aligned using GeneWise, greatly reducing the execution time. Thanks to the parallel alignment processes, results for the entire genome were obtained in few days.

2.5 COMPARATIVE GENOMICS

The rational behind the usage of comparative genomics for gene prediction is that coding regions are greatly conserved in phylogenetically related genomes. Indeed, accumulation of mutations in the coding regions brings to the loss of biological function of genes. Thus, patterns of conservation between DNA sequences of closely related organisms probably highlight syntenic portions, and more specifically, coding regions.
The increasing availability of genomes offers the possibility to analyze the conservation patterns and sintenies of entire chromosomes. In the same time, it needs appropriate alignment algorithms able to deal with large genomic sequences. In fact, the usual alignment programs, like BLAST, FASTA, etc., becomes enormously inefficient and time-consuming for genome-level comparisons. To address these needs, programs such as Blastz [91], Lagan [18], WABA [11], MUMmer [35; 36] have been developed for whole genome alignments. All such algorithms essentially share the same *anchor-based* approach. This procedure implies a) a fast sorting of exact or lightly degenerate matches, named *seeds*, b) a clustering procedure to group together neighboring seeds, named *anchors*, c) the determination of the longest collinear subset of not-overlapping anchors, that constitutes the alignment base-chain, and d) a final accurate alignment (e.g. Smith-Waterman) to refine the regions of anchors and between the anchors.

Whole Genome Alignment software is built according to an anchor-based approach

In the developed platform, a comparative module was inserted realizing the pairwise comparison of *Vitis vinifera* with the three

available plant genomes: *Arabidopsis thaliana*, *Populus trichocarpa* and *Oryza sativa*. The comparative module was realized with MUMmer software. It is a program that searches for Maximal Unique exact Matches between genomes in a very fast way, using a suffix-tree method. In particular, the PROmer [65] aligner was used, which performs all the matching and alignment routines on the six amino acid translation of the DNA input sequences. The reason is to increase the sensitivity because DNA sequence is not highly conserved as the amino acid translation.
The results of PROmer execution are set of matches for each pairwise comparison, done chromosome by chromosome. These matches can highlight some levels of conservation, maybe signaling hypothetical coding portions. However, they can also correspond to pseudogenes, to regulative patterns, to conserved no-coding regions, or, at worse, to false positives. Coding regions are more conserved than non-coding ones, although sequence conservation may also occur in regions other than the protein coding ones, particularly in closely related species. Thus, the PROmer matches have to be filtered in some ways. Some Perl scripts were developed in order to clean the match set from spurious alignments and to build a tentative gene structure. The pipeline consists of several steps:

No-coding regions can be conserved in closely related organisms

- *filtering*: selection of matches with 50% identity, a length greater than 45 nucleotides and a small quantity of in-frame stop codons.
- *correction*: the coordinates of matches with stop codons are adjusted such as to outline the largest portion of the match without stop codons.
- *clustering*: matches at a distance less than 6,000 nucleotides are grouped together.
- *merging*: in case of overlapping matches (putative exons), the longest ORF is selected.
- *construction*: a gene model is built for each cluster of matches.

The resulting gene structures do not claim to be real, but only to give a putative coding evidence. In the next future, the comparative module will be improved, at first allowing a multi-alignment between genomes rather than a pairwise one. This should increase the matching reliability. Secondly, the gene model reconstruction has to be enriched with further sequence-based constraints, as evaluations on possible ORFs, etc.

2.6 GENOME BROWSER

Genome browsers are useful web-based applications for displaying genomic annotation and other features. These tools offers the possibility to the end-user to "surf" the genome, scrolling

and zooming through arbitrary regions of a genome.
At CRIBI laboratory, there is a Gbrowse server [100] that is linked with Grape database. All evidence lines coming from the above described prediction methods are loaded on the Gbrowse, allowing a better visualization of evidence tracks and helping in the analysis of gene models.
The figure 2.3 shows a genome region in a Gbrowse view.

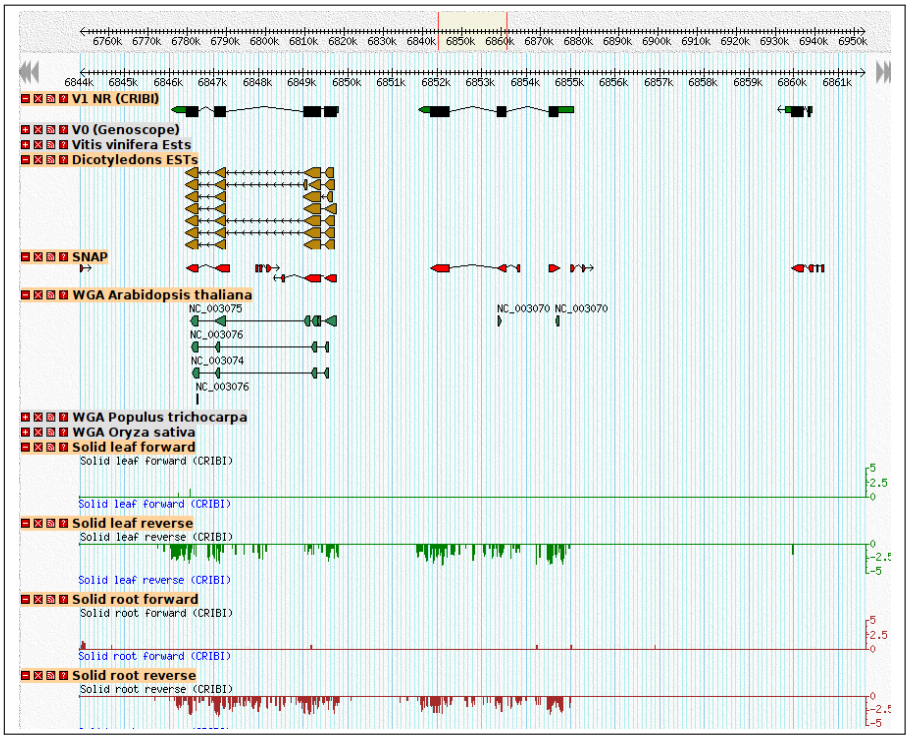


Figure 2.3: In a Gbrowse view, the gene evidence tracks are divided in sections of different styles and colors. The colored blocks denote putative exons, the block-linking lines describe introns. Green and red spikes show SOLiD signal in different organs.

3

COMBINING THE EVIDENCES

CONTENTS

In the chapter 2, the *ab-initio* and comparative methods for gene prediction adopted in the platform have been described. Each track produces a single line of evidences of the coding or transcribed portion of the genome. In this chapter, a software system combining evidence sources and producing the final consensus prediction is presented. This tool is suitable to resolve ambiguous gene structures and genome information-poor regions.

3.1 RESOLVING THE CONFLICTS

The methods for gene finding implemented in the platform represent a small part of available gene prediction techniques. Gene finding is still a subject of active research and new, efficiently performing software are almost daily released. For example, there are several tools that incorporate expression data such as ESTs or proteins directly in the ab-initio programs to improve prediction quality, or that based on phylogenomic approach coupled to GHMM to build significant gene structures [97; 72; 109]. All the research efforts are directed to produce a single, unique, high-quality prediction based on the greatest number of available evidences. In the CRIBI platform, this goal is reached by a successful *integrative* gene finder, called JIGSAW [6; 5], that seeks to integrate and combine multiple evidences from different sources, to produce the final consensus prediction. The great modularity of this software was decisive for the purposes. In fact, it is able to deal with disparate and heterogeneous data, as protein or EST matches, gene predictions from one or more programs, splicing sites predictions, etc. and, from a theoretical point of view, it has no limitation in using informa-

An integrative gene-finder combines multiple evidence sources to build the final consensus prediction

tion coming from different sources. This flexibility is an important trait in case of availability of any novel data type, e.g. data coming from new sequencing technology as SOLiD and Solexa. Moreover, JIGSAW allows to independently define the ability of each evidence track to predict gene features, e.g. EST alignment tracks can be configured for predicting exons or introns rather than start and stop codons. The integrative flexibility and the ability to model each evidence contribution were decisive in the choice of using JIGSAW as the final combiner in the platform.
In the gene prediction procedures there are two main problems to be faced for good achievement:

- the **lack of expression data**: the mapping of EST to the genome sequence is considered the "gold standard" and the more important step for defining the true exon-intron structure. However, there are genome regions that are made up of rarely expressed genes, have only a limited number of expressed sequence tags, and are supported by conflicting gene finder predictions. In most cases, these genes are not discovered in the prediction step.

- the **gene structure definition**: predictions from several gene finding programs or expression data alignments are able to define gene regions, but fail to infer precise gene structures. Indeed, the evidence tracks can give diverse gene models, creating conflicting situations that must be correctly resolved.

An example of conflicting predictions is shown in Fig. 3.1. Each evidence line is described by means of blocks and block-linking lines that outline the predictions. The first line, denoted as V1 NR CRIBI, is the JIGSAW final prediction that results from the combination of the tracks below. The red circles highlight the ambiguous regions, where the different evidence lines disagree on gene structure. In particular, we can see the lack of an intron predicted by SNAP, an alternative EST splicing of dicotyledon EST alignments, an exon break in the whole genome alignment of *Arabidopsis thaliana*, but mainly the exon-island showed by *Vitis vinifera* EST alignments. These situations make hard the detection of a precise gene structure, allowing for one, two or more gene models. The decision about the correct or alternative gene structures is usually left to time-consuming manual curation procedures.

JIGSAW determines gene models, solving all the conflicts

JIGSAW is an automated, statistically principled method that tries to solve these issues by a different evaluation of evidence sources.

3.2 JIGSAW COMBINER

JIGSAW is a gene finding system that automates the process of predicting gene structure from multiple sources of evidence.

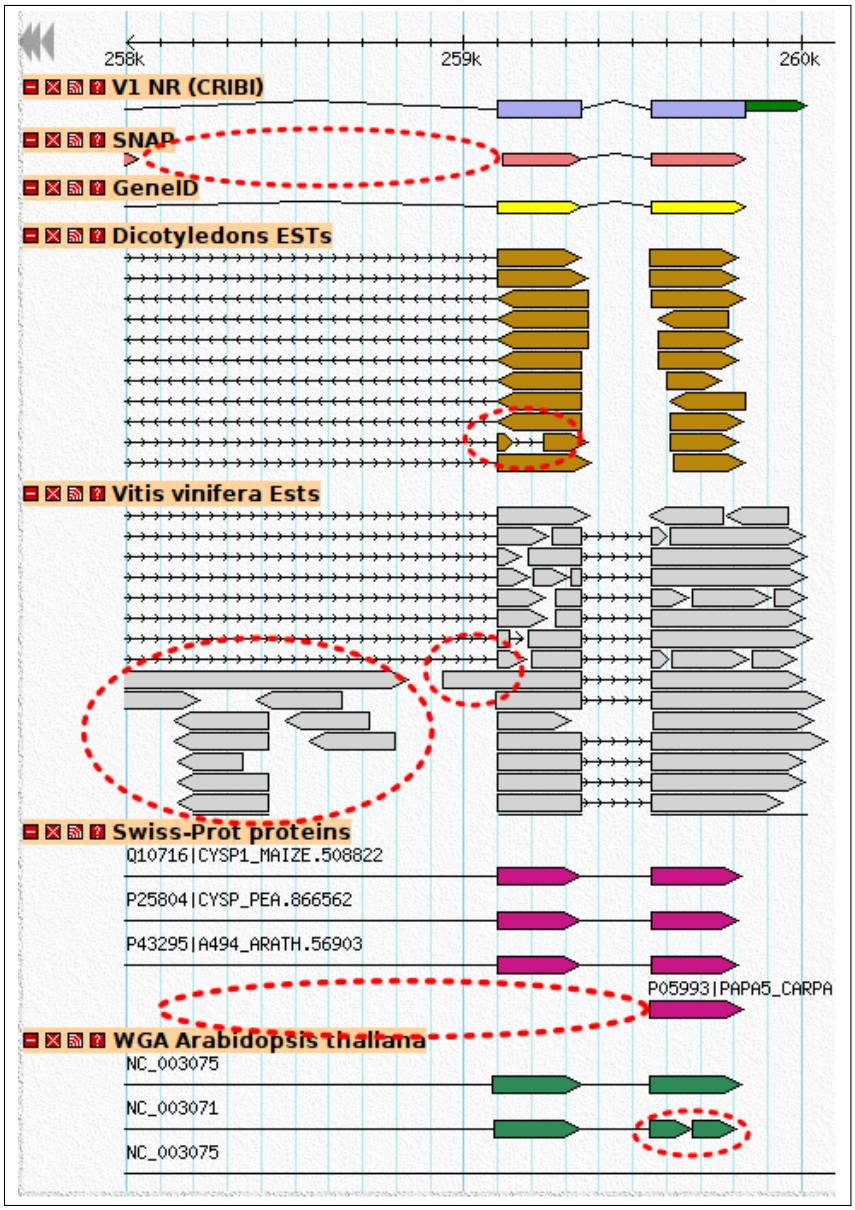


Figure 3.1: Evidence lines are described trough boxes, representing putative EXONS, and box-connecting lines, representing putative INTRONS. The red circles highlight the conflicts.

It assigns to each evidence line a weight computed by statistical evaluations on a training set. This value estimates the prediction accuracy of different tracks and favors the most reliable evidence lines in the solution of overlapping, disagreeing predictions. From algorithmic point of view, JIGSAW adopts a mixed strategy, combining GHMM based algorithm and a statistical approach for the evaluation of evidence lines. A brief description of the main features of JIGSAW program is given below [6; 4].

3.2.1 Gene structure model

Gene features are modeled with states

Gene structures are modeled with ten states l that stand for exon, intron and intergenic regions. In particular, exons are described by *single*, *initial*, *internal1*, *internal2*, *internal3* and *terminal* states. Intron has *intron1*, *intron2*, *intron3* labels. Finally, intergenic regions are represented by *intergenic* state. The three different labels for intron and internal exons are necessary to model the phase of codon break due to introns. In this way, a gene model is formed by a series of sequence labels $l_1, l_2, ..., l_z$.

Genome intervals are determined by signals and evidence boundaries

At first, JIGSAW partitions the input sequence in subsequences $S_{x...y}$, where x and y can be a) location of signals or b) boundaries of evidence alignments. In the first case, partial gene models are computed linking together signals as start codons, stop codons, donor and acceptor sites. The linkage between signals has to be biologically meaningful, e.g. a stop codon can be linked back to a previous acceptor in terminal exons, or start codon in single exons, but not to a donor site. In the second case, intervals are determined by boundaries of an evidence that does not necessarily span a complete exon. For example, in Fig. 3.2 three evidence tracks are shown. The boundaries of alignments determine the intervals K_i, K_{i+1}. In any interval, the evidences have the same behavior and scoring pattern.
At this point, a dynamic programming algorithm computes scores for each interval, using GHMM features, as transition probabilities between states, and vectors decoding evidence information, to define the most probable state l_j of a specific interval I_j. The final gene model results from the products of probabilities for each interval.

3.2.2 Evidence representation

The evidence information content is coded by a six feature vector

An example of JIGSAW flexibility is that it allows each evidence source to model independently six gene features: start codon (sta), stop codon (stp), intron (inr), coding (cod), donor (don), acceptor (acc). The information coming from the lines of evidence is represented by a six feature vector v_{type}, one for each feature type: $(type_k^1, ..., type_k^m)$, where $type$ is one of the above mentioned six gene features, k is the position in a S sequence and m represents the evidence source. In other words, acc_k^x is the predicting confidence of the program x on nucleotide

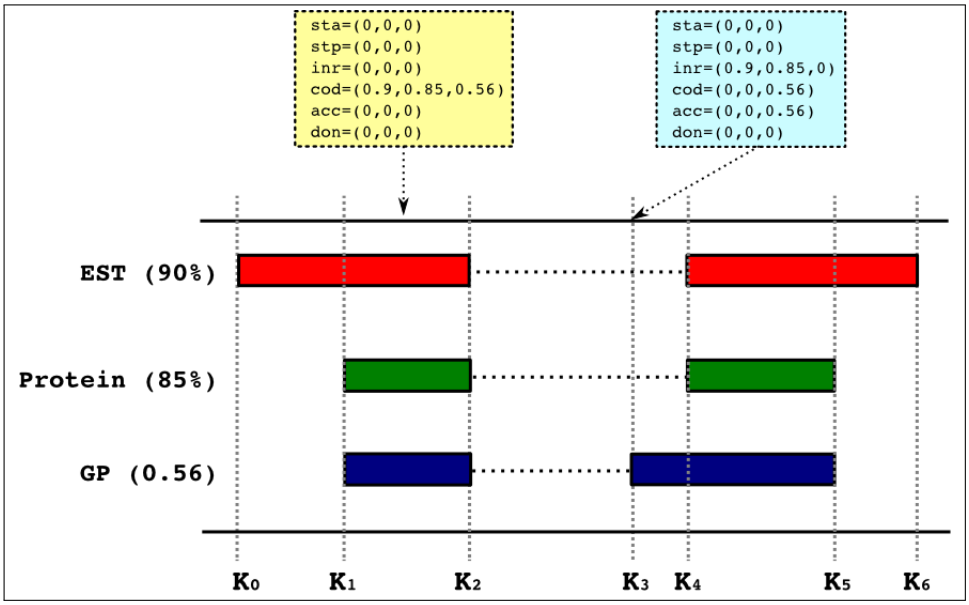


Figure 3.2: The intervals K_i, K_{i+1} are determined by alignment boundaries and sequence signals. The three evidences are EST alignments with 90% identity, protein alignments with 85% identity and a gene finder with 0.65 accuracy value. Examples of evidence vectors are shown for interval $k_1 + 1, k_2 - 1$ and for k_3 position.

k for an acceptor site. The score represents the confidence in the program accuracy, and it can be 1 or 0, to indicate respectively the presence or absence of a prediction, or a similarity value, in case of transcript or protein alignments. Figure 3.2 shows an example of vectors for three evidence tracks (EST, Protein, GenePredictor). The vector set for positions from $K_1 + 1$ and $K_2 - 1$ is

$$\begin{aligned} B_{k_1+1,k_2-1} &= \{v_{sta}, v_{stp}, v_{inr}, v_{cod}, v_{acc}, v_{don}\} \\ &= \{(0,0,0),(0,0,0),(0,0,0),(0.9,0.85,0.56),(0,0,0),(0,0,0)\} \end{aligned}$$

The JIGSAW gene prediction problem is to find the most probable parse[1] t given a S sequence and a set of E input evidences. The contribution of evidence sources is encoded by an independent conditional probability to obtain a parse t given a set of feature vectors B_k and a sequence S. This is evaluated as the product of six independent probability values, each conditioned on one of the gene feature vectors, $P(type_i|v_{type})$, determined in a training procedure. For instance, given an *Initial* q state and a S[i, j] interval, the first nucleotide i should correspond to the beginning of a start codon and to the first coding base of a protein. The nucleotide at $j + 1$ position should be the first base of a donor site. In this case, given the evidence, the probability is determined by the product of $P(sta_i|v_{sta})$, $P(cod_{i,j}|v_{cod})$, $P(don_{j+1}|v_{don})$ and $1 - P(type_{i,j}|v_{type})$ for the three remaining

1 series of states spanning the entire sequence.

feature type, that do not align with *Initial* state. In this way, JIGSAW realizes a probabilistic model to compute the probability of a parse conditioned on the input evidence.

3.2.3 Training procedure

JIGSAW needs a training process to estimate both parameters for GHMM transition probabilities and evidence vectors. Above all, the encoding of evidence vectors represents the core of JIGSAW statistical approach and it presents a valuable procedure to determine the evidence vector probabilities. This process contemplates an accuracy classification of the entire set of possible vectors, using a *decision tree* and basing on the labeled sequences of the training set.

A decision tree realizes an accuracy classification of the gene feature vectors

The parameter estimation is independently computed for all six gene features and depicts the probability to obtain a gene feature given the evidence vector: $P(type|v_{type})$.
At first, the observed feature vectors are divided into two, accurate and inaccurate groups, on the basis of a $c(v_{type})$ value. This is the percentage of cases in which v_{type} is observed to correctly predict $type$. If $c(v_{type}) > 0.5$, the v_{type} is assigned as *accurate*. At this point, the decision tree tries to maximize the separation between accurate and inaccurate vectors, calculating cut-off values that define subregions in the feature vector space. In general, a decision tree is recursively constructed from the root down to the leaves, where at each recursive step it selects the rule that maximally reduces the entropy of the distribution of categories among the training set [73]. In this case, the decision tree divides the vector space in V_n subregions of similar accuracy and determines the rules that allow a test vector to be assigned to each subregion. By this way, the probability of a test vector $P(type|v_{type})$, assigned to the V_2 subregion by decision tree rules, is the average accuracy of the individual training vectors that constitute V_2: $\frac{\sum_{v \in V_2} c(v)}{|V_2|}$.

The feature vector space is divided in subregions of similar accuracy

3.3 v1 PREDICTION RELEASE

In the section 1.2, the existence of an official prediction release of *Vitis vinifera* genome, called v0, have been mentioned. This prediction has been released by Genoscope, using a new method that considers the WTS[2] data coming from Solexa sequencing method [37]. The v0 prediction release is composed of 26,347 genes.
However, experimental and computational analysis have highlighted some genome regions that show strong transcription evidences but are uncovered by predicted genes. This observations convinced the italian members of Grape consortium of the need of a new reliable prediction release. Therefore, CRIBI group

2 Whole Transcriptome Shotgun.

was charged to produce a new prediction release plugging much gaps as possible.
It has been decided to hold the v0 prediction, in spite of its problems, as the *starting point* for the new release, denoted as v1.
The unofficial v1 release is the result of the integration between v0 and the CRIBI prediction, that is based on JIGSAW.
The CRIBI prediction was obtained using the platform described in the previous chapters, that consists of:

v1 release results from the integration of v0 and CRIBI predictions

- *ab-initio predictors*: it was used two gene finders, SNAP and GeneID, with *Arabidopsis* parameters. A well-curated set of 600 full-length genes of *Vitis vinifera* was available for training procedures. However, due to the small number of genes, the annotated gene set was used for JIGSAW training. Thus, the *Arabidopsis* parameters were chosen because they are the sole parameters for plant species supplied by gene finders.

- *EST alignments*: the platform used three EST evidence tracks. Each one represents a different library: dicotyledons, *Vitis vinifera*, *Vitis* (*vinifera* excluded). In addition to *Vitis vinifera* library, some ESTs coming from berry and leaf sequenced with Roche/454 were integrated.

- *protein alignments*: the protein matches were obtained by a fine-grain alignment of a filtered UniProt database.

- *comparative genomics*: three plant genomes were aligned chromosome by chromosome against grape genome. Putative gene structures are derived from the union of near matches, filtered by similarity and length criteria.

- *integration step*: the final prediction is obtained using JIGSAW, trained with 600 known full-length genes. All evidences were configured to predict all six gene features, with the exception of comparative tracks, allowed to predict only coding and intron features.

The integration between v0 and v1 was done by treating the v0 prediction as a novel line of evidence, like an EST track. By this way, much trust was put in the ability of JIGSAW to appropriately weight the v0 line. The whole procedure can be divided as follow:

- *training step*: a training procedure in the 600 full-length genes was necessary to calibrate the JIGSAW parameters for the added v0 line.

- *integration step*: v0 and CRIBI predictions were merged using JIGSAW, resolving the possible conflicts and producing the MERGED-JIGSAW-GAZE version.

- *enrichment step*: MERGED-JIGSAW-GAZE was further enriched with genes predicted exclusively in v0 or CRIBI, but not included in the MERGED-JIGSAW-GAZE version by JIGSAW.

In case of conflicts caused by overlapping genes, they were left out from the new prediction, considering the related region not resolved. The enriched version is referred to as ENRICHED-MERGED-JIGSAW-GAZE.

- *cleaning step*: ENRICHED-MERGED-JIGSAW-GAZE was cleaned by filtering genes with $\geqslant$ 30% of their length similar to transposons, mobile elements, repeats or low-complexity regions, that were searched by RepeatMasker (unpublished Smit et al. http://repeatmasker.org). This cleaning procedure cut 3,885 genes from the final prediction, v1, consisting of 29,971 genes.

v0 and v1 releases are very similar, showing a great percentage of overlap as showed in figure 3.3. However, there are some important differences that outline two different predictive profiles. In the table 3.1 are reported some statistics about v0 and v1 prediction releases.

Table 3.1: Genome prediction statistics.

Genome feature	v0	v1	% v0	% v1
genome length	486,265,422	486,265,422	100	100
GENE bp	170,122,387	153,895,835	34.99	31.65
CDS exon bp	29,958,906	32,839,888	6.16	6.75
UTR bp	9,096,250	7,065,410	1.87	1.45
INTRON bp	131,067,231	113,990,537	26.95	23.44
INTERGENIC bp	316,143,035	332,369,587	65.01	68.35

Genome feature	v0	v1
n.genes	26,347	29,971
n.cds exon	156,767	142,337
n.utr	35,815	42,400
n.intron	135,707	117,789
CDS exons/gene	5.95	4.75
n.single-exon genes	2,132	6,377

Genome feature	mean v0	mean v1	median v0	median v1
GENE length	6,456.99	5,134.82	3,574.00	2,741.00
CDS exon length	191.10	230.72	122.00	129.00
UTR length	253.98	166.64	185.00	127.00
INTRON length	965.81	967.75	212.00	249.00
entire CDS length	1,137.09	1,095.72		

3.3.1 Prediction models

The data showed in the table 3.1 outline some important differences between the two predictions, that allow to characterize and describe the prediction behavior. Two main phenomenons can be observed:

1. **Gene fragmentation** in v1: in v1 there are more genes, 29,971, compared with 26,347 of v0, and the mean gene length is greatly smaller. Moreover, v0 predicts a greater

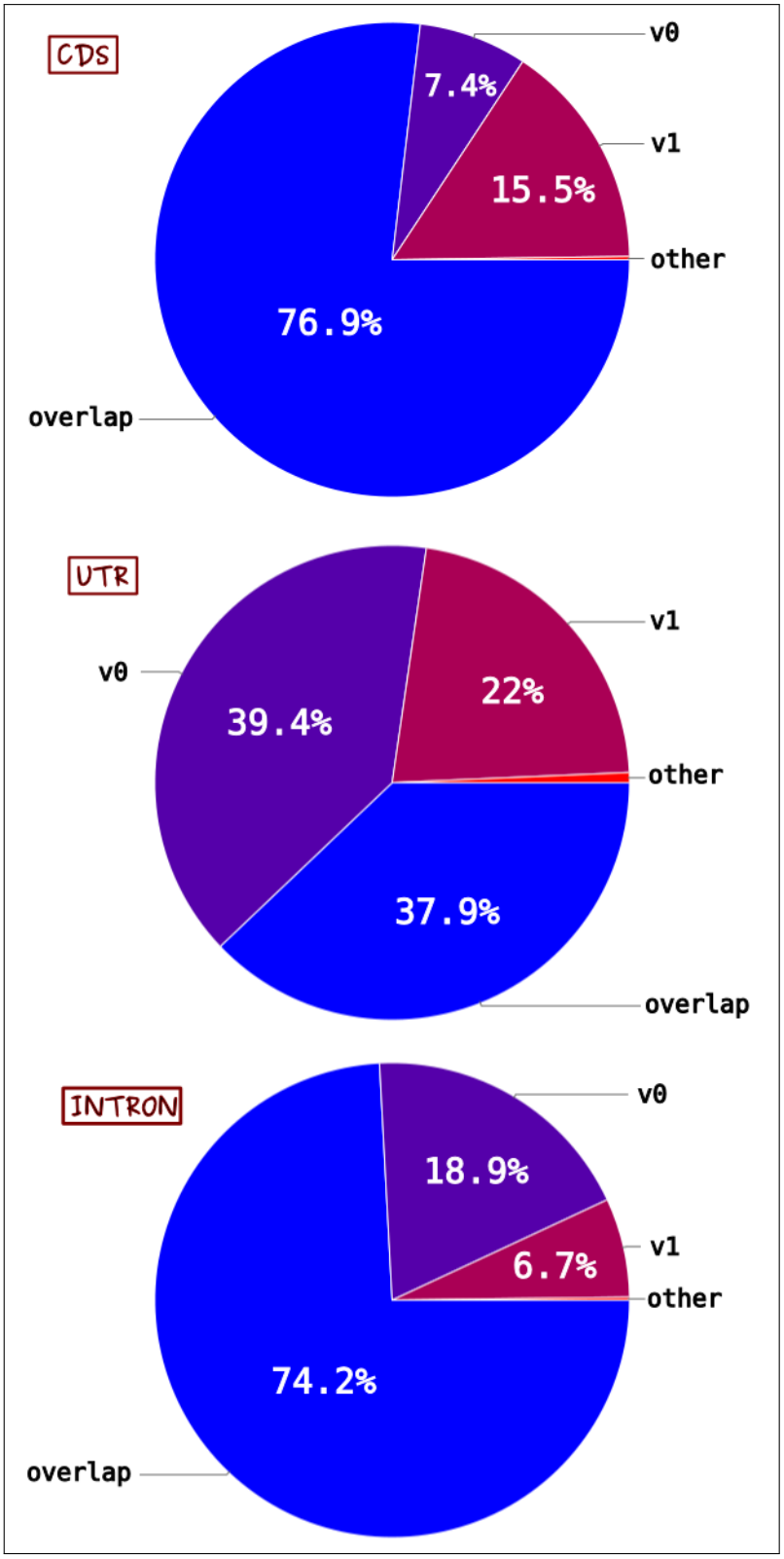


Figure 3.3: The three pies represent the nucleotides commonly predicted (*overlap*) by both v0 and v1 for CDS, UTR and INTRON categories, respectively. *v0* and *v1* sections refer to the nucleotides exclusively predicted by each single release; *other* section refer to unclassified nucleotides, e.g. nucleotides predicted by both releases as belonging to the same category but in opposite directions.

amount of *intron* bases than v1 equal to about 3% of the entire genome. At same time, there is a little increase of coding portion in v1. This aspect could mean that v0 joins genes that in v1 are considered independent, by forming intronic bridges. According to this hypothesis, there should be less and longer v0 genes, with an increase of intron nucleotides, which was exactly the observed situation. A noticeable statistic supporting this hypothesis concerns the small number of single-exon genes that are present in v0. Moreover, it seems that v0 has a general tendency to predict genes with more exons (Fig. 3.4). This is particularly evident in genes with low number of exons.

2. **Exon crumbling** in v0: this terms stands for the subdivision of a single exon in two or more smaller ones, separated by small introns. This phenomenon is confirmed when observing the number of CDS exons per gene and the average of CDS exon length. v0 predicts one exon more than v1 on average in a gene, but these exons are shorter. This observation is confirmed by the greater absolute number of CDS exons in v0 despite the decrease of coding nucleotides compared with v1.

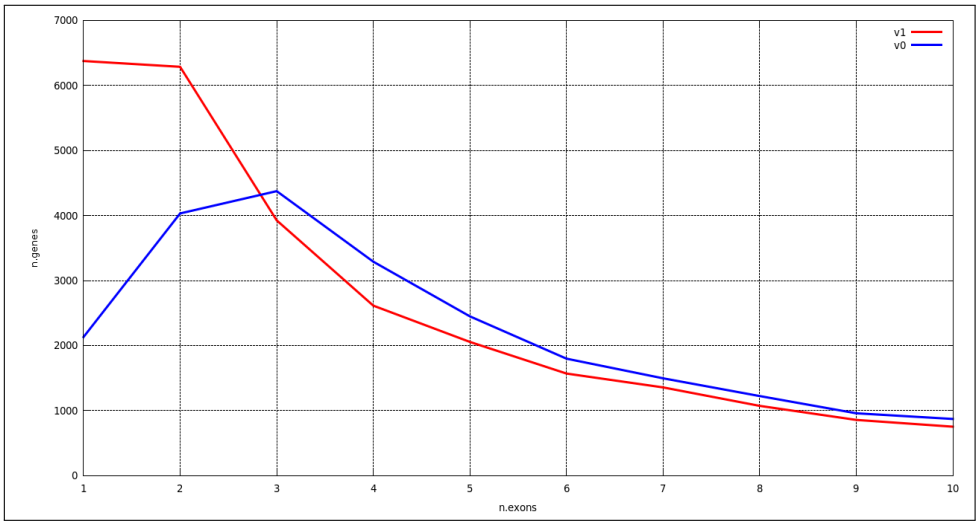


Figure 3.4: The graph outlines the distribution of genes as a function of the number of exons. The two lines describe the two prediction profiles, v0 (blue) and v1 (red).

The predictive model shows two phenomenons: gene fragmentation in v1 and exon crumbling in v0

To better explain the behavior of the prediction releases, a model outlined in Fig. 3.5 was developed. The graph represents an hypothetical genome region with some gene evidences, defined by the genes A and B in v0 and by genes C, D and E in v1. The model describes the exact situations represented by table 3.1: v0 has more CDS exons (5 versus 4), more introns (3 versus 1), a greater amount of intron nucleotides ($2 + 8 + 6 = 16$ versus 8). v1 has more genes (3 versus 2), a smaller number of CDS exons per gene ($4/3 = 1.33$ versus $5/2 = 2.5$) and a greatest number of

shorter utrs (6 utrs covering $4+1+1+1+1+1=9$ bases in v1 versus 3 utrs covering $12+2+5=19$ bases in v0). The v1 **gene fragmentation** phenomenon is clearly visible between the genes D and E. In v0 the related region outlines an unique gene, B, formed by two CDS exons linked by a 6 base-long intron. This behavior explains also the increase of single-exon genes and the smallest mean size of genes in v1. In the meantime, the v0 **exon crumbling** phenomenon can be observed in the gene A and C between CDS exon 1, 2 and 6. In this case, in v1 the exon 1 and 2 are merged in a longer exon 6 with the disappearance of the interposed intron, partially explaining the smaller number of exons per gene and the decrease of intron number and nucleotides. The presence of these short introns within an exon affects the mean intron length. In the table 3.1, this value is lower in v0. This event is not expected in a model with distant exons connected by long introns (gene fragmentation). This fact is due to the small size of the exon-breaking introns (exon crumbling), that enormously increase the number of introns with low effect on the intron nucleotide numbers, thus, decreasing the mean intron length in v0.

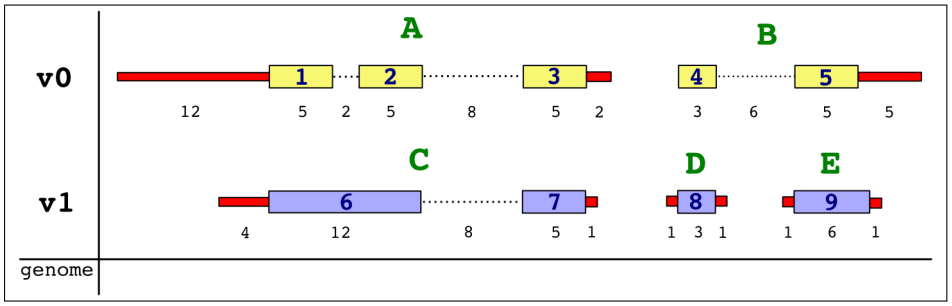


Figure 3.5: The prediction behavior of v0 and v1 in an hypothetical genome region. UTRs are in red, CDS exons in yellow (v0) and blue (v1) and introns are represented by dotted lines.

4

GENE PREDICTION VALIDATION

CONTENTS

In the previous chapter, a model that describes the different predictive behaviors of v0 and v1 have been presented. At this point, the task was to set up a method for validating and evaluating the prediction releases in absence of a positive control. A valuable solution could be represented by the comparison of the prediction with gene structures identified by several comprehensive EST libraries. However, the low availability of biological evidences and the partial and incomplete nature of these libraries bring to an unreliable quality estimate. This fact is particularly evident in low-transcribed genome regions, where lack of information can bias a prediction evaluation.
The next-generation sequencing technologies (NGS) offer an innovative resource for gene prediction refinement and evaluation, thanks to their depth of sequencing and sensitivity in de novo transcript discovery. These methods open new possibilities in genomic and transcriptomic research, but also arise new bioinformatic problems due to new data types that have to be managed and elaborated.
In this chapter, the main NGS methods, a software tool managing NGS data developed at CRIBI and its use in the validation of v0 and v1 predictions are described.

4.1 NGS METHODS

The Sanger method has been the standard approach for DNA sequencing for more than 30 years. Only in the last 4-5 years a new generation of sequencing techniques have started to become commercially available, and thanks to their innovative characteristics have opened new opportunities and potentialities in life

sciences [8; 108; 77]. Moreover, additional platforms will be available in the near future. To date, three NGS methods (Roche/454, Illumina/Solexa and AB/SOLiD) are available. They differ in their engineering configurations and sequencing chemistries, but share the principle of massively parallel sequencing of amplified DNA templates that are spatially separated in a flow cell. This parallel method is the true difference with Sanger technique that is based on the electrophoretic separation of chain-termination products produced in individual sequencing reactions. The NGS methods envisage a polymerase-based sequencing-by-synthesis (454 and Solexa) and ligase-based sequencing-by-ligation (SOLiD) chemistries.

NGS novelties regard sequencing chemistries and amplification protocols

The main NGS advantages are the very high-throughput sequence generation and the single DNA molecule amplification. In fact, the massively parallel process allows the NGS methods to produce up to gigabases of nucleotide-sequence ouputs in a single run. The NGS sensitivity due to this deep-sequencing is decisive in applications like de novo transcript discovery or mRNA expression profiling. Secondly, the amplification of DNA templates is made by *emulsion PCR* or *bridge PCR* techniques by which a single DNA molecule is immobilized onto specifically designed DNA capture beads or surfaces and is amplified independently, excluding competing or contaminating sequences and avoiding the need for cloning of DNA fragments.

NGS advantages are high-throughput and single DNA molecule amplification

The main limitation of NGS technologies is the short length of sequence outputs and the huge amount of generated data. These aspects represent a great challenge to the developers of analysis software [86].

NGS disadvantages are the shortness and management of output sequences

Possible applications of NGS technologies cover a wide range of fields and analyses as: the characterization and profiling of mRNAs, small RNAs, regulatory regions, structure of chromatin and DNA methylation patterns (ChIP-seq), microbiology and metagenomics [74; 93]. In particular, RNA-seq is a new powerful approach to map and quantify transcripts in biological samples, and it has shown some advantages over gene expression arrays. Indeed, after sequencing, reads are aligned to a reference genome, avoiding the hybridization problems affecting the microarray technology. In addition, RNA-seq [80; 112] shows a greater ability to determine RNA isoforms or sequence variants and to code the expression level. Moreover, it demonstrates an outperforming ability to detect low-level transcripts. Finally, RNA-seq is a fundamental resource to revise gene annotation, due to its ability to define the 5′, 3′ and exon-intron boundaries.

4.1.1 Roche/454

The main 454 features are pyrosequencing and emulsion PCR

The 454 technology [75] is the combination of single-molecule emulsion PCR amplification procedure and pyrosequencing, a polymerase-based SBS[1] strategy. The DNA template fragments

1 Sequencing-By-Synthesis.

are obtained through nebulization or sonication, and are ligated to adapter oligonucleotides. Subsequently, the library is diluted to single-molecule concentration, denatured and attached to individual beads carrying sequences complementary to adaptors. At this point, the beads are compartimentalized into water-in-oil microvesicles where the emulsion PCR amplification step is carried out. For each fragment, this results in a copy number of several million per bead. Then, the beads containing DNA templates are loaded into individual picoliter-plate wells, allowing one bead per well. Each well is enriched with sequencing enzymes. The pyrosequencing strategy is based on chemiluminescent detection of pyrophosphate released during polymerase DNA extension. Successive flow addition of the 4 dNTP and the incorporation of nucleotides complementary to the template strand result in a reaction that produce a light signal that is recorded by a CCD camera. The well images are decoded, filtered and translated into a sequence output by GS FLX *Titanium* instrument. A single run of GS FLX generates 400-600 million of high-quality bases (>1 million reads) with read length of $\geqslant$400 bases in 10 hours. The longer read length is the strength of 454 technology compared with Solexa and SOLiD, allowing an easier de novo assembly.

454 is the NGS technique that produces the longest sequences

4.1.2 Illumina

The Solexa Genome Analyzer has been the first example of short read sequencer [13]. It uses a bridge PCR to amplify the DNA fragments and an elongation process mediated by reversible dye terminators. At first, the template DNA is fragmented and the fragment ends are modified for attachment of oligonucleotide adapters. The DNA templates are denatured and deposited into a transparent slide on the surface of which are bound oligonucleotide anchors, complementary to DNA template adaptors. The bridge PCR amplification consists in the bending of anchor-coupled fragment that attaches the free end to an adjacent anchor olignucleotide, forming an arch. The adapters on the surface act as primers for PCR amplification. Multiple amplification cycles convert the single-molecule DNA template to an amplified *cluster*, each one containing about 1,000 clonal molecules. The reaction mixture for DNA synthesis and sequencing contains primers, the DNA polymerase and four reversible terminator nucleotides each labeled with a different fluorescent dye. After incorporation into the DNA strand, the nucleotide fluorescence is detected by a CCD camera. Therefore, the terminator group at the 3′ end is removed from the base and the synthesis cycle is repeated. A single run of Solexa produces >25 million 36 base-long reads (>1 billion bases) in 2.5 days. This technology can be provided with a *paired-end* module (50 bases-long reads) that is useful in assembly protocols.

The main Solexa features are bridge PCR and SBS based on reversible dye terminators

4.1.3 AB SOLiD

The main SOLiD feature are sequencing-by-ligation and color-space coding

The SOLiD (Supported Oligonucleotide Ligation and Detection) system is a short read sequencing technology based upon ligation [92]. The amplification procedure is carried out by emulsion PCR and is very similar to Roche/454 method. DNA fragments are ligated to adapters, bound to beads and clonally amplified with emPCR. At this point, DNA is denatured and the beads are deposited onto a glass support surface. The first step in the sequencing process is the hybridization of a primer complementary to the adapter at the adapter-template junctions. The primer is oriented to provide a 5′ phosphate group. This is necessary for ligation to oligonucleotides octamers, that consist of 2 probe-specific bases and 6 degenerate bases with one of 4 fluorescent labels. The 2 probe-specific bases is one of the 16 possible 2-bases combinations. In the first ligation step, octamer probes compete for the annealing to the template sequences immediately adjacent to the primer. After annealing, the octamer is ligated, the fluorescence signal is detected and a cleavage process is performed involving the last three octamer bases. These cleavage step removes the fluor and rigenerates the 5′ phosphate for a subsequent cycle. Seven cycles are accomplished for the first primer. Therefore, the synthetized strand is denatured and a new primer is annealed one position before the previous primer in the adapter. This procedure is repeated five times in order that each nucleotide is sequenced twice. However, the sequence output is not a nucleotide series, but is decoded in *color-space* by which one color (one fluorescent signal) corresponds to a couple of bases. Thus, there is the need to interpret the output, translating it into *base-space* sequences.
The SOLiD 3 system generates 30-60 gigabases and up to 1 billion reads per run in 10 days. The reads have length of 50 bases. As the Solexa system, there is the possibility to sequence paired-end libraries. Beyond the very high-throughput, SOLiD guarantees an high accuracy in read quality due to the double check of each nucleotide in the ligation step.

Table 4.1: NGS comparison. SBS: sequencing-by-synthesis, SBL: sequencing-by-ligation, py: pyrosequencing, dt: dye terminators, rev-dt: reversible dye terminators.

	Roche/454	**Illumina GA**	**AB SOLiD**	**Sanger**
sequencing	SBS	SBS	SBL	SBS
chemistry	py	rev-dt	ligase	dt
amplification	emPCR	bridge PCR	emPCR	cloning
read length (bp)	up to 500	up to 50	up to 50	800
run time	10h	2.5 days	10 days	3h
bases per run	500 Mb	1.5 Gb	up to 60 Gb	96 Kb

4.2 SHORT-READS ALIGNMENT

The large volume of data produced by NGS technologies present fascinating challenges for data management, storage and analysis. Therefore, new bioinformatic approaches and tools have to be developed for maximizing the advantages coming from these new technologies. The developing effort must be mainly directed to algorithms for the alignment of short reads to reference genomes, that is a key-step in the NGS analysis. In effect, the standard DNA alignment programs, such as BLAST or FASTA, are inadequate to align millions of short reads against a genome, while SOAP [67], ELAND, SHRiMP [88], ZOOM [69] are examples of software specifically developed and optimized for this goal. According to this tendency, an extremely sensitive, efficient and fast algorithm for aligning millions of NGS reads allowing gaps and mismatches has been developed at the CRIBI laboratory and named **PASS**[2] [23].

PASS: a program to align short sequences

PASS is based on the creation of a genome index, that is a gene structure containing the genome positions of all seed words (12 bases as default). After the genome index production, PASS tries to align each input read in three steps:

1. identification of the query seed words in the genome index.
2. check for possibility to extend the alignment in the seed flanking regions.
3. refinement of the alignment with a modified Smith-Waterman algorithm.

In particular, the alignment extension uses a simple but effective approach that allows an immediate analysis of the flanking regions adjacent to seed words (Fig. 4.1). It makes use of Precomputed Score Tables (PST) of all the possible short words aligned against each other. The length of these short words can vary between 6, 7, 8 bases, forming several PSTs. The score of each alignment is computed using Needleman and Wunsch algorithm, using different values for matches, mismatches and gaps. These PSTs are already created and are supplied together with PASS. They are loaded in RAM allowing fast execution time. Thus, when an input read finds a seed word in a genome region, PASS verifies the possibility to extend the seed analyzing the flanking regions with PST. If these scores are higher than a pre-defined threshold, the input read passes to the last step by which PASS performs an exact dynamic alignment of a narrow region around the match. In addition to PST, PASS applies low-complexity and AT rich regions filters.

Pre-computation of all possible n-mers aligned against each other allows a rapid evaluation for alignment extension

PASS is able to align all NGS sequences in base-space and color-space and supplies modules for paired-end alignments, SNP and IN/DEL detection and spliced alignments.

2 http://pass.cribi.unipd.it/cgi-bin/pass.pl

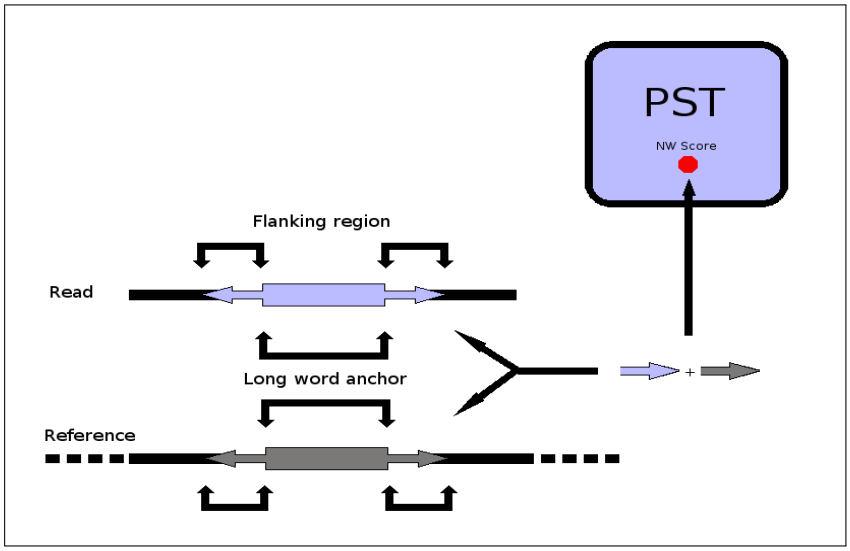


Figure 4.1: PASS alignment extension. The flanking regions adjacent to seed word are rapidly scored using a PST matrix.

4.3 RNA-SEQ ANALYSIS

The sequencing of whole transcriptome libraries with NGS methods offers a great opportunity to validate genome annotation, thanks to NGS ability to detect low-level transcripts and sensitivity to define the boundaries of exons, utrs and genes.
The CRIBI laboratory has a strong knowledge in genome research and long experience in genome sequencing projects. Their resources consist of several Sanger sequencers, a Roche/454 system and, more recently, a SOLiD System Analyzer v3. One of the first utilization of SOLiD technology was the sequencing of two transcriptome libraries coming from root and leaf organs of *Vitis vinifera*. This run produced about 150 millions of 35 base-long reads coded in color-space.

Leaf and root transcriptome libraries sequenced with SOLiD were used to evaluate prediction releases

Taking advantage of the large amount of transcriptome short reads and the availability of an efficient software for their alignment, the reads alignments were used to evaluate the two prediction releases, v0 and v1, and to establish what prediction model is more close to reality. To do that, the read distribution along the genome was analyzed together with the percentage of predicted splicing sites confirmed at least by one read in each prediction version.

4.3.1 Coverage distribution

To analyze the read distribution, 150 millions of 35 base-long reads were mapped on the genome using PASS, allowing 2 mismatches, 0 gaps and using the best-hit alignment option. Then, the genome was divided in four categories **utr**, **cds**, **intron** and **extragene** according to the v0 or v1 coordinates and the number of bases covered by short reads was counted for each category. Theoretically, due to the transcriptomic nature of reads, the map-

Transcriptome data should cover nucleotides classified as CDS or UTR

ping data should show a great coverage of *cds* or *utr* regions, and a low coverage of *intron*, probably due to unspliced mRNA, and *extragene*, probably unannotated genes.
The mapping results for v0 and v1 are summarized in table 4.2. The table shows in the first column prediction data, as percent-

Table 4.2: Coverage comparison.

	Annotation	*Coverage*		
		leaf	**root**	**leaf+root**
v0				
utr	1.87	48.40	42.46	57.78
cds	6.16	47.71	47.11	59.86
intron	26.95	9.69	12.25	17.03
extragenic	65.01	3.85	5.11	7.18
v1				
utr	1.45	52.77	45.47	61.83
cds	6.75	49.95	49.15	62.44
intron	23.44	8.68	11.36	16.04
extragenic	68.35	4.07	5.38	7.53

age of whole genome bases classified as *cds*, *utr*, *intron* or *extragene*. In the second column, there are the percentages of bases covered by reads coming from leaf, root or both libraries. The transcriptional landscape emerging from table data outlines a v1 clear increase of *cds* and *utr* coverage, a decrease in *intron* coverage and a small increase in coverage of *extragene* portions. This tendency is confirmed for all library combinations. This results in a higher *specificity* and *sensitivity* (2-3 percentage points) of v1 prediction[3]. The *utr* and *cds* coverage distance to 100% is due to constitutive errors in predictions, but, above all, to library limitations, since a transcriptome library coming from one single organ can not cover the entire gene set.
The differences in transcriptional landscape are better observable in the fig. 4.2 and 4.3. These *saturation curves* outline the covered nucleotide number for each genome category as a function of the number of mapped reads. A screenshot of the genome coverage at intervals of 10 millions read alignments was taken. The last right-hand interval represents the mapping of the entire read set. It need to consider that the discrepancies between the number of input reads (≈150 millions) and the number of final alignments (≈122 millions) are due to the quality filter applied by PASS in the alignment step.
In the *saturation curves*, a further category called *near*, representing the 300 bases upstream and downstream of the genes, was

3 *specificity*: $\frac{\text{covered(CDS+UTR)}}{\text{total(CDS+UTR)}}$; *sensitivity*: $\frac{\text{covered(CDS+UTR)}}{\text{total\_covered}}$.

added. This class was introduced to capture the mis-annotated *extragene* regions, that likely are *utrs*.

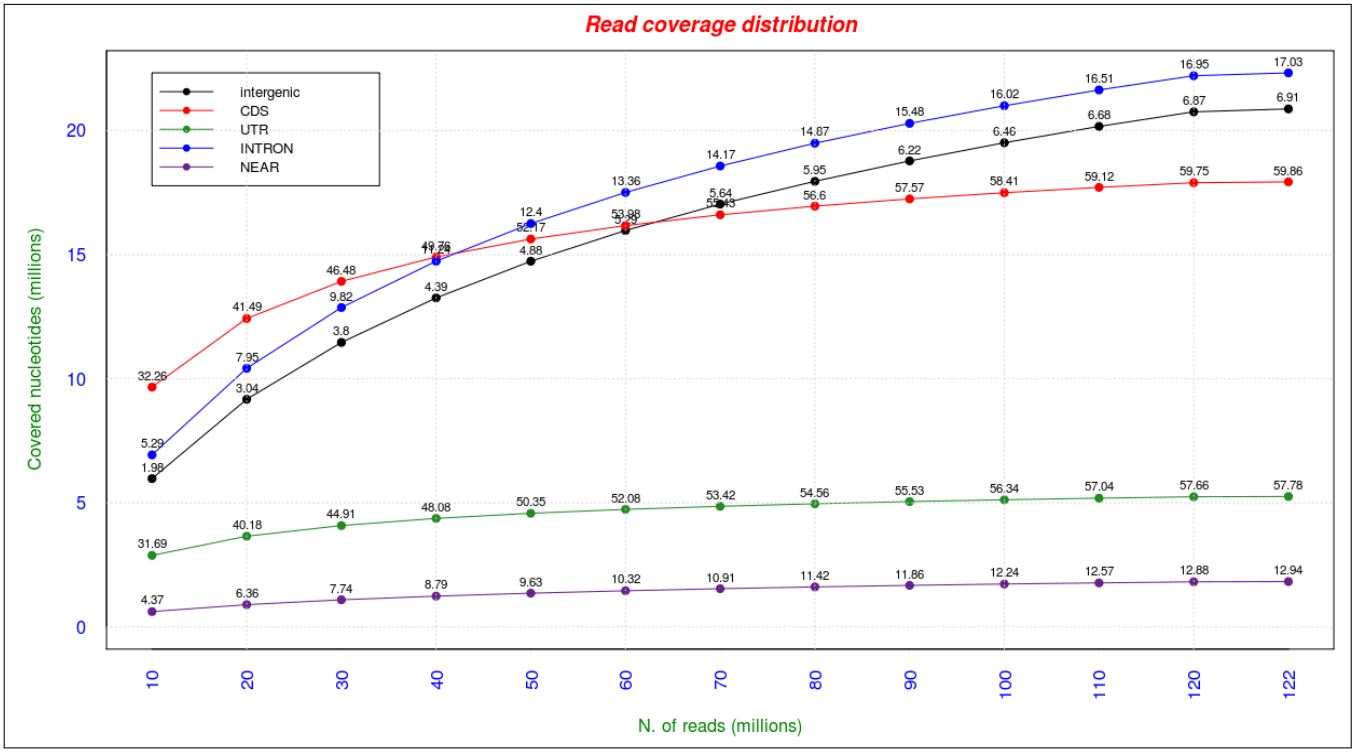


Figure 4.2: Coverage saturation curves for v0 prediction. CDS (red), UTR (green), INTRON (blue), EXTRAGENE (black), NEAR (purple).

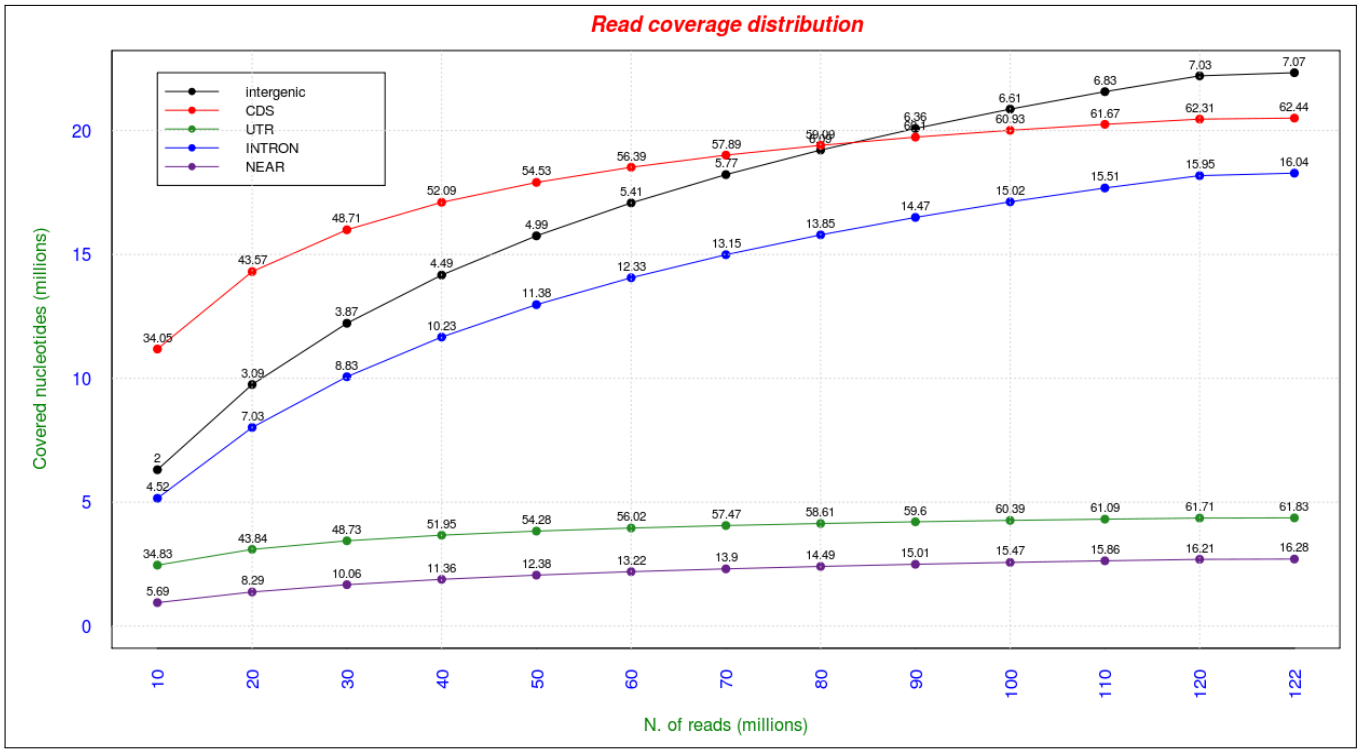


Figure 4.3: Coverage saturation curves for v1 prediction.

At first, it is worth to note that the number of covered bases reached a plateau, indicating that a grater amount of mapped reads would not much affect the coverage distribution and that in the final step an *equilibrium* is achieved. According to this, a single organ transcriptome would cover about half of the whole gene set.

The saturation curves demonstrates the v1 decrease of intron and the parallel increase of CDS and extragene covered nucleotides

The most important observation regards the *intron* class: the v0 number of covered intron nucleotides dramatically decreased in v1, being replaced by a correspondent parallel increase of *extragene* and *cds* nucleotides. Otherwise, the decrease of absolute number of *utr* bases in v1 is balanced by a correspondent increase of the nucleotides classified as *near*, indicating that the v1 *utr* loss is caused by the prediction of shorter utrs.
All these observations show that v1 prediction release is more close to reality, covering a greater percentage of *cds* and *utr* and a lower percentage of *intron*. Moreover, the data support the v0 and v1 prediction models described in the section 3.3.1. In particular, they offer a strong evidence for the *gene fragmentation* in v1 and *exon crumbling* in v0 (Fig. 4.4 A,B). Indeed, the first phenomenon is explained by the decrease of *intron* and the increase of *extragene* covered nucleotides; in the exon crumbling phenomenon, the coverage of a region, that results from the merging of two adjacent exons in a longer one with the parallel bridge-intron disappearance, causes the growth of covered *cds* nucleotides and the decrement of covered *intron* bases.

Extragene covered nucleotides could represent unannotated genes

However, both prediction releases demonstrate some critical aspects. At first, the great amount of covered *extragene* nucleotides need to be futher investigated: since, besides highlighting transcribed regions not detected in the prediction stage (unannotated genes), they could also be false positives due to random alignments, repetitive elements, ncRNA or pseudogene regions. A first attempt to analyze these *extragene* covered regions revealed that most of the covered nucleotides are grouped in clusters, excluding random alignments.

Intron coverage could stand for unpredicted splicing variants or transposons

Secondly, a possible explanation for *intron* coverage is the presence of unspliced mRNA in the starting libraries or unpredicted splicing variants. However, a great number of covered intron islands were noticed: within a predicted intron it is quite possible to find an extended SOLiD signal in both strands (Fig. 4.4 C). A further analysis showed that a common feature of these regions is the presence of integrase or retrotranscriptase domains, thus indicating putative transposable element domains [52].

4.3.2 Splicing site evaluation

Transcriptome reads corresponding to exon junctions can hardly be mapped on the genome

In the last paragraph, the grape genome was used as the reference database for mapping the transcriptome short reads. However, by this way it is not possible to map the reads straddling the splicing sites, because adjacent exons that forms a transcript are placed in distant regions at genome-level. The reads that correspond to the exon boundaries can not be matched in the

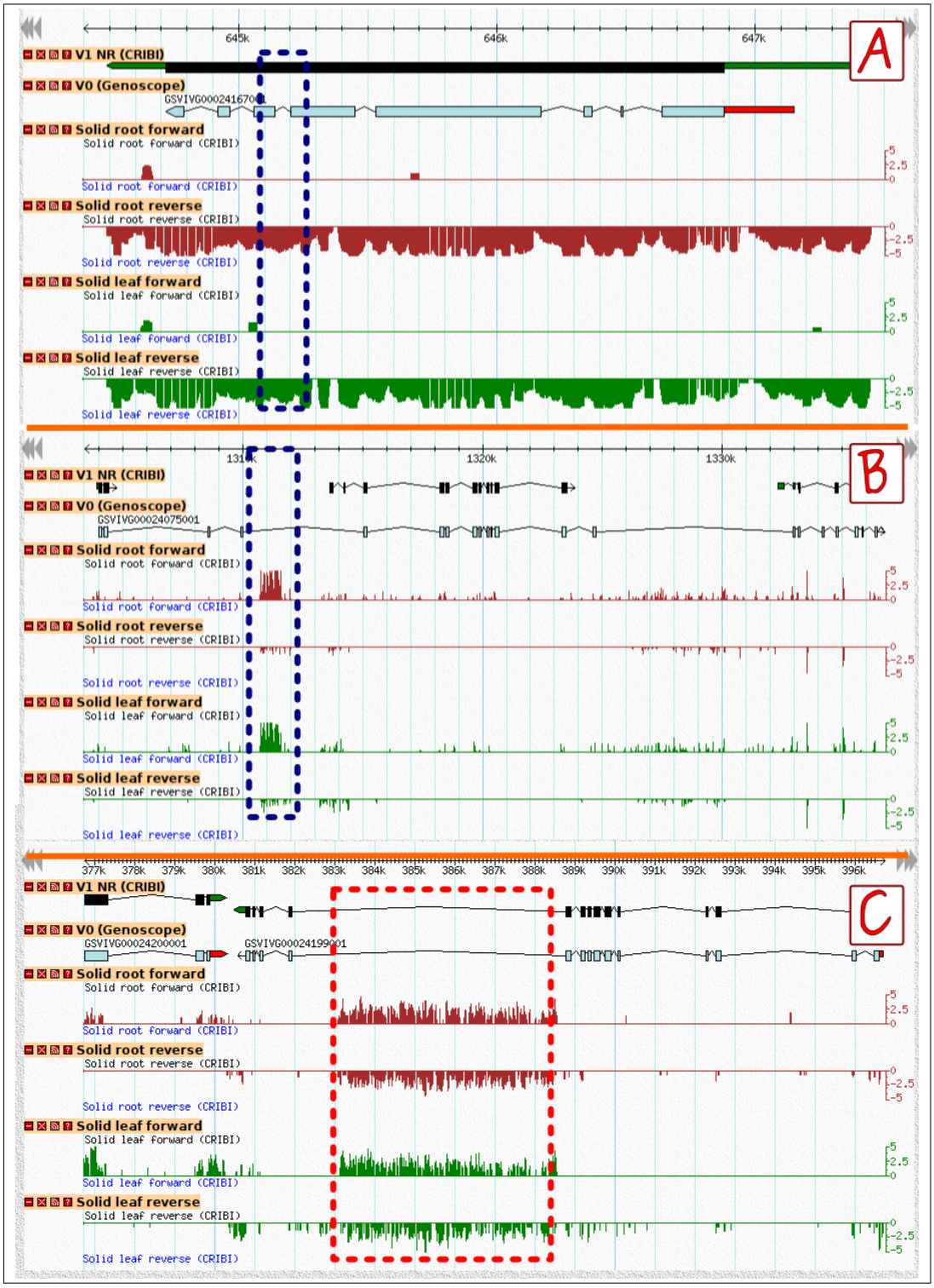


Figure 4.4: Genome region with v1 (black), v0 (light blue) prediction and SOLiD transcriptome evidences for leaf (green) and root (red). In the figures A and B, the SOLiD data seems to agree with v1 model, highlighting the exon crumbling in v0 and gene fragmentation in v1 (dotted blue square). In the figure C, the red dotted square shows a SOLiD signal evidence within a predicted intron.

genome or they can be mis-placed in uncorrect genome regions. To diminish the presence of false positives or false negatives, the genome database was integrated with sequences of 60 bases that recreate all the splicing sites at transcript-level. The 60 bases are formed by 30 nucleotides upstream the *donor* site and 30 nucleotides downstream the *acceptor* site. By this way, using best-hit alignments, uncorrect mapping data can be decreased without loss of information.
Three types of splicing sites have been modeled, allowing possible alternative splicings (Fig. 4.5):

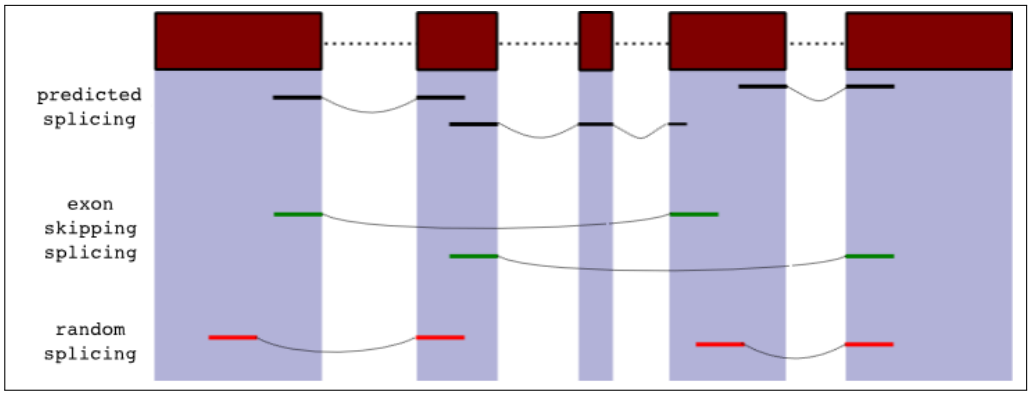


Figure 4.5: The dark red block connected with a dotted line is an hypothetical gene model. Below, there are the three types of splicing sites used for 60 base-long sequences construction: *predicted splicing*, *exon skipping splicing*, *random splicing*.

predicted splicing: the splicing sites are built based on the precise order of predicted exons.

exon skipping splicing: the splicing sites are recreated through all possible combinations of predicted exons, maintaining the exon order and rejecting the *predicted splicings*.

random splicing: to model splicing sites that do not involve *donor*/*acceptor* pairs already individuated in the prediction (as inner splicing), it is necessary to search the genome for *de novo* splicing sites. Genesplicer [85], a tool for splicing site prediction, was used to build novel *donor*/*acceptor* possibilities, filtering for direction, order and distance criteria ($5 \leqslant (\mathtt{acceptor} - \mathtt{donor}) \leqslant 10,000$).

By this way, about 30 millions of 60 base-long sequences representing about all the splicing sites possibilities were obtained. These sequences came from *predicted splicings* and *exon skipping splicings* of v0 and v1 prediction releases in addition to *random splicings* found along the genome. The mapping of short reads pointed out about 5 millions of alignments in these sequences.
These data offer a valid resource for the evaluation of v0 and v1 prediction. Indeed, it is possible to count and compare the number of splicing sites confirmed by at least one read in the two predictions. The results are summarized in the table 4.3. The data show that v1 predicts a smaller number of splicing sites,

v1 seems to have a greater specificity value

as expected. Moreover, v0 seems to have more splicing sites confirmed as absolute number. However, the situation changes when the percentage values are considered: v1 has a greater *specificity*, demonstrating that a great number of predicted v0 splicing sites are likely wrong.

Table 4.3: Splicing site comparison. (*) only the covered genes are considered for computing the total number of splicing sites.

	leaf	root	leaf+root
v0			
total predicted (*)	100,824	101,282	109,212
covered	54,551	51,969	67,945
covered (%)	54.11	51.31	62.21
v1			
total predicted (*)	90,262	90,881	97,362
covered	53,085	50,700	65,939
covered (%)	58.81	55.79	67.73

To better define the splicing site prediction quality, an analysis based on a gene-level approach was performed. For each i gene, the percentage of confirmed splicing sites over the total of predicted ones were computed: $p_i = confirmed\_ss_i / total\_ss_i * 100$. Then, the percentage values were divided in ten categories, at intervals of 10 p.p., and the distribution of genes according to their p_i was computed. The results (Fig. 4.6) showed that 67% of v1 genes have more than 60% of splicing sites covered at least by one read, compared with 59% of v0. This analysis seems to further support the v1 prediction model, although the differences with v0 are not dramatic.

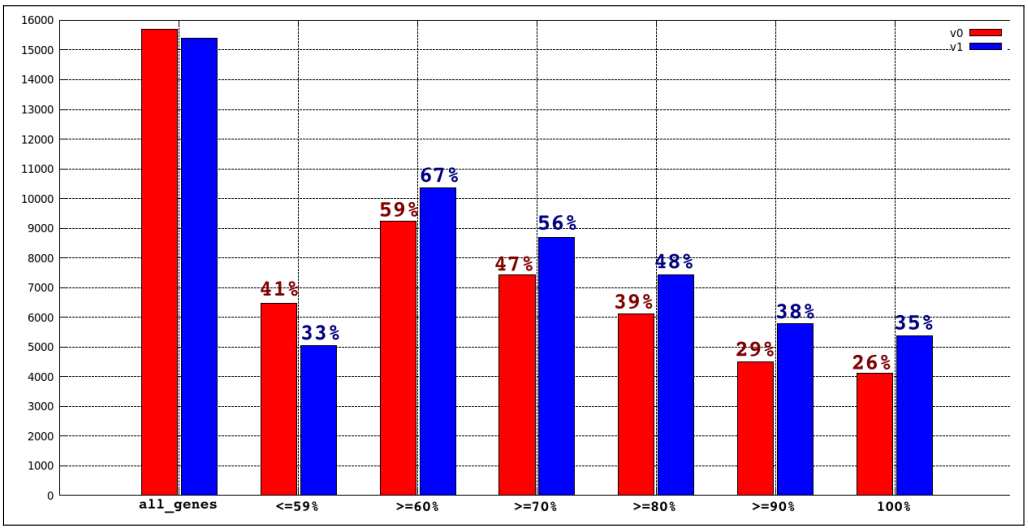


Figure 4.6: The histogram represents the number of genes with different percentage values of splicing site coverage. The colors refers to different predictions (v0:red, v1:blue). The number at the top of bars represents the relative percentage value.

Part III

GENOME FUNCTIONAL ANNOTATION

5

GETTING THE INFORMATION

CONTENTS

The functional annotation is the action of characterizing the set of predicted genes, assigning them a biological function, a metabolic role or structural features. In other word, the annotation stage allows to compile a sort of "identity card" for each gene product resulting from the translation of the coding sequence. This kind of information is extremely useful for the subsequent fine analyses focused on the relevant gene families or genes involved in specific metabolic pathways.
In the world wide web a great amount of resources, tools and databases allowing to infer gene and gene products properties are available. The developed annotation platform pursued two main strategies, a *similarity approach* and a *predictive approach*. The former extracts information for inter-species sequence similarities. The latter uses software tools predicting structures or domains based only on sequence properties. Moreover, a module for gene clustering to group genes of the same genome was developed, able to form gene families and highlight intra-species evolutionary relationships.
It has to be noted that the annotation platform is an automated computational procedure, useful for seeking any type of large-scale functional evidence. For this reason, a manual review is advisable to refine data and discard poorly-confirmed or unreli-

able annotations.
In this chapter, an overview of the main annotation methods and the first annotation results for grapevine genome are presented along with a strategy to interpret functional data and to point out significant genes at genome-level.

5.1 SIMILARITY APPROACH

According to a similarity approach, functional information is collected based on inter-species sequence similarity data, assuming that regions highly conserved maintain the same functions or roles in different species. Since protein folding and function depends on protein primary structure, proteins sharing amino acidic sequence, or part of the sequence, probably have the same or correlated biological behavior. Starting from this assumption, the first thing to do with a functionally uncharacterized gene set is to search for sequence similarities with annotated proteins from other organisms and/or species.

Sequence conservation can highlight functional or structural constraints

This search can be carried out for similarities that span either the entire sequence or, small parts of the sequence or *domains*. Furthermore, some amino acids in specific non-adjacent positions are decisive and sufficient for the definition of the biological function. If global similarities can be searched with the usual alignment algorithms, as Blast, the situation is more complicated for elucidating domains or position patterns. To address these needs, different strategies have been pursued and several tools developed in the last years. All these tools exploit the availability in the world wide web of biological databases containing well-annotated protein sequences and domains, patterns and profiles.

It needs to be careful in a similarity-based annotation

The similarity-based approach raises two problems, the error propagation due to the continuous transfer of annotations among different organisms, and the different function of proteins with high similarity. The first case occurs when the annotations have been not manually curated and undergone many transfers from genomes to genomes. By this way, some falsely characterized cases can "infect" the ongoing annotation transfers [51]. For instance, an annotating protein A could be annotated with functions coming from a protein B, which inherited the annotation from a protein C, etc. If the similarity between B and C is questionable, protein A will acquire dubious annotations.
The second problem occurs when proteins in spite of sharing a high sequence similarity display different biological functions because of the presence of small specific domains that are fundamental for differentiating their biological localization, structures and physiological role.
Both problems are implied in automated computational methods, that have no direct control over the punctual annotation, although conservative annotation parameters are adopted. Indeed, the choice of more or less stringent cutoff values or different protein databases can heavily influence the annotation sensitivity.

To avoid mis-annotations due to one of the above described problems, the platform annotates gene products according to different independent methods, ranging from orthologous proteins to protein domains.
In this scenario, it becomes clear that a manual review process with experimental data support is fundamental for a complete, reliable and high-quality functional annotation. However, it is worth to note that such a process is extremely time-consuming at genome-level. Automated computational procedures represent the first tier of the functional annotation stage, giving a draft rather than the final annotation book.

A manual review is fundamental to obtain high-quality annotations

5.1.1 Biological databases

Nowadays, an huge amount of biological data have became available thanks to the development of molecular biotechnology techniques and sequencing methods. This continuous production of nucleotide or amino acids sequences requires appropriate repositories where these biological data can be stored and made accessible for the scientific community. For this reason, in the last years there has been an enormous increase of biological databases, that are hosted and maintained by different research centers spread around the world. They can be classified in several categories according to the data type, that range from nucleotide or protein sequences to protein domains, from motives and profiles to transcription factors binding sites, etc.
In this paragraph, the attention is focused on the description of a protein database, that was used as reference by the annotation platform in the preliminary annotation step. This database, called UniProtKB [3], represents a comprehensive, high-quality and freely-accessible resource of protein sequences and is provided with rich functional information. The other well-known protein database is the NCBI nr [111], that greatly overlaps UniProtKB since it shares a similar set of sequence repositories and sources by which the sequences are derived [101]. To avoid redundancy, UniProtKB has been chosen for the assignment of preliminary annotations to the set of predicted gene products of *Vitis vinifera*.

UniProtKB is an universal resource of protein sequences

UNIPROTKB The UniProt Knowledgebase is a resource for the collection of functional information on proteins. These accurate and consistent data include disparate information as the amino acid sequence, the protein name and description, the taxonomy group, the biological ontologies and classifications, etc.
The protein sequences are derived from the translation of the coding sequences submitted to the public nucleic acid database INSDC[1], the EMBL-Bank/GenBank/DDBJ database, or they come from PDB database. They can also come from direct protein sequencing, sequences scanned from literature and sequences de-

Protein sequences mainly derive from CDS translation

1 International Nucleotide Sequence Database Collaboration

rived from CDS not submitted to INSDC.
UniProtKB is divided in two parts:

- **UniProtKB/Swiss-prot**: containing non-redundant manually annotated records supported by experimental data and with information extracted from literature and curator-evaluated computational analysis. In this section, proteins encoded by a same gene are merged into a single UniProtKB/Swiss-prot entry.

- **UniProtKB/TrEMBL**: containing unreviewed computationally analyzed records that have been obtained with large-scale functional characterization and that are awaiting full manual annotation.

In the annotation method implemented in the platform, a BLAST analysis against UniProtKB database is executed for each *Vitis vinifera* gene product. In particular, a filtered UniProtKB database is used, where a restriction on the plant taxonomic range limits the number of proteins to query. This restriction is important because it cuts the hits with no-plant proteins that could supply not appropriate functions to grapevine proteins, and it decreases the execution time, necessary to obtain blast results for whole gene set. In the Blast results, the annotation procedure selects all the hits that satisfy predefined criteria of identity, similarity, coverage and e-value and uses them as source for annotation. This approach results less conservative compared to the one that consider only the best-hit (which is the probable ortholog) as source of annotation. Such an approach evaluates also a series of differently scored hits and overcomes the problem that most of the time the best hit, that can be a putative or hypothetical protein, does not give any useful annotations. Instead, the second or third hit often gives more informative annotations. However, enough stringent cut-off values were selected for all the above listed criteria. At the moment the parameters are represented by an e-value of $1e^{-5}$, a percentage identity greater or equal to 30%, a similarity greater or equal to 60% and a percentage coverage greater or equal to 60% for both query and subject. Different parameters were evaluated and these ones seemed to be a good balance between stringency and sensitivity.

A series of protein matches passing predefined cutoff values are considered for annotation

5.1.2 Protein domains

Database searching is an useful tool to individuate in novel sequences structural and functional properties already found in other well-annotated sequences. However, sequence-based searching methods, e.g. Blast analysis against protein databases, are not sufficient to detect evolutionary and functional correspondences for two main reasons:

- these techniques are focused on the global sequence, searching for similarity evidences that span much nucleotides or

amino acids as possible. This kind of approach does not allow to find sequence motifs or modules that are conserved in the course of time despite of the global sequence divergence.

- alignment algorithms, e.g. Blast, compare two sequences and determine their similarity scores using standard substitution matrices and gap penalties, attributing one single score for each substitution of one amino acid with another, independently from the context. However, amino acids in specific positions may have different conservation patterns in different contexts. A solution for similarity searches could be represented by the usage of substitution matrices that reflect the amino acids frequencies in specific positions according to different gene families [31].

A working out of the sequence-based similarity methods is represented by the profile searching techniques. They are based on an higher-level similarity concept by which amino acids positions and frequencies are contextualized and the alignment matches are not necessarily determined by stretches of similar or identical adjacent amino acids. Moreover, these methods do not search for similarities on the entire sequence, but focus on the individuation of protein domains, motives and patterns.

Proteins found in nature are composed of structural and functional domains

Protein domains are structural or functional modules that compose proteins. The combinations of different domains produce the diverse range of proteins found in nature. The identification and analysis of specific protein domains can help in elucidating the function of the entire protein. Motives and domains can be coded (and stored in public databases) through simple regular expressions or through more complex forms as multi-alignments, profiles, position weight matrices (Fig. 5.1) or Hidden Markov Models. The set up of all these novel encoding systems have been paralleled by the development of appropriate alignment programs.

Profiles and profile-HMMs are used to model protein domains

Protein profiles or matrices derive from multiple alignments of members of protein families. In a multiple alignment, the most important functional or structural residues are extremely conserved among all the sequences of the multi-alignment. A profile summarizes the alignment information combining data from the critical conserved positions of the sequences, the substitution frequencies and the propensity of gap inserts. Another efficient method is represented by the usage of HMMs that define a probabilistic model describing and generalizing protein domains or families.

The underlying idea of these approaches is to compare a sequence to a statistical model that describes a family or pattern of sequences as opposite to a simple pairwise comparison of single amino acids. Thus, these methods facilitates and improves the search of distantly related sequences and the identification of conserved functional or structural domains.

In the developed platform, the predicted gene products are com-

pared with three popular databases of protein domains and patterns: Pfam [41], SMART [66] and Prosite [49].

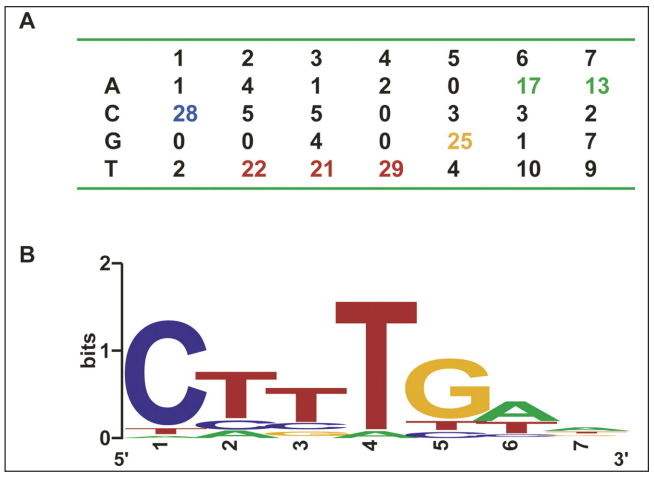


Figure 5.1: An example of a Position Weight Matrix for a nucleotide sequence [33]: horizontally there are the positions along the sequence and vertically there are the four possible DNA bases. Each number in the matrix defines the frequency of a base in a specific position. The most conserved nucleotide is highlighted for each position. In B, the PWM is visualized with sequence logo.

PFAM The Pfam database is a large collection of protein families having in common functional or structural domains. Each family is represented by multiple sequence alignments and Hidden Markov Models. Each Pfam entry is characterized by one of the following types a) *family*, joining proteins with the same domains, b) *domain*, defining a structural unit present in different families, c) *repeats*, representing short units present in multiple copies in globular proteins, d) *motifs*, consisting of short units outside globular proteins. Moreover, Pfam provides also a *clan* classification, grouping related families according to similarity of sequence or profile-HMM.

The Pfam database is divided into Pfam-A and Pfam-B. Pfam-A entries are derived from the most recent release of UniProtKB and family groups are detected using profile-HMM searches. For each Pfam-A family there is a curated seed alignment, formed by the most representative members of the family, profile-HMMs generated from seed alignment and an automatically produced full alignment containing all the family proteins (Fig. 5.2). Otherwise, Pfam-B entries are automatically generated, unannotated and of lower quality, but useful to identify conserved regions where Pfam-A entries fail. Actually Pfam database contains 11,912 families.

A software implementation that compares query sequences against profile-HMM libraries is HMMER [38]. It assigns to the comparison a score that represents the probability for the sequence to be related to the given model.

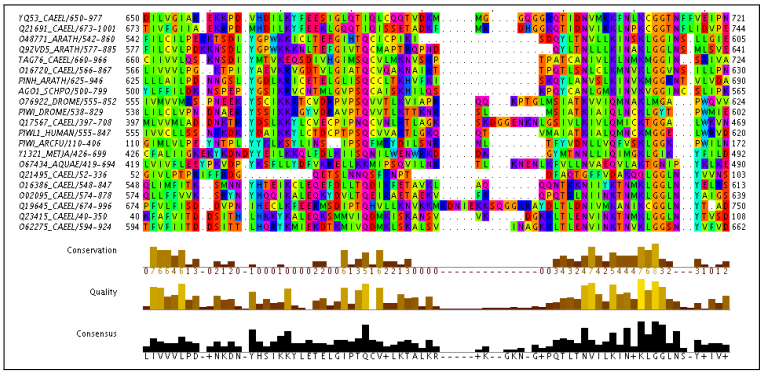


Figure 5.2: A Pfam multi-alignment from the Pfam web-site.

SMART The Simple Modular Architecture Research Tool (SMART) is a web resource that collects data for the identification and annotation of protein domains and the analysis of protein domain architectures. This collection is characterized by annotation quality and completeness. It can be used by two ways, according to the kind of underlying protein database, that can be *normal*, containing Swiss-Prot, SP-TrEMBL and stable Ensembl proteomes, and *genomic*, representing only proteomes of completely sequenced genomes (Ensembl for metazoans and Swiss-Prot for the rest). At present, *normal* SMART contains manually curated models for 784 protein domains and *genomic* SMART contains proteomes for 630 genomes. As Pfam, SMART library consists of alignments, profiles and profile-HMMs and can be inspected by HMMER software.

PROSITE PROSITE is a database of protein families, domains and functional sites. They are determined by *in silico* analysis or literature-confirmed experimental evidences and are provided with rich descriptions and annotations about structures and functions. Moreover, each PROSITE entry is classified by its reliability and confirmation level. The data contained in PROSITE database are represented by patterns and profiles. Profiles are the same position-specific weight matrices or profile-HMMs already used in Pfam and SMART database. Patterns are short amino acidic signature defined by a grammatical syntax similar to that of regular expressions (Fig. 5.3).
PROSITE currently contains patterns and profiles specific for more than a thousand protein families or domains. The ScanProsite software program is used to compare a sequence against the PROSITE database.

5.1.3 Metabolic pathways

The functional annotation methods above described have been focused on the characterization of the functional or structural properties of a single gene or gene product, representing the ultimate goal of annotation procedures. However, the biological

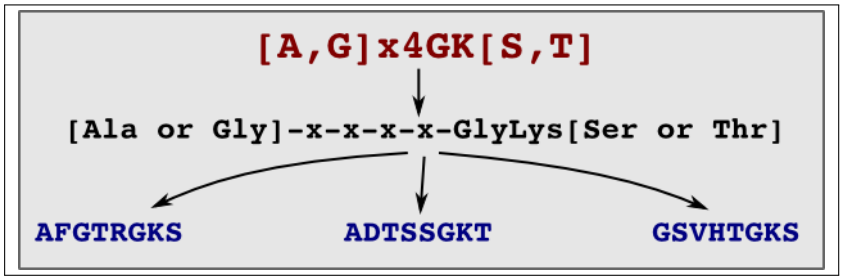


Figure 5.3: An example of PROSITE pattern (red) and its translation for a better explanation (black). Xs mean that any amino acid is allowed. Square brackets indicate possible alternatives. In blue there are short signatures recognized by pattern.

Gene is a simple unit and makes part of biological systems and networks

function of the living cell is a result of many metabolic pathways and interactions between molecules and proteins, and it can not be attributed to a single gene or gene product. Therefore, the genes are simple units that acquire a biological mean thanks to the interactions with molecules or products of other genes and thanks to their role in specific metabolisms. Genes are important as part of higher-level systems or networks and different biological contexts determine their expression or repression, and so the realization of their functions. Thus, in genome annotation projects the determination of metabolic roles and identification of localization in biological networks of gene products become extremely important.

KEGG (Kyoto Encyclopedia of Genes and Genomes) [55] is a database resource for understanding high-level functions of the biological systems from genomic and molecular information. It gives a computational representation of biological systems, consisting in building blocks and wiring diagrams necessary for simulation, modeling and data analysis. The KEGG integrated database consists of 16 databases classified into systems, genomic, and chemical information: systems information represents functional aspects of the biological systems, e.g. *pathway* database, while genomic and chemical information represents the molecular building blocks of life, e.g. *genes* or *enzyme* databases.

KEGG annotation results from association between genes, EC numbers and metabolic pathways

In the annotation platform, a module that makes use of KEGG databases has been developed to associate predicted genes to metabolic maps, via enzyme targeting. In particular, for *Vitis vinifera* annotation, the reference database was the proteins contained in the KEGG *genes* database. This database was filtered to exclude no-plant proteins. Then, a Blast analysis was carried out using the grapevine predicted proteins as query and the KEGG plant proteins as reference database. The parameter setting was identical to the UniProt configuration (see above). The association of grapevine gene catalog with pathway information was achieved by the EC numbers associated to KEGG protein hits: the predicted genes inherited the EC numbers (if present) that denote specific enzymes participating to specific metabolic pathways. Moreover, the $genes \rightarrow EC \rightarrow map$ associations were graph-

ically transposed using KEGG pathway diagrams, formed by series of nodes (e.g. enzymes) and edges (biological relationships). The figure 5.4 shows an example of this graphical representation that highlights the annotated grapevine genes (enzymes) for the gluconeogenesis pathway.

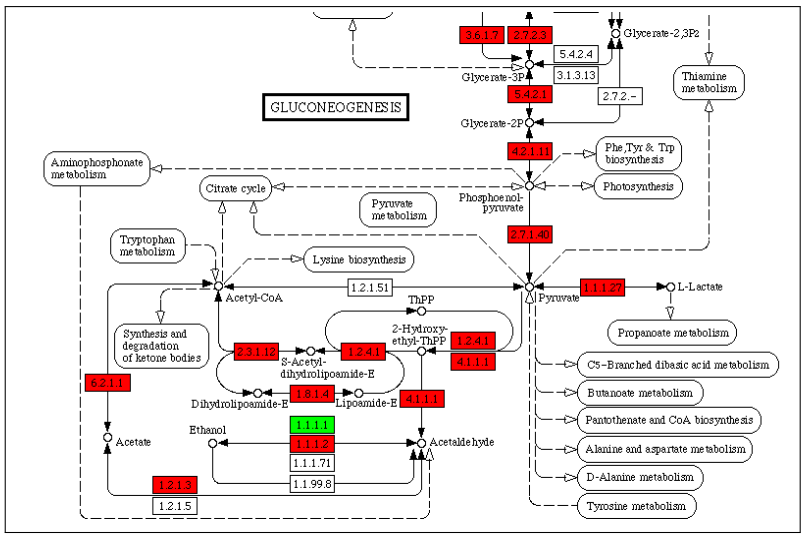


Figure 5.4: A typical KEGG representation of metabolic pathways. Nodes are represented by blocks (gene products) and circles (chemical compounds). Edges represent molecular interactions or relations.

In the next future, KEGG annotations will be integrated with expression profiles. The idea is to code the expression values coming from RNA of different tissues and organs sequenced with NGS technologies by using different colors. This would allow to further investigate and rapidly interpret the expression profiles in different tissues, in relation to specific metabolic maps.

5.1.4 Gene Ontology

There is the need of standard vocabularies

In annotation projects, a great problem is represented by the gene or protein nomenclatures. Frequently, the annotation procedures assign to genes the gene symbols or descriptions inherited from proteins used for annotation, e.g. UniProtKB hits. This habit arises some issues because there are cases where the same gene has different names in different organisms (e.g. caused also by typographical errors) or where genes with the same name have different functions [51]. In this way, the database searches by *gene name* criteria can have an high rate of false positives. To avoid problems deriving from mis-annotations, methods defining standard descriptions of biological functions or processes have been developed. By this way, the annotations can be transferred among organisms and biological databases without disfiguring the annotation information.
To address these needs, the Gene Ontology (GO) Consortium

(GOC) [9] has developed a controlled and structured vocabulary to describe biological properties for each gene or gene product, assuming that a large fraction of genes, and the related biological functions, are shared by organisms and/or species.

Gene Ontology provides a controlled vocabulary for describing gene products

The Gene Ontology consists of three vocabularies describing genes and gene products, the cellular component, the molecular function and the biological process ontologies. A gene product might be associated with or located in one or more cellular components; it is active in one or more biological processes, during which it performs one or more molecular functions.
All the biological concepts are defined by terms, identified by an unique numerical ID (e.g. GO:0006915). GO is implemented as a Directed Acyclic Graph (DAG) allowing multiple parent terms for each child term. From the top moving down to the bottom of the DAG, the terms (or nodes) become more specialized. At the end of the path the terms are called leaves. GO terms are linked by three relationships: *is_a*, *part_of* and *regulates*. A particular protein can be associated with more than one node within the three ontologies, reflecting the fact that it may function in several processes, contain domains that carry out different molecular functions and be localized in different cell compartments.
GO terms were assigned to grapevine gene products using data produced by similarity-based methods: protein database searching (UniProtKB), protein domains identification (Pfam, SMART and Prosite) and enzymes associations (Kegg). A platform module realizes the GO associations based upon two different data sources provided by Gene Ontology Consortium:

- *annotation file*: the GOA[2] group provides high-quality GO annotations to proteins in the UniProtKB, generated by means of a combination of electronic and manual techniques [12].

- *mapping files*: mappings between GO terms and concepts from other databases, e.g. Pfam domains, Kegg enzymes, etc.

Genes are annotated with GO terms by means of inheritance process

At first, the procedure starts collecting the results of the blast analysis of query grapevine genes against UniProtKB database. The protein hits, necessary for gene annotation, are frequently associated with GO terms. These associations are provided by GOA annotation file. Thus, each gene product inherits the GO terms associated with the related protein hits. In the second step, the obtained GO terms are enriched with the annotation coming from other resources. Pfam or SMART domains, Prosite patterns or Kegg enzymes can be mapped, using the available mapping files, to correspondent GO terms. Therefore, gene products inherits further GO terms from the associated domains or enzymes, only if the GO terms represent novel annotations.
Finally, the GO annotation module envisages the assignment of

2 Gene Ontology Annotation

GO slim terms. GO slims are cut-down versions of the GO ontologies containing a subset of the terms in the whole GO. They give a broad overview of the ontology content without the detail of the specific fine grained terms. For *Vitis vinifera* genes, the plant GO slim provided by TAIR[3] was used.

5.1.5 Plant Ontology

Plant Ontology describes plant structures and developmental stages

The Plant Ontology (PO) Consortium (POC) [102] shares with GO the need to describe biological properties in a structured and standardized manner. However, the PO differences are relevant because it focuses on plant species and on different biological aspects as plant structures and developmental stages. Indeed, in plants the nomenclature used to describe anatomy and development varies across taxa. Hence, its main objectives are to provide a shared language to describe tissue-specific gene expression and phenotype information in plant databases and in literature. The Plant Ontology vocabularies cover two biological domains: *plant structure* and *plant growth and development*. Plant structure terms describe the morphological and anatomical structures of whole plants, including organs, tissues and cell types, e.g. stamen, parenchyma, guard cell, etc. Growth stages terms describe whole plant growth stages and plant structure developmental stages, e.g. seedling growth, leaf development stages, germination, etc.
PO has the same DAG structure of GO and the relationships between terms are a) *is_a, type_of* (e.g. a *silique* is a type of *fruit*), b) *part_of* (e.g. *pericarp* is part of the *fruit*) and c) *develops from* (e.g. a *leaf* develops from a *leaf primordium*).
In the TAIR website a PO annotation of *Arabidopsis thaliana* proteins is provided. To annotate grapevine gene products with PO terms, *Arabidopsis* proteome was considered as reference database for a blast analysis. In a procedure similar to that used for GO annotations, grapevine gene products exploit sequence similarities to inherit PO terms associated to *Arabidopsis thaliana* protein hits.

5.2 PREDICTIVE APPROACH

In the last sections, methods for the functional characterization of gene products based on inter-species similarity evidences and database searching procedures have been described. However, there are different computational techniques that mainly focus on the composition of protein sequence itself for functional assignments. These techniques strongly resemble to *ab-initio* software tools, using parameters obtained by training procedures and avoiding the usage of databases. In particular, a great importance is given to specific sequence portions and physical-

3 http://www.arabidopsis.org/

chemical properties of amino acids that make up the protein. Such type of approach refers to software tools suitable for the identification of specific sequence features such as targeting peptides or transmembrane domains.

Ab-initio methods are adopted to predict secondary structure or cellular localizations

However, the distinction between similarity-based and *ab-initio* approaches are not so evident, because frequently some *ab-initio* methods use homology evidences to improve their results and the employed algorithms are similar to those used in identification of protein domains.

5.2.1 Protein targeting and cellular localization

An important role for the functional annotation of genes is played by the identification of signal peptides. The importance of signal peptides derives from the discovery that proteins have intrinsic signals governing their transport and localization in the cell, determining the protein targeting. Indeed, protein targeting or protein sorting is the mechanism by which a cell transports proteins to the appropriate locations into the cell or outside of it where they can perform their tasks.

Protein targeting drives proteins to their correct localization

A signal peptide is a short peptide chain frequently placed at the N-terminus and the correctness of the sorting signals becomes fundamental for the cell life because errors can lead to biological disorders or diseases. The targeting motives can be cleaved by signal peptidase or maintained in the mature protein after reaching the final destination.

Targeting sequence motives can be divided into secretory signal peptides (SP) and transit peptides (TP) [39]. The secretory signal peptide targets a protein for translocation across the endoplasmic reticulum (ER) membrane. Transit peptides refer to targeting signals for chloroplasts and mitochondria. Otherwise, global sequence properties as amino acid composition can define proteins of different subcellular compartments.

Secretory signals, transit peptides, transmembrane domains and global sequence properties drive the identification of protein localization

A large class of proteins is also represented by integral membrane proteins. The sequence of transmembrane proteins have one or more hydrophobic domains that cross the membrane once or several times.

Experimentally identifying protein targeting signals is cost intensive and time consuming. For this reason, in the last years numerous methods for prediction of signal peptides, transmembrane domains and subcellular localizations have been developed. These methods are based on amino acid composition of the protein, specific sorting signals or targeting sequences contained in the protein sequence, or homology search in databases of proteins with known localization. However, the annotation platform envisages the usage of sequence-based tools, that do not require database searching procedures and predict subcellular localizations through sequence properties.

Prediction methods differ in the employed algorithms, ranging from weight matrices to machine learning techniques as k-nearest neighbor methods, support vector machines (SVM), neural net-

works or Hidden Markov Models. However, no prediction method is able to cover all the different types of signal.
There are two significant advantages of using prediction methods based on sequence composition for the annotation of grapevine gene catalog: a) signal predictions can be obtained also for incorrect gene predictions, lacking N-terminal residues, and b) there is the possibility to predict localizations for which sorting signals are not known or not well defined.
In the annotation platform, WoLFPSORT [47] and TargetP [40] are the software tools used for identification of protein cellular compartments, and TMHMM [64] and HMMTOP [105] are the programs that annotate transmembrane domains. The combinations of similar programs was adopted to obtain more reliable predictions.

The combined usage of similar programs increases the prediction reliability

WOLFPSORT It is an extension of PSORTII, it uses the PSORT localization features, some iPSORT parameters and amino acid composition. For the final classification, it adopts a k-Nearest Neighbors method. It contemplates the possibility of a dual localization prediction.

TARGETP It predicts the cellular location of eukaryotic proteins, based on the presence of any of the N-terminal signal: chloroplast transit peptide (cTP), mitochondrial targeting peptide (mTP) or secretory pathway signal peptide (SP). In addition, it provides the cleavage site location. TargetP integrates SignalP and ChloroP tools for the identification of SP and cTP respectively. It uses a combination of Neural Networks, Hidden Markov Models and weight matrices techniques.

TMHMM It is a transmembrane α-helix predictor, based on a HMM approach. It provides also the transmembrane topology, that is the in/out orientation of helices relative to the membrane. The implemented model describes various regions of membrane proteins: helix caps, middle of helix, regions close to the membrane, and globular domains. HMM approach is well suited for transmembrane prediction because it integrates several important features as hydrophobicity, charge bias, helix lengths, and grammatical constraints.

HMMTOP It provides a topology prediction of helical transmembrane proteins, using an HMM that computes the topology with highest probability among all the possible topologies of a given protein. It is based on the assumption that the transmembrane domains are determined by the differences in the amino acid distributions in various structural parts of the protein rather than specific amino acid compositions in these parts.

5.3 GENE FAMILIES

A common feature shared by many genomes is the presence in the DNA sequence of gene groups characterized by an identical or similar sequence. These groups are named **multigene families** and derive from gene duplication events in the course of evolution.
Gene groups can be divided in *simple* (or *classical*) and *complex* multigene families [17]. The *simple* class refers to gene clusters where all the members have identical or nearly identical sequences. The presence of multiple copies of the same gene implies the strong action of a conservative evolutionary process that allows limited substitutions in all gene copies. Such behavior is typical for genes or gene products required in great abundance, e.g. rRNA genes.

Members of simple gene families have nearly identical sequences

The *complex* class refers to gene families that contain members similar in sequence denoting a common evolutionary origin but sufficiently different to have distinct properties. The duplication and the following specialization events caused by evolution may indicate the need to accomplish similar tasks in different developmental stages or tissues, or the competitive action for the same biological process. The *complex* gene families can be clustered but also dispersed around the genome.

Members of complex gene families may be characterized by different functional or structural properties

Sometimes, there can be sequence relationships between different families forming *gene superfamilies*.
The importance represented by gene families in the post-genomics era required the development of a platform module for the individuation and description of gene families at genome-level. This module consists of a comparison between genes of the same organism (in this case *Vitis vinifera*), a grouping step according to sequence similarity criteria and a final description of the evolutionary relationships between members of the same group by means of evolutionary trees. In practice, the developed module is formed by three levels:

1. *clustering*: gene groups were identified comparing protein sequences of the grapevine gene catalog. The utilized clustering algorithm was CDHIT [68] that grouped the sequences with 90% identity of the global alignment.

2. *multi-alignment*: the members of each group identified in the clustering step were multialigned, resulting in the construction of a multi-alignment for each gene family. For multi-aligning sequences, the ClustalW algorithm [103] was used.

3. *tree construction*: PHYML software tool [45] was used to build evolutionary trees, implementing a phylogeny reconstruction method, based on maximum-likelihood principle. The resulting tree topologies and branch lengths help in the interpretation of the evolutionary relationships occurring between gene family members.

In addition, to inspect gene family trees, a web-applet, based on PhyloWidget [53] libraries, was integrated. It offers the opportunity to graphically visualize the evolutionary trees in an interactive manner (Fig. 5.5).

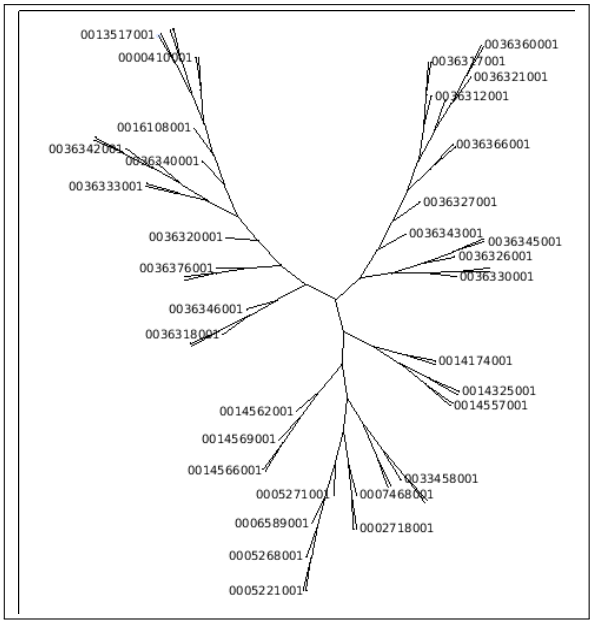


Figure 5.5: The evolutionary unrooted tree relates to terpene synthase family in grapevine genome.

5.4 ANNOTATION IMPROVEMENTS

In this chapter, the main annotation methods integrated in the platform are described. However, the final goal has not been achieved and further efforts have to be done in order to obtain more reliable and rich annotation for predicted genes. The platform extension and enrichment could follow different directions:

- *Biological databases*: the main existent annotation method based on sequence similarity is the comparison of gene catalog with UniProtKB database. This resource is absolutely complete and richly-annotated, but the usage of additional sequence databases can enlarge the annotation coverage.

- *InterPro integration*: functional domains and patterns are discovered using Pfam, SMART and Prosite databases. However, InterPro [50] is a web resource, developed at EBI, that allows to parallely search for these information in different distributed databases. The integrated InterPRO databases include Prosite, Pfam, Prints, ProDom, TIGR, PDB, Panther, etc.

- *Structure annotation*: at present, in the platform, tools for predicting or annotating protein secondary or tertiary structures are partially present (or not at all). These kind of

information is extremely important to assign protein function and cell localization. Database as PDB [14], CATH [76] or software tools like PSIPRED [19] or JPRED [32] could be implemented in the platform.

5.5 ANNOTATION RESULTS

At present, annotation results are not biologically interpreted

In this section, the first results for the functional annotation of *Vitis vinifera* genes obtained with the above explained methods are presented. They comprise the main functional data regarding the more abundant domains or GO categories, the pathways more populated and the type of enzymes more common, but they will not be biologically interpreted as the data analysis step will begin in the next future and functional data results have to be confirmed and validated. However, a brief functional landscape could be useful to design the general map of the grapevine genome.
In the table 5.1 there are a summary of the data showing the number of genes annotated with different methodologies and the annotation redundancy determined by the number of *objects* relative to each methods. For example, there are 16,923 genes annotated with 4,174 different GO terms, or 17,054 genes annotated with 2,909 Pfam domains: in the first case, the *object* refers to *GO term*, while in the second case refers to *Pfam domain*. When the *object* represents transmembrane domain, the number 2 describes its presence or absence in the sequence.

Table 5.1: Functional annotation data.

Method	**Gene number**	**Redundancy**	
		Object	*Number*
Uniprot	17,738	*protein*	172,587
GO	16,923	*GO term*	4,174
PO	17,580	*PO term*	353
KEGG	2,044	*ec*	649
		map	102
PFAM	17,054	*domain*	2,909
SMART	7,843	*domain*	412
Prosite	9,986	*pattern*	1,150
WoLF-PSORT	26,093	*localization*	24
TargetP	26,347	*localization*	4
TMHMM	26,347	*transmembrane dom.*	2
HMMTOP	26,263	*transmembrane dom.*	2
Gene family	17,555	*family*	3,973

5.5.1 Pfam, SMART and Prosite

In the figure 5.6 there are three pie-charts, representing the data of Pfam, SMART and Prosite, respectively. In particular,

each pie shows the 7 most represented domain categories found in grapevine genes. A rapid inspection of the three charts points out a similar pattern of domain abundance, showing little differences.

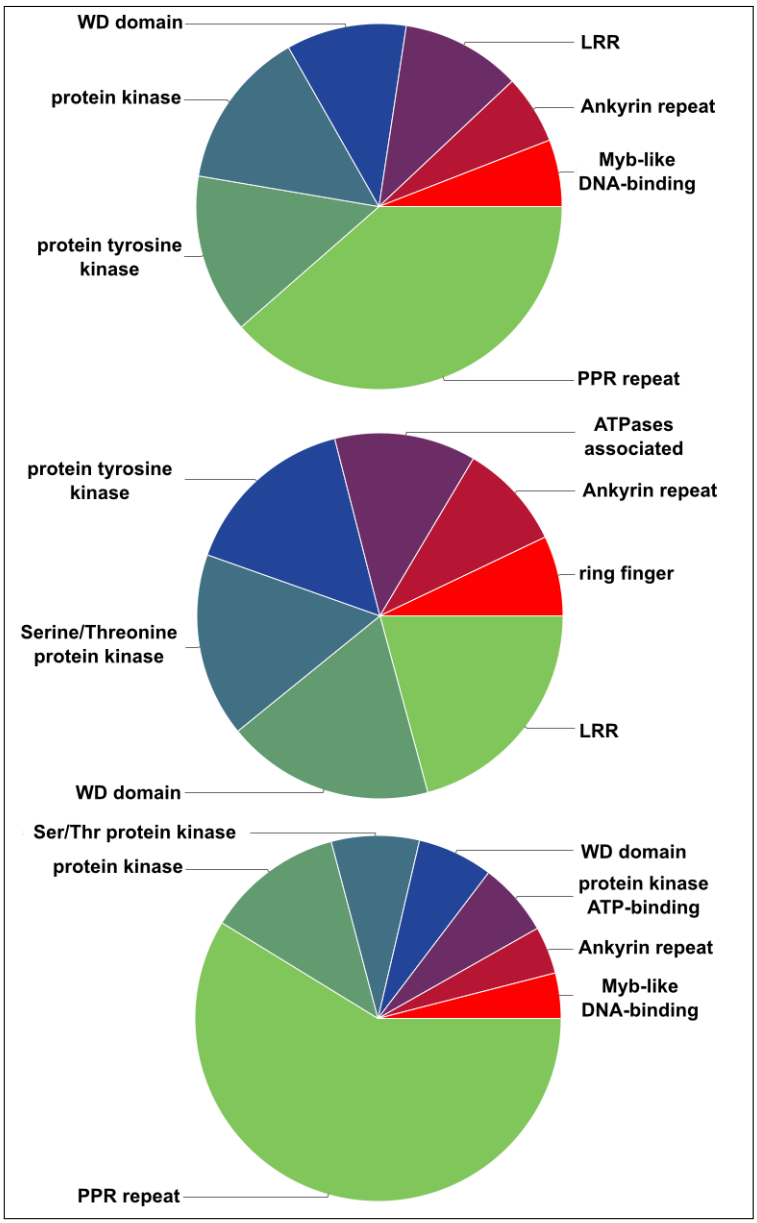


Figure 5.6: The three pie-charts represent the proportions of the 7 more abundant domains respectively in Pfam, SMART and Prosite. *NB: the dimension of a pie section reflect its percentage value compared with the other 6 sections.*

The more important domains are[4]:

- *PPR*: pentatricopeptide repeat (PPR) proteins are characterized by tandem repeats of a degenerate 35 amino acid motif. Most of PPR proteins have roles in mitochondria or plastids. These proteins seem to play a role in post-transcriptional processes within organelles, RNA stabiliza-

4 http://www.ebi.ac.uk/interpro/

tion and processing. It is known that this family is greatly expanded in plants.

- *WD domain*: WD-40 repeats (WD or beta-transducin repeats) are short ≈ 40 amino acid motifs, often terminating in a Trp-Asp (W-D) dipeptide. WD-repeat proteins are a large family found in all eukaryotes and are implicated in a variety of functions ranging from signal transduction and transcription regulation to cell cycle control and apoptosis. In Arabidopsis, several WD40-containing proteins act as key regulators of plant-specific developmental events.

- *protein kinase*: protein kinases can be divide in two groups: serine/threonine specific and tyrosine specific. They are a group of enzymes that possess a catalytic subunit which transfers the gamma phosphate from nucleotide triphosphates (often ATP) to one or more amino acid residues in a protein substrate side chain, resulting in a conformational change affecting protein function and for this reason they have a role on a multitude of cellular processes.

- *Ankyrin repeat*: ankyrin repeat is a 33aa-long tandemly repeated module and it is one of the most common protein-protein interaction motifs in nature. They have no clear functions.

- *Myb-like DNA-binding*: the myb-type domain is a DNA-binding, helix-turn-helix (HTH) domain of $\approx$ 55 amino acids, typically occurring in a tandem repeat in eukaryotic transcription factors and specifically recognizing the sequence YAAC(G/T)G.

- *LRR*: leucine-rich repeats (LRR) consist of 2-45 motives of 20-30 amino acids in length and appear to provide a structural framework for the formation of protein-protein interactions. Proteins containing LRRs are involved in a variety of biological processes, including signal transduction, cell adhesion, DNA repair, recombination, transcription, RNA processing, disease resistance, apoptosis, and the immune response.

- *ATPase associated*: AAA ATPases (ATPases Associated with diverse cellular Activities) domain is responsible for ATP binding and hydrolysis and proteins containing AAA play a large number of roles in the cell including cell-cycle regulation, protein proteolysis and disaggregation, organelle biogenesis and intracellular transport. They also act as DNA helicases and transcription factors.

5.5.2 GO annotation

The GO terms obtained by inheritance from UniProtKB proteins, protein domains and Kegg enzymes permit to annotate

16,923 genes of *Vitis vinifera*. This means that nearly 64% of grapevine genes are annotated with at least one GO term of any ontologies and that there are on average about 6 slim associations per gene. In particular, 56% of genes are annotated with *molecular function* terms, 49% with *biological process* terms and nearly 43% with *cellular component* terms (Fig. 5.7).

64% of grapevine genes are annotated with at least one GO term

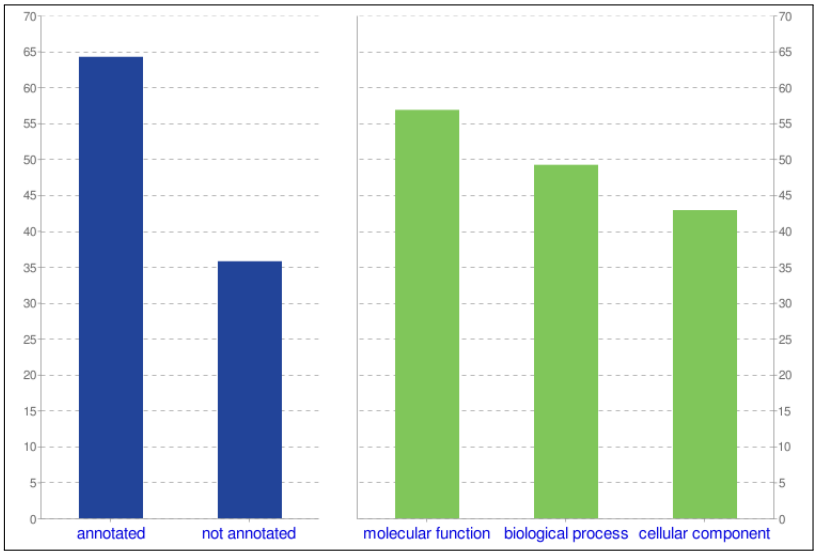


Figure 5.7: The left-hand graph shows the percentage of genes annotated with at least one GO term. The righ-hand graph describes the disjoint distribution of annotations amongst the 3 different ontologies.

Moreover, the figure 5.8 describes the distribution of GO annotations among ontologies, showing that 8,663 genes are annotated with terms coming from all 3 ontologies and approximatively 80% are associated with terms belonging to at least 2 different ontologies. Finally, the figure 5.9 shows the 10 most represented terms for each category.
GO annotations are very useful when there is the need to compare two situations, e.g. microarray experiments, or to extract and collect a set of genes with some specific functional features or cellular localizations in database searching processes. However, the genome-level analysis of GO annotations is not trivial. The description of a genome by means of more or less populated GO terms can result useless or uninformative. Thus, a method to analyze GO annotations at genome-level will be described in a following section.

5.5.3 Protein targeting and transmembrane domains

The transmembrane helices prediction was accomplished by TMHMM and HMMTOP software. To analyze the transmembrane domain distribution, only the common predictions between the two programs are considered. Shared transmembrane predictions regard 18,268 genes, representing nearly 70% of grapevine genome. The remaining 30% demonstrates conflicting predic-

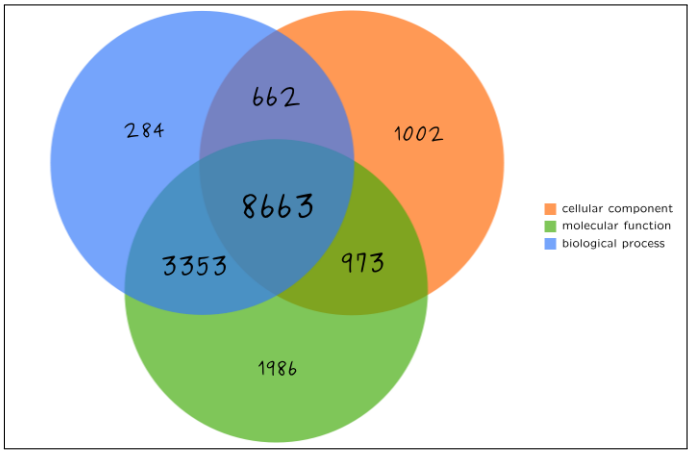


Figure 5.8: The Venn diagram describes the joint distribution of GO annotations amongst the 3 different GO categories.

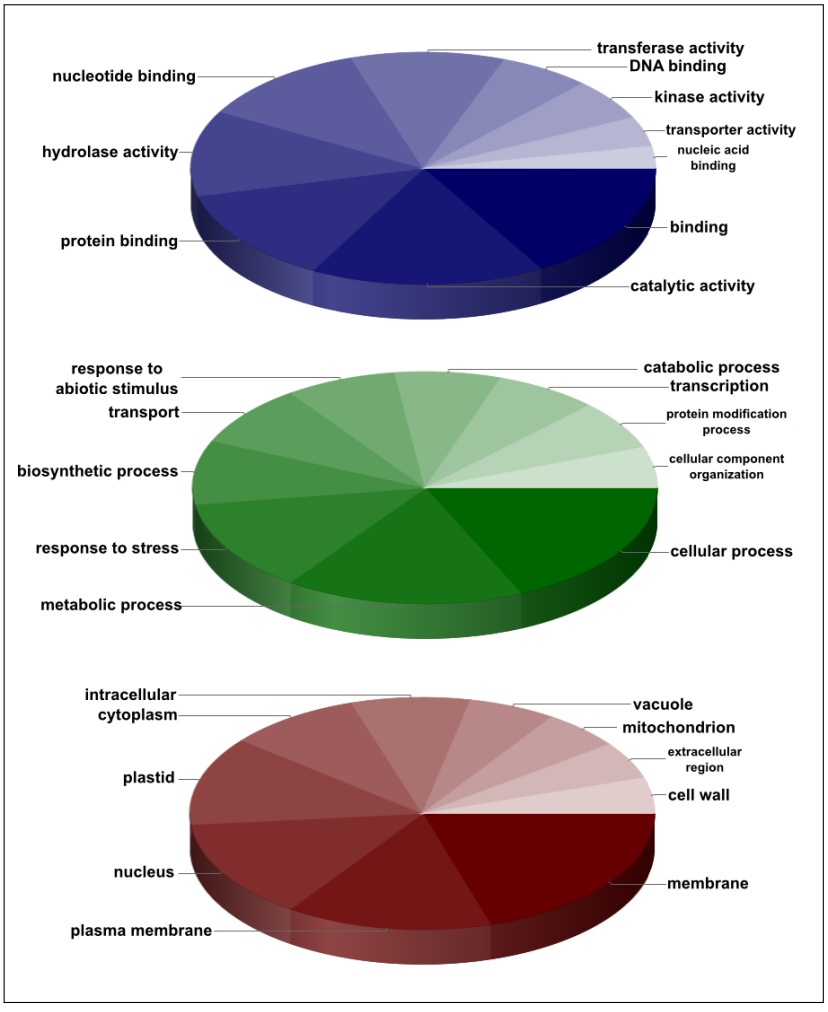


Figure 5.9: These three pie-charts represent the 10 most populated GO terms for *molecular function* (blue), *biological process* (green) and *cellular component* (red) ontologies, respectively. *NB: the dimension of a pie section reflect its percentage value compared with the other 9 sections.*

tions and can be referred as **unclassified**. The following analysis shows that nearly 20% of **shared** predictions found at least one transmembrane domain, ranging from 1 (about 60%) up to 25 (only one case) domains (Fig. 5.10).

20% of genes has a transmembrane domain

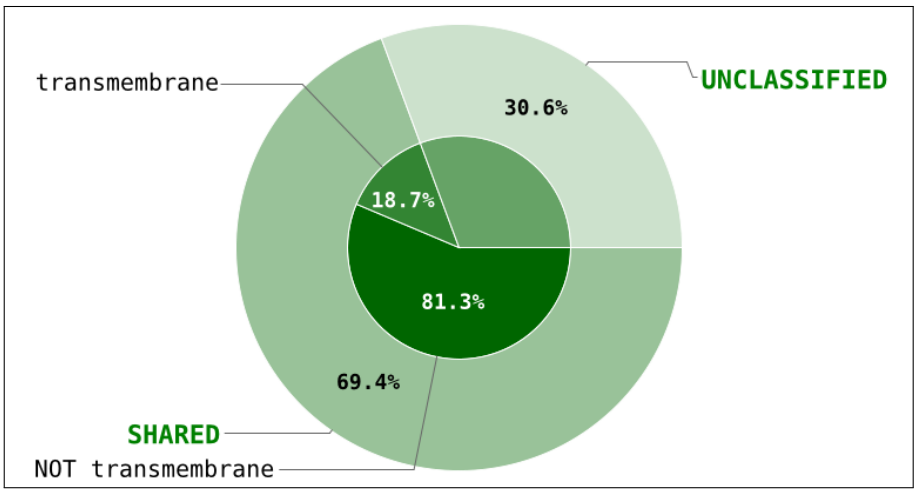


Figure 5.10: The outer circle describes the percentage of common (*SHARED*) and dubious (*UNCLASSIFIED*) predictions between two programs used for transmembrane domain finding. The inner circle focuses on the *SHARED* predictions, highlighting the percentage of the proteins having at least one transmembrane domain.

Gene localization was performed using TargetP and WoLFPSORT software. Both programs assign to their results a score, symbolizing the reliability of the predictions, and give in output several alternatives. Thus, the combined analysis of sub-cellular localization predictions is critical and there is a need to normalize and weight the diverse scores. To address this need, for each i gene the normalized scores were computed according to the formula:

$$\frac{(\mathrm{TargetP\_score}_i^k + \mathrm{WoLFPSORT\_score}_i^k)}{(\mathrm{BEST\_TargetP\_score}_i + \mathrm{BEST\_WoLFPSORT\_score}_i)}$$

where k is a common localization prediction from the set of alternative ones of TargetP and WoLFPSORT, and BEST score is the prediction with the highest score. For example, we suppose that the scores for *chloroplast* localization of TargetP is 3.2 and of WoLFPSORT is 4. However, while the localization with the highest score predicted by WoLFPSORT is effectively *chloroplast* (score 4), the best localization for TargetP is *mitochondrion* with score 5. In this case, the reliability value for *chloroplast* results from $(3.2+4)/(5+4)*100 = 80$.
Using this scoring system, three classes of reliability were obtained: 100%, 80% and 50%. 100% means that the two predictions agree and both have the best score.
The sub-cellular localization results are summarized in the figure 5.11, where the most represented cellular localizations are plotted with percentage values and colored with three different

Nearly 30% of genes has a nuclear destination

colors according to the belonging reliability class. From the figure, it can be clearly visible that nuclear, chloroplast and cytosol are the preferred destinations of grapevine gene catalog.
A careful inspection of data and graphs can arise some doubts about the correctness of annotations. For example, if we focus on membrane proteins, we can note that there are some apparent discrepancies between GO annotations, transmembrane predictions and targeting results. Indeed, GO annotations regarding *cellular component* (CC) ontology show that membrane, or plasma membrane, are the terms more abundant amongst all CC annotations (Fig. 5.9). Otherwise, transmembrane predictions suggest that only the 20% of genes have a transmembrane domains, and moreover, the targeting predictions assign to plasma membrane the 8% of genes. These error effect becomes less evident if we consider the absolute number of different annotations, all classifying approximately 3,000 genes as having a transmembrane domain or belonging to plasma membrane. Probably, the incorrect interpretation is due to artifacts given by different numbers of considered annotations. Indeed, GO CC annotations only regard about 11,000 genes, the greatest part of them annotated with the term *membrane*, maybe because membrane proteins are better studied or simply because they have a greater number of GO annotations.

Diverse annotation distributions can bias the data interpretation

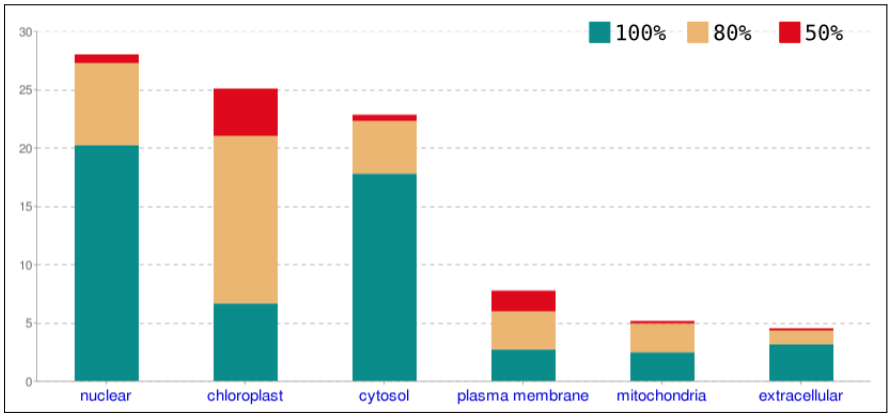


Figure 5.11: The most represented cellular compartments, scored as a percentage value on the total number of genes. The three colors stand for different reliability classes.

5.5.4 Metabolic pathways and enzymes

The gene associations with enzymes and metabolic maps were accomplished using KEGG database. Unfortunately, the analyses of most abundant enzymes and metabolic pathways could be biased because of the small number of annotations (only 2,044 genes have a KEGG annotation). However, the figure 5.12 shows the 5 metabolic pathways with more genes and the most represented enzymes. It is worth to note the predominant presence of maps related to the biosynthesis of secondary metabolites and

the high number of genes involved in the cell wall modification and alteration (pectinesterase and beta-glucosidase).

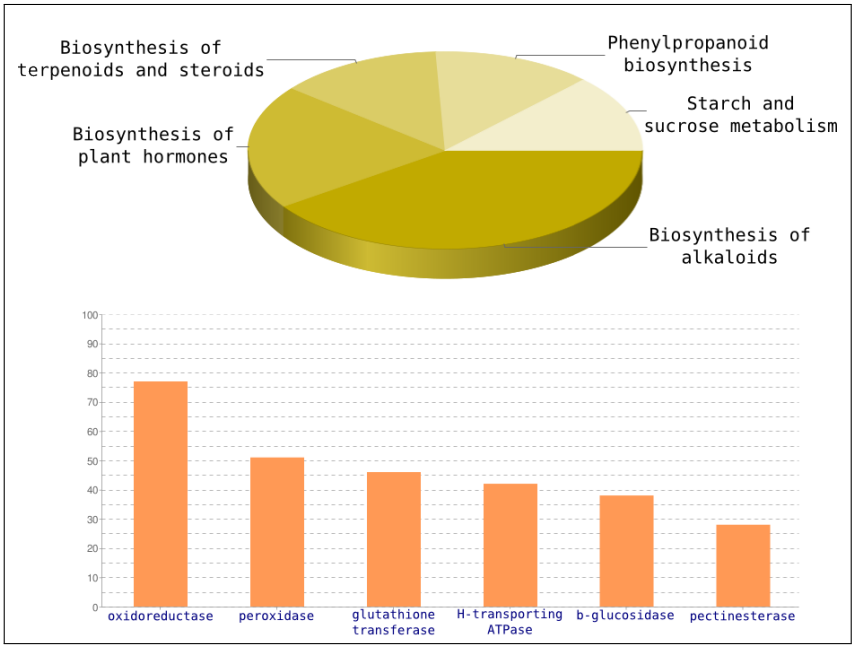


Figure 5.12: The pie-chart highlights the 5 most populated metabolic pathways. The histogram shows the number of genes classified according to enzyme category.

5.5.5 GO analysis

As already pointed out, a genome-level functional analysis by means of GO terms arises some problems. The choice of the correct level of specificity, the stringency used in the annotation, the quality and abundance of associations between proteins or domains and GO terms strongly affect the final annotation, leading to results difficult to interpret. Moreover, the description of the genome functional landscape through more or less populated GO categories is useless and uninformative. Otherwise, the comparison between GO categories differently populated in different organisms could be more useful. This would allow to follow the GO categories dynamics through evolution, highlighting metabolisms maintained in some organisms and depleted in others. It could be interesting to observe if some differences could be visible already at genome-level and not only at transcriptional level.

Evolution dynamics of GO categories could highlight functional constraints

Thus, *Vitis vinifera* was compared with three different organisms: *Arabidopsis thaliana*, *Oryza sativa* and *Populus tricocharpa*. To avoid bias due to annotation methods, the three organisms were reannotated with GO module of CRIBI platform. In this way, all the organisms were annotated with the same method. Because the platform annotates genes with most specific terms (leaf nodes in the GO tree), it could be possible to consider differently annotated genes that are described by parent terms (annotation at

Genome-level functional insights can be inferred from pairwise annotation comparisons

different level of the same path). This possibility was avoided by annotating genes not only with leaf nodes, but also with all ancestors. Then, the procedure made a pairwise comparison between *Vitis vinifera* and the others three organisms. For each comparison, a proportion test was performed for each GO category, assuming an equal distribution between organisms as null hypothesis, followed by a Bonferroni correction. The results of this procedure were the GO categories that are significantly over/under-represented between organisms (with a q-value = 0.01), and thus, evolutionary remarkable.
A graphic clustering script was developed to aid correct interpretation of obtained results (Fig. 5.13).

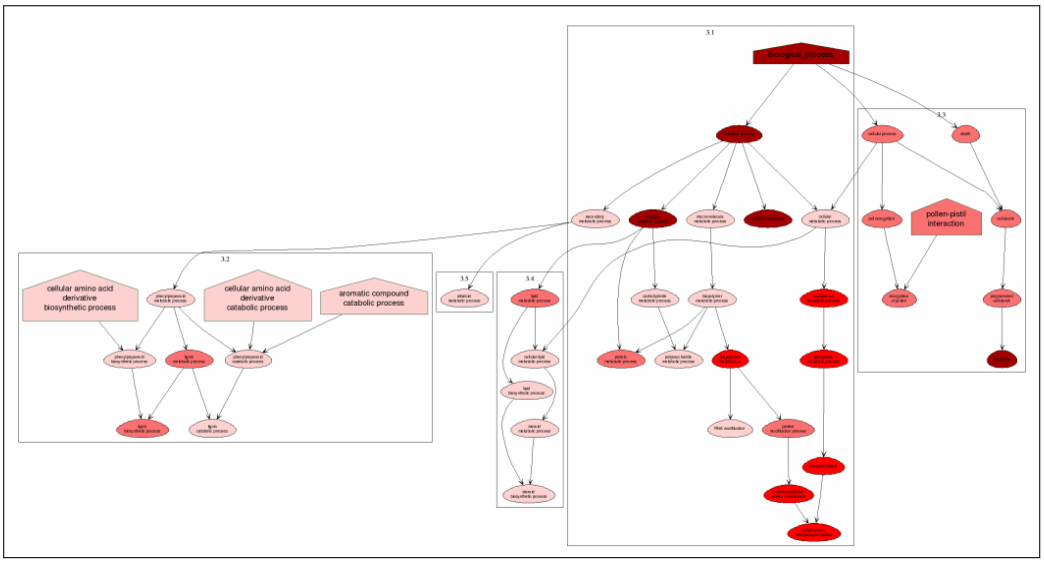


Figure 5.13: Graphic representation of the significant GO terms resulting from a pairwise comparison between *V.vinifera* and *A.thaliana*. The red color intensities indicate the over-representation of GO term in *V.vinifera* compared to *A.thaliana* (dark red → great over-representation).

The preliminary results coming from this method highlight two different aspects:

- the number of significant GO categories follows the phylogenetic relationships between the four organisms: poplar is the organism with the smallest number of significant GO categories for all three ontologies, while rice has the greatest number of significant GO categories.

- a rapid analysis of the GO terms over-represented in *Vitis* compared to *Arabidopsis* shows that grape has more genes involved in the lignin catabolic process.

These preliminary results are not sufficient to validate the method, but they indicates that the direction is correct. However, there could be some problems due to differences in number and quality of annotations that can bias the final results and erroneously consider remarkable some GO terms that are not.

5.5.6 Orthology analysis

The word homology refers to any similarity between biological sequences that is due to their shared ancestry. Genes with highly similar DNA or amino acid sequences are likely homologous. Orthologs and paralogs are two types of homologous sequences. Orthology describes genes in different species that originate from a common ancestor, thus, separated by a speciation event. Orthologous genes may or may not have the same function. Otherwise, homologous sequences are paralogous if they derived from a gene duplication event and they can be present in the same organism.
For the study of orthologous relationships of *Vitis vinifera* with other plant species, an approach similar to that used for GO analysis was adopted. Indeed, the grapevine predicted proteins were aligned against *A.thaliana*, *P.trichocarpa* and *O.sativa* proteomes, in order to search the orthologs in these three species. To qualitatively estimate the level of orthology between grape genes and genes of other plant species, a similarity score was assigned to each alignment. At first, each grapevine gene i was aligned with the other plant counterparts using the Smith-Waterman algorithm, finding the ortholog. The resulting score s_i is compared with the *best score* BS_i obtained by the self-alignment of grapevine gene. The final score for an othologous alignment is computed with the formula $\frac{s_i}{BS_i} * 100$.
At this point, the scores are partitioned in four classes with intervals of 25 p.p., and each species-specific alignment are assigned to the correct class according to its score. Then, the number of genes for each similarity class are computed and plotted. The results, summarized in the figure 5.14, shows three clear tendencies:

An evolutionary landscape can be obtained measuring the similarity amongst orthologs

1. a great majority of *O.sativa* genes has poor similarity values. Indeed, the rice genes outperform the genes of other organisms in the first two classes that represent up to 50% similarity values.

2. an huge amount of *P.trichocarpa* genes has high similarity values, outperforming the genes coming from other organisms in the last two classes, that correspond to similarity values grater than 50%.

3. *A.thaliana* genes maintain a medium behavior in all similarity classes.

This analysis seem to confirm the phylogenetic relationships that consider poplar as the nearest neighbor and rice as the most distant organism.

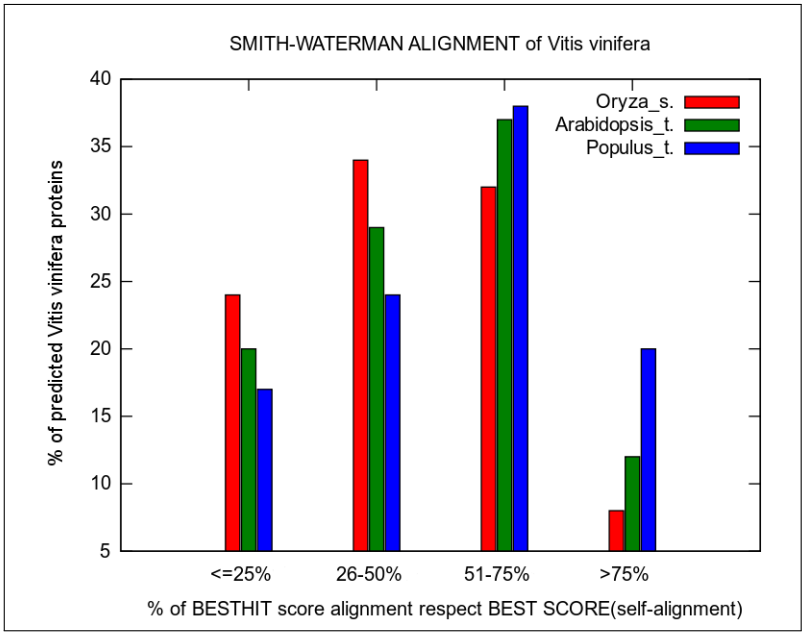


Figure 5.14: The histogram shows the similarity scores of orthologs in different plant species. In x-axis the similarity scores are subdivided in four classes. In y-axis the percentage of grape proteins are indicated. The grape proteins show a geater similarity score with poplar than with rice and *A.thaliana* orthologs.

Part IV

RETRIEVING THE GENOME INFORMATION

6

DATA STORAGE AND DATABASE INTERFACE

CONTENTS

In the previous chapters, the main methods for computational prediction and functional annotation along with their application in the *Vitis vinifera* genome were described. The annotation data production is only the first step in the study of a genome, because they have to be inspected, analyzed and validated by the scientific community. Indeed, computational methods are automated systems and they can easily produce incorrect predictions or annotations, due to stringency choices, quality of reference database annotations, etc. Thus, the annotation data have to be made available to the researchers for the subsequent biological analysis and validations.
In this chapter, the grapevine database structure, storing all genome data, is described. However, genome analysis procedures require efficient, flexible and scalable solutions to facilitate access to these data in a rapid and interactive manner and from disparate locations around the world. The underlying idea is to give to the end-users an interface to the *Vitis vinifera* genome database, where it can be possible to easily choose the query criteria and collect the desired information. A new modular query system that represents an useful platform to access to the grape data was developed with CRIBI collaboration.

6.1 DATABASE STRUCTURE

The database structure has a gene-centered star-topology

Grapevine annotation data were stored in a MySQL database[1], developed for the purpose. The designed database structure is simple and sufficient for current data, but can easily adapted and modeled to face the future increasing of data complexity, e.g. alternative splicings, protein isoforms, protein tertiary structures, etc. The database has a star-topology, meaning that all database tables refer to a central object, that in this case is *gene*. Each table contains data relative to diverse, independent biological concepts. Indeed, there is the table containing chromosome coordinates of genes, the table with GO terms associated to each gene, table of paralogs and orthologs, etc. All these tables are linked to the central object *gene_name*, ideally represented as a table with the sole gene name field. However, the actual central table is *gene_structure*, that describes the gene chromosome, the positions in the chromosome, the plus or minus direction of gene and the exon-intron structure (e.g. composition of the gene by means of CDS or UTRs). The other surrounding tables are:

- *Gene_seq*: this table contains the nucleotide sequences of genes and the relative translated amino acid sequences.
- *Gene_annotation*: it contains the UniProtKB proteins used for preliminary annotation of the grapevine genes.
- *Grape2GO*: in this table, genes are associated to correspondent GO terms. In addition, beyond the description and belonging ontology relative to GO terms, there are associations with GO plant slim terms and sources of annotation (e.g. UniProt, Pfam, Kegg, etc.). It is worth to note that redundancy is minimized: it is not possible to find the same gene→GO term association coming from more than one source.
- *Grape2pfam*: a protein domain table stores all data regarding code and description of domains and position in the protein sequence. Repeat domains can have several table records for the same gene, representing the diverse matches in the protein sequence. SMART and Prosite data have the same table structure.
- *Grape2kegg*: in this table, each gene is associated to one or more enzymes and to metabolic pathways involving the associated enzymes.
- *Grape2targetp*: it resumes the information regarding the subcellular localizations of genes predicted by targetP. The possible localizations are *chloroplast*, *mitochondrion*, *secretory pathways* and *other*. A reliability score is recorded for each gene→localization association. A similar table structure is present for WoLFPSORT data. In this case, the lo-

1 http://www.mysql.com/

calization alternatives are more abundant, contemplating also the dual localization.

- *Grape2tmhmm*: this table contains the position (if present) of transmembrane helices in the protein sequence. Moreover, the domain topology is described. The same structure is present for HMMtop data.

- *Gene_family*: each gene can be member of a gene family. In this table there are the gene families identified by CD-HIT with the relative number of components.

- *Gene_syn*: it contains the possible alternative names, gene symbols or synonyms of genes found in the annotation procedures.

The current database structure offers a rapid, simple but exhaustive solution for the management of grapevine data. However, in the next future there is the need to re-model and extend the database structure to deal with an huge amount of heterogeneous data and to fully exploit the relational potentialities.

6.2 DATABASE INTERFACE FEATURES

Genome databases management systems already exist [94; 60; 43], providing fast and flexible querying procedures of large biological data sets and integration with third-party data and tools. However, these systems are useful for integrating information deposited into diverse databases around the world and dealing with not domain-specific knowledge. This implies an high-level configuration process, reflecting a database complexity not present in the Grape database. For these reasons, a simpler system that maintains flexibility but involves a more agile configuration procedure was implemented. This does not exclude in the future, with the growth of grape data volume and complexity, to extend and improve actual functionalities or adopt a solid system as BioMart [94].

Interface to Grape database has to be simple but efficient in data mining processes

The main goal of this project is to provide to the users a powerful tool for the delivery of customized sets of grape genome data. In the planning phase, there was the definition of the main properties that a database interface must have to be useful in data retrieval processes: effectiveness, efficiency, modularity, ease of use and configuration. Moreover, since database are organized around *genes*, the system can be defined as gene-centric. All the possible queries are built in relation to the central object, that is gene. In this way, all delivered data refer to the presence or absence of some properties on the genes. The query results are lists of genes that have some protein domains, that are annotated with some kind of GO term or that participate in specific metabolic pathways, etc., according to defined criteria.

EFFECTIVENESS Two common problems affect the enquiring of genome databases: the large amount of information that can result from query procedures and the little flexibility in the choice of query criteria. In this way the results analysis and the selection of the significant data become extremely difficult.
A possible solution for the first aspect can be represented by classification systems, that assign a significance score to results. Therefore, in the query platform a **ranking system**, that can aid the end-user in the evaluation of results, was implemented allowing a direct assessment of result significance. In particular, all resulting data are shown in decreasing order on the base of satisfied criteria, chosen in a previous step.
For what concerns the second aspect, the database interface is extremely flexible because it can be easily extended with new queries according to the user needs, covering any information present in the database.

Ranking system helps the user in the evaluation of results

EFFICIENCY The choice of a large number of selection criteria can result in long waiting periods, because the server needs a lot of time to process complex queries. This issue was resolved dividing complex queries in simple ones. Each simple query is independently processed by the server, decreasing the execution time. The merging of results is accomplished in a post-processing phase, when the ranking system collects all the output of simple queries and produces the final weighted results.

MODULARITY In a genome project there is a continuous increase of data volume and heterogeneity, and database update or extension events are not rare. Thus, it is essential for a database interface to be modular and flexible, adapting to new structures and allowing the composition of new selection possibilities or the modification of the old ones. In the query platform this modularity was implemented separating the software implementation from configuration level. In this way any modification of query possibilities is realized without the modification of software code. To do that, the interface software has to maintain the same behavior in presence of different configurations, that describe the query parameters. This is possible by structuring the information and encapsulating it in a general data structure that describes and generalizes all database queries. Thus, different configuration data are presented to the software with the same form. This goal was achieved using an XML data structure. XML (eXtensible Markup Language) is a meta-language that defines and describes the structure of information (appendix A). In particular, an XSD (XML Schema Definition) was designed to guide the construction of XML configuration files. In this way, through a simple editing process of XML files, it is possible to change the database interface, adding (or removing) database interrogation possibilities without intervening on software code.

Modularity faces the interface update or extension problems

USER-FRIENDLINESS A database interface is a tool developed for users that probably have no computing expertise. Thus, it has to be easy and immediate. The interface software employs intuitive web forms for searching and it has been implemented through CGI scripts, that automatically read XML files and translate them in HTML pages. Hence, the user can interactively fill out forms, choosing the criteria for filtering the output data. The results are shown in another web page, where they are ordered according to score assigned by the ranking system.
On the configuration side, although XML helps the developers in the description of queries, it is not easy to edit without appropriate tools. For this reason, there is a planning to develop a Graphical User Interface that improves and facilitates the configuration procedures (reading and writing the XML configuration files).

6.3 INTERFACE IMPLEMENTATION

The first release of database interface has been developed using a web-based system written in Perl. The querying of Grape database through this interface is organized into three web pages that accomplish all interface functionalities: **query page**, **result page** and **gene report page**. Below is a detailed description of each page. The system is designed around three levels:

- the first level consists of the Grape database, implemented in MySQL RDBMS. It contains all genome data organized in several tables gene-centered.
- the second level is represented by the web interface and the XML configuration files. In particular, there are two XML files that guide the construction of *query page* and *gene report page*. Both files are built on an unique XSD starting from two different root nodes. These files contain mainly the information about query construction, but also for graphical templates and layouts.
- the third level is centered on the *engine*, a server-side program that collects the user input provided using the *query page*, transforms them in SQL statements, processes them and elaborates the results through the ranking system. Finally, it outputs the scored list of genes in the *result page*.

The modular architecture based on XML gives the possibility in future releases to extend this interface to other implementation forms, as *web services*.

6.3.1 Query page

The query page represents the first stage in the querying procedures (Fig. 6.1). It is the real query interface that allows users

Query page allows to configure the searching criteria

to group and refine data based upon many different criteria. The page is organized in several sections, each one representing a simple, independent query to database. At present, for convenience of thinking, each section corresponds to a single biological property of the gene (e.g. presence of protein domains, signal patterns, annotation with some GO terms, etc.), but it is possible to create more complex queries involving different biological domains. In any case, the linkage between diverse biological features is highlighted by the ranking system in following steps, and there is not particular reason to complicate server requests.
Each section is described in the XML query file, that specifies the parameters used for database request (database, table, fields to extract, filters, etc.), the form layout (scrolling list, popup menu, textfield), the data to use for selection (`select` HTML tag in scrolling list), the graphical template, the titles and header comment and position in the left-side menu. Moreover, some sections could represent data of great importance, and that are essential for further analysis. In this case, for each section there is the `mandatory` box that excludes from the output, if checked, all the genes that do not satisfy the criteria chosen in that particular section. The sections are grouped in biological class (Protein domains, ontologies, etc.) and subclass (Gene Ontology, PFAM, etc.), according to the information extracted from the associated query. This classification is summarized in the left-side menu.
In this page, the user has only to select the interesting sections and the criteria to filter the results. After the "`Submit`" button clicking, the chosen parameters are sent to the *engine* for query processing.

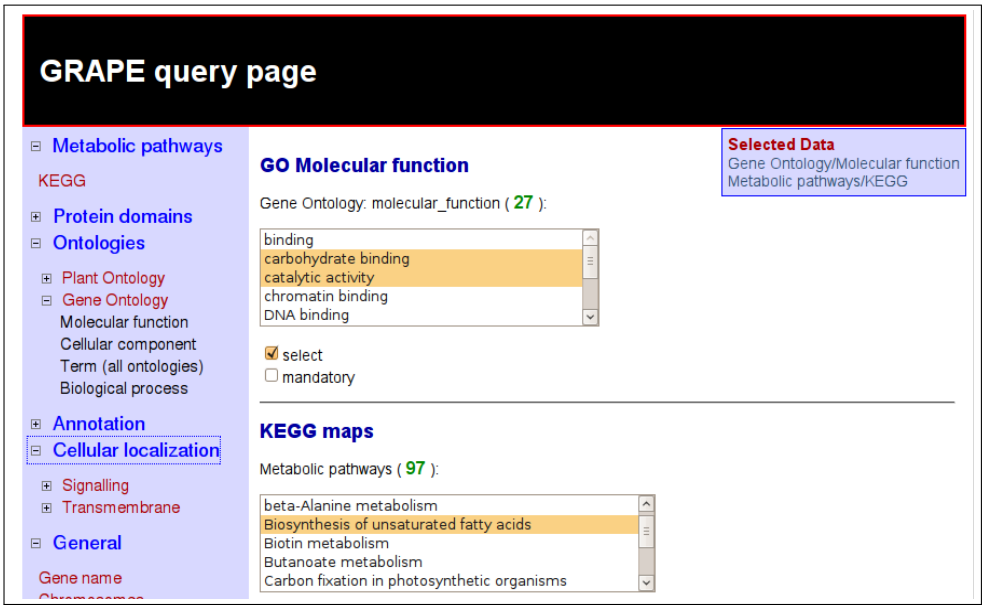
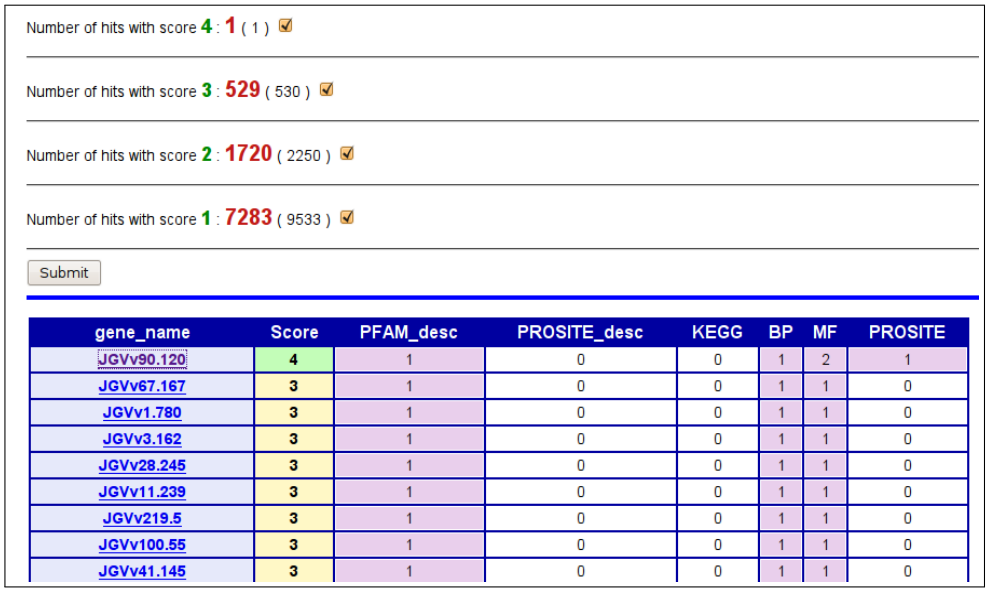


Figure 6.1: Page of query selection criteria.

6.3.2 Result page

Results page implements the ranking system for evaluating the result data

This web page is directly generated by the *engine*. This script receives the user input provided by the query page and automatically generates all of the structured query language (SQL) required to process the query. As explained in the previous paragraph, each section corresponds to a simple, independent query. The engine system processes each query independently from the others. The single query results consist of two terms: the *key term*, that is `gene_name`, and the database field corresponding to the filtering criteria. Only in the next stage the results of the single queries are merged referring to the key term. The ranking system counts for all the gene occurrences in the different queries and sorts genes in decreasing order based on the number of satisfied queries. Obviously genes that do not satisfy *mandatory* sections are cleaned from the ranking.
The generated web page gives a brief summary of the score distribution amongst results, allowing to decide the score threshold for filtering (Fig.6.2). The gene list is presented in a table where rows correspond to genes and columns to submitted queries. Each row describes the *scoring pattern* of a gene, where the cell related to satisfied criteria is highlighted.



gene_name	Score	PFAM_desc	PROSITE_desc	KEGG	BP	MF	PROSITE
JGVv90.120	4	1	0	0	1	2	1
JGVv67.167	3	1	0	0	1	1	0
JGVv1.780	3	1	0	0	1	1	0
JGVv3.162	3	1	0	0	1	1	0
JGVv28.245	3	1	0	0	1	1	0
JGVv11.239	3	1	0	0	1	1	0
JGVv219.5	3	1	0	0	1	1	0
JGVv100.55	3	1	0	0	1	1	0
JGVv41.145	3	1	0	0	1	1	0

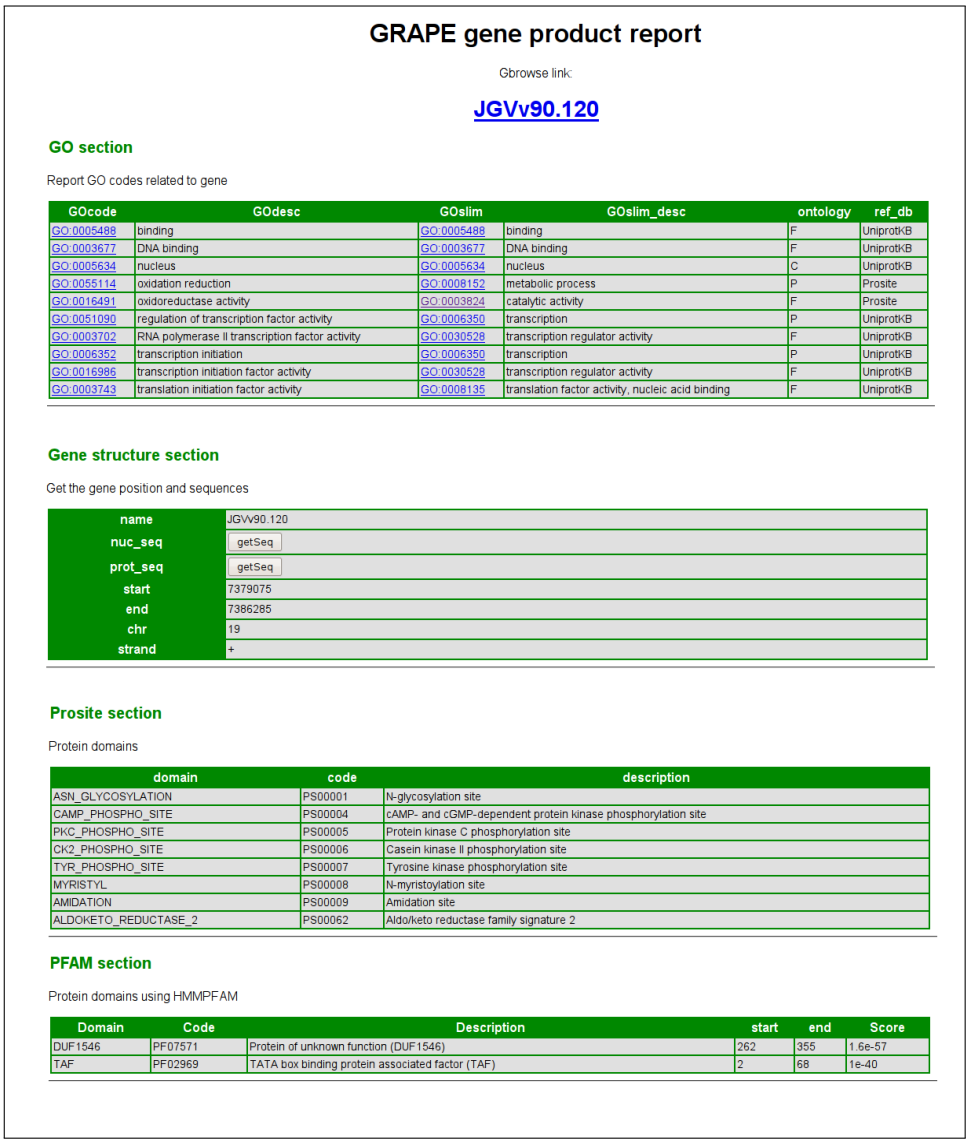
Figure 6.2: Page of query results.

6.3.3 Gene report page

Gene report page offers a complete landscape of gene properties

Once obtained a gene list of a particular interest, a biologist may want to further investigate other genomic aspects, not involved in the filtering phase. In the *result page*, each gene is linked to another web page, called *gene report*, where all information about gene stored in the database can be visualized (Fig. 6.3). As the *query page*, this report is organized in sections correspond-

ing to different biological features. An XML file, similar to that which governs the creation of query page, guides the sections construction. In this case, it is designed to deal with descriptive data, focusing on information presentation rather than query filters. An important feature is the possibility to define in the XML a web link for a particular record presented in the report, giving the possibility to reach external resources, increasing the analysis potentialities. An example is the gene families section, where the `family_id` is linked to a script that allows to investigate graphically the phylogenetic tree.



GRAPE gene product report

Gbrowse link:

JGVv90.120

GO section

Report GO codes related to gene

GOcode	GOdesc	GOslim	GOslim_desc	ontology	ref_db
GO:0005488	binding	GO:0005488	binding	F	UniprotKB
GO:0003677	DNA binding	GO:0003677	DNA binding	F	UniprotKB
GO:0005634	nucleus	GO:0005634	nucleus	C	UniprotKB
GO:0055114	oxidation reduction	GO:0008152	metabolic process	P	Prosite
GO:0016491	oxidoreductase activity	GO:0003824	catalytic activity	F	Prosite
GO:0051090	regulation of transcription factor activity	GO:0006350	transcription	P	UniprotKB
GO:0003702	RNA polymerase II transcription factor activity	GO:0030528	transcription regulator activity	F	UniprotKB
GO:0006352	transcription initiation	GO:0006350	transcription	P	UniprotKB
GO:0016986	transcription initiation factor activity	GO:0030528	transcription regulator activity	F	UniprotKB
GO:0003743	translation initiation factor activity	GO:0008135	translation factor activity, nucleic acid binding	F	UniprotKB

Gene structure section

Get the gene position and sequences

name	JGVv90.120
nuc_seq	getSeq
prot_seq	getSeq
start	7379075
end	7386285
chr	19
strand	+

Prosite section

Protein domains

domain	code	description
ASN_GLYCOSYLATION	PS00001	N-glycosylation site
CAMP_PHOSPHO_SITE	PS00004	cAMP- and cGMP-dependent protein kinase phosphorylation site
PKC_PHOSPHO_SITE	PS00005	Protein kinase C phosphorylation site
CK2_PHOSPHO_SITE	PS00006	Casein kinase II phosphorylation site
TYR_PHOSPHO_SITE	PS00007	Tyrosine kinase phosphorylation site
MYRISTYL	PS00008	N-myristoylation site
AMIDATION	PS00009	Amidation site
ALDOKETO_REDUCTASE_2	PS00062	Aldo/keto reductase family signature 2

PFAM section

Protein domains using HMMPFAM

Domain	Code	Description	start	end	Score
DUF1546	PF07571	Protein of unknown function (DUF1546)	262	355	1.6e-57
TAF	PF02969	TATA box binding protein associated factor (TAF)	2	68	1e-40

Figure 6.3: Gene report page.

6.4 QUERY XSD

The main feature of this query system is the modularity and flexibility that allows to extend the interface in few simple oper-

Query information is encapsulated in XML files, defined by appropriate XSD

ations. This goal can be achieved by structuring the query SQL statements and describing them in an XML file. By this way, query information is encapsulated in a standard data structure and the system software can build and process the SQL statements according to the parameters present in specific xml-tags. Each XML file can be represented with a tree structure, composed of nodes and node-linking branches, that obeys the rules defined in an XSD (Xml Schema Definition). XSD is a language that describes XML files and defines the allowed elements (or nodes), the associated data-types and the hierarchical relations between elements. In a XSD many *root nodes* can be identified as starting elements guiding the XML formation. In other words, the XML file represents the instance of an XSD, given a root node.
For the query system, an XSD was designed to define the structure necessary for querying procedures and, since this is a web-based system, features describing the layout and web environment. In this manner, for adding a section in the *query page* it is enough to add a xml section in the `query.xml`, filling out all the fields necessary to build a well-advised SQL statements and HTML code.
The designed XSD consists of two root nodes, *query* and *report*, that create respectively the `query.xml`, used by query page, and `report.xml`, used by gene report page. There are few differences between these files reflecting the different task accomplished by the related stage. In fact, gene report page is focused more on descriptive features and does not allow the interactivity given in the query page.
Starting from the *query* root node, the XSD provides for three child nodes, as summarized in Fig. 6.4:

- SECTIONS this node embodies all the query sections, that correspond to the data structure used for queries. It has one child node, that is *section*, with a $1 : n$ multiplicity. This means that *sections* node can have min one and max n section children.
- DATABASE this node contains all the information for the connection to Grape database.
- LAYOUT the title, comments and web page templates related to the *query page* can be defined trough this layout node.

Below is a description of xml trees that generate from these three basic nodes.

6.4.1 Section definition

The SECTION node allows to completely define each section that will appear in the *query page*. Each section consists of two levels: a **background** and a **foreground** level. The former contains the knowledge needed to build the SQL query as the reference database table or the table fields to extract; the latter groups

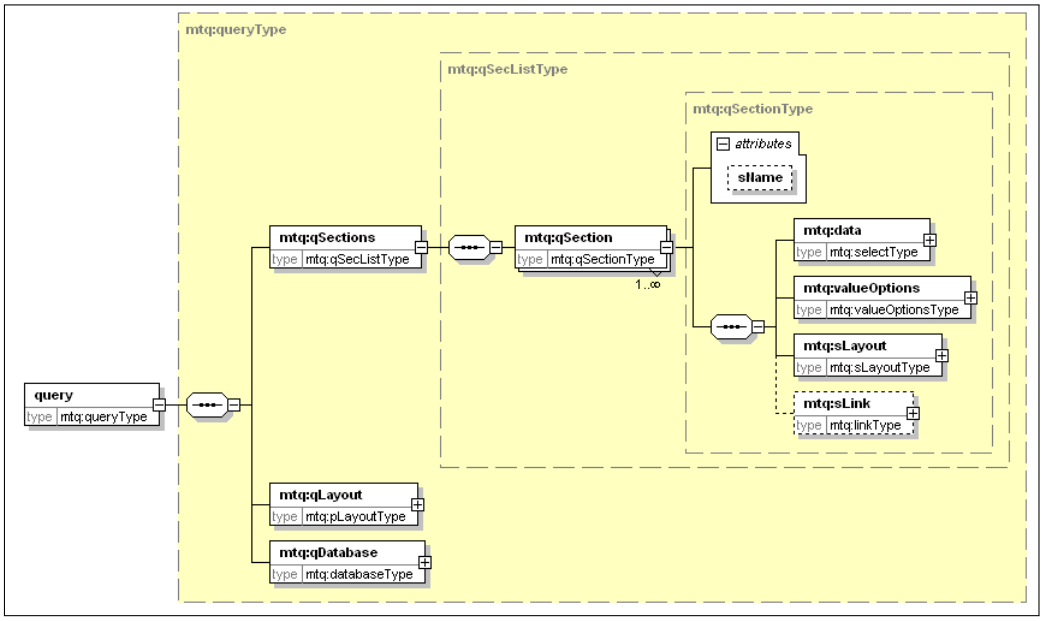


Figure 6.4: Relation between QUERY root and its three child nodes: sections, database and layout.

all the visual and interactive features as the section title, the web template, the web form layout and mainly the filtering criteria that build the SQL *where clause*.
Each SECTION has an attribute *sName*, that is an unique identifier, and is formed by four nodes (Fig. 6.4):

1. DATA: it contains all parameters necessary to the creation of SQL statements. It is a mandatory node, because each section must be associated to a database query. It generates a *selectType* subtree (Fig. 6.5) and its child nodes reflect the typical data required by a *select* SQL statement: the DISTINCT option, FIELDS representing the data to extract from database, reference TABLES, CONDITIONS for the *where clause* and the optional nodes ORDER and LIMIT.
   In particular, the TABLE node offers the possibility to intersect two or more tables, if necessary (Fig. 6.6). The binary relation for table pairs can be defined through the RELATION node that allows to specify the intersection fields and logic. A particular subtree class, named *predType*, is used in several tree regions where data characterizing a database field (reference table, field name and optional alias) are required, e.g. P1 or P2 child nodes of RELATION.

2. VALUEOPTIONS: it is the XSD portion that describes the filtering criteria of each section. It gives to the user the possibility to limit the initial search to a subset with particular characteristics. It consists of a FLAYOUT node that defines the visual web form types (e.g. scrolling list, popup menu, textfield), the possibility of multiple choice or the form size. In the case of a selection web form, the different options to choose are determined by OPTDATA node, that

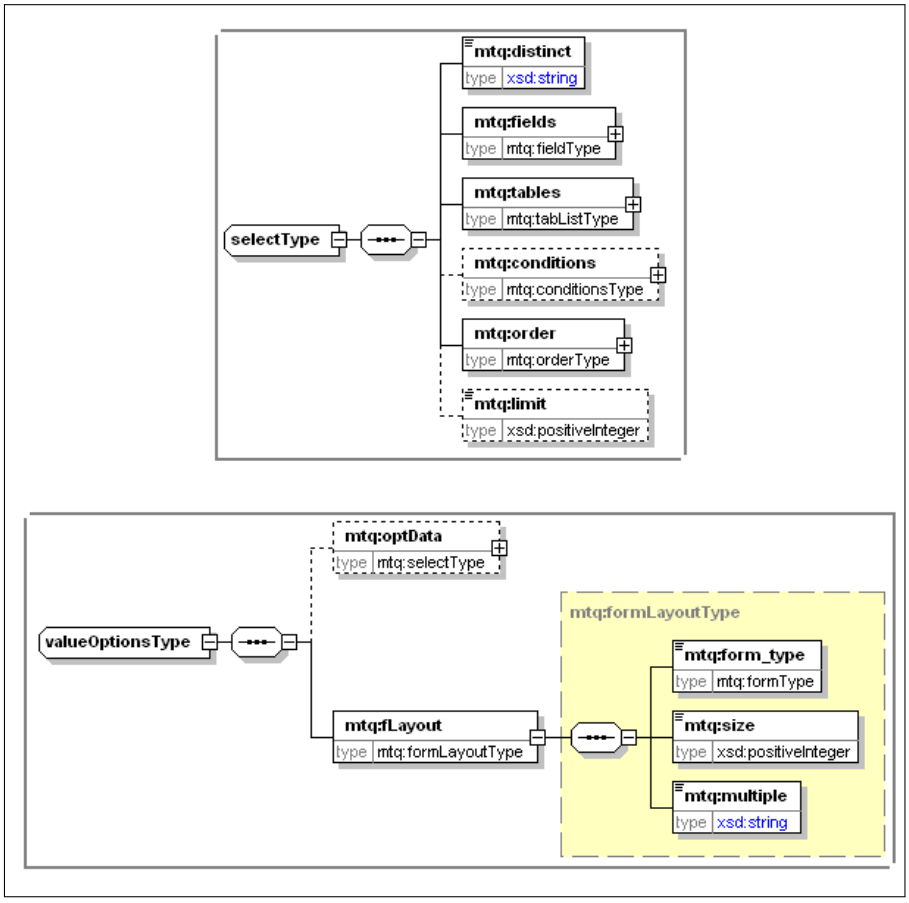


Figure 6.5: Part of the SECTION subtree is represented. The nodes necessary to construct the SQL query are expanded: DATA (figure on the top) and VALUEOPTIONS (figure on the bottom).

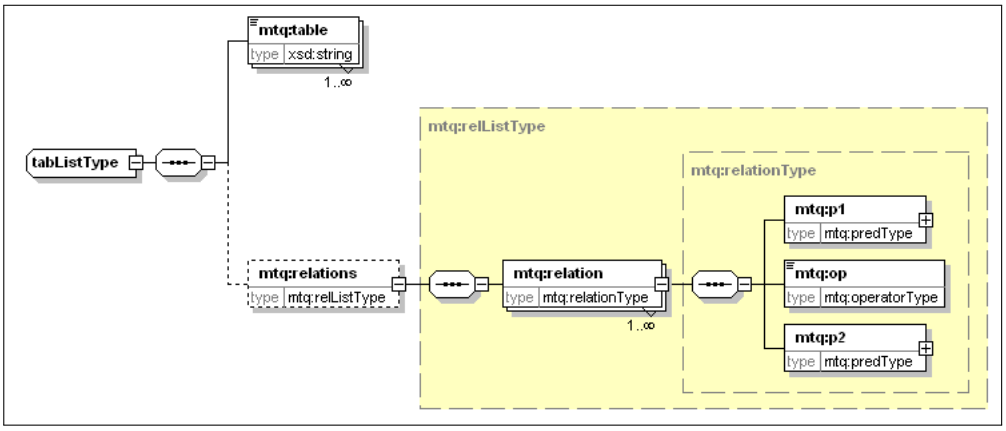


Figure 6.6: The TABLE node allows to specify the tables involved in the query: they can be linked through the RELATION optional node that defines a binary connection.

builds a *selectType* subtree (described above). In this way, the listed options are specified by another database query. The criteria selection accomplished by the user will complete the *where clause* of the SQL statement defined by the DATA node.
VALUEOPTIONS outlines the interactive part of the XSD and this node represents the main difference between *query* and *gene report* xml files, because the report stage does not contemplate interactivity.

3. SLAYOUT: this node is necessary to set-up the web environment related to the specific section (Fig. 6.7). It is possible to insert a section title and a brief description of the associated query, or choose a visual template and position the section in the left-side menu of query page. In particular, the menu classification consists of at maximum three categories that stands for the abstraction levels: in increasing order of specificity there are class, subclass and final nodes.

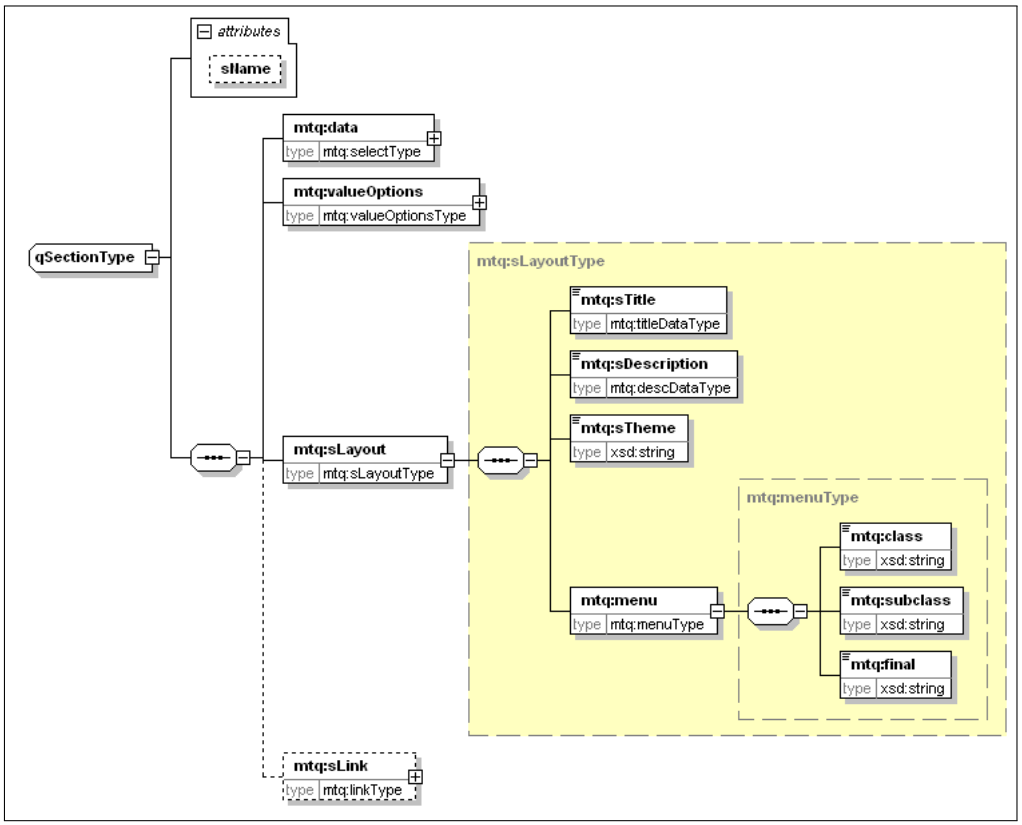


Figure 6.7: The SLAYOUT subtree is represented.

4. SLINK: in some cases it can be useful the association of a web link (e.g. a simple URL or a web server script) to a specific data record. For this purpose, the XSD affords the *linkType* subtree allowing to specify the link typologies and the potential parameters needed by the URL or script. In this case, the SLINK is a *linkType* node that binds the query

results of one specific section to a well-defined link in the result page.

Complex queries definition

When there was the description of *selectType* subtree, the CONDITIONS node was mentioned: it defines the *where clause* of a SQL statement. This statement part is very important because it determines the query filters. The usage of several databases and tables or restriction conditions involving different fields and values makes the queries extremely complex. The structuring of this kind of information, and so the xml transformation, becomes very difficult without an appropriate strategy.

Query filters have to be encoded in XML files

For this reason a system describing the *where clause* was studied. This system represents a good balance between information clarity and complexity. To better understand the adopted approach, it is necessary to introduce the concept of **simple** and **complex** predicate. A typical SQL statement is:

```
select f1,f2 from t1,t2 where pred1 and pred2
```

In this *where clause* model, two **simple** predicates, represented by *pred1* and *pred2*, are connected trough a logic operator (e.g. AND, OR), forming a **complex** predicate. Otherwise, a simple predicate is a "key-value" pair bound by relational operators (e.g. =,! =,>,etc.). In other words, simple states are filters and complex states represent the logic relation between simple states.
Combinations of simple and complex predicates can create complicated nested situations, as shown in Fig. 6.8, where there are three simple and two complex predicates. To facilitate the schema design and the software management, context-free grammars were used. They provide a simple and precise mechanism to describe the rules by which sentences in written language are created starting from smaller blocks, e.g. words. In this case, the sentence stands for the SQL *where clause*. Thus, the smaller blocks and the rules that guide the sentence construction were defined. A context-free grammar[2] is formed by a finite set of states and relations amongst states, called *rules* or *productions*.

Free-context grammars help the definition of a standard structure

The resulting grammar is:

$$\begin{aligned}
&\mathbf{S} \rightarrow (\mathrm{POP})|A|\epsilon \\
&\mathrm{P} \rightarrow (\mathrm{POP})|A \\
&A \rightarrow BDC \\
&B \rightarrow k \\
&C \rightarrow v \\
&D \rightarrow > \mid >= \mid < \mid <= |! = \\
&O \rightarrow \mathrm{OR}|\mathrm{AND}
\end{aligned}$$

S is the start symbol that shows the possible alternatives for *where clause* construction: complex predicate (POP), simple predicate A or null (no filters). Complex predicates has the form POP

2 *context-free* means that the application of a rule is independent from the context, formed by previous or following states.

where two *objects* P (simple or complex predicates) are connected by AND/OR logic. For example, in the fig. 6.8 the *wrapper B* is a POP state where the right-hand P represents a simple predicate (*field3*) and the left-hand P is a complex state, that is *wrapper A*. Thus, the structure (POP) can be explicited as ((POP)OA). The wrapper A embodies a second (POP) state formed by two simple predicates (*field1* and *field2*). Thus, ((POP)OA) becomes ((AOA)OA). In this way, this grammar contemplates all the possible combinations of simple predicates, guiding a complete definition of any SQL *where clause*.
According to the grammar rules, in the XSD there are two nodes, SIMPLE and COMPLEX, that are both children of CONDITIONS node (Fig. 6.9). Both SIMPLE and COMPLEX have two ID attributes, respectively *sid* and *cid*, that are fundamental for the construction of *where* sentence. In particular, the COMPLEX node has three child nodes, F1, LOGIC and F2 that reflect the grammar POP state. The F1, F2 content is an unique identifier corresponding to a *cid*, calling another COMPLEX node, or a *sid*, referring to a SIMPLE predicate. Differently from COMPLEX, SIMPLE node coincides with the grammar simple state, BDC, representing the key-value pairs. Moreover, the 1 : n multiplicity of VALUE allows the insertion of several values for each simple predicate, linked each other by an OR logic operator.

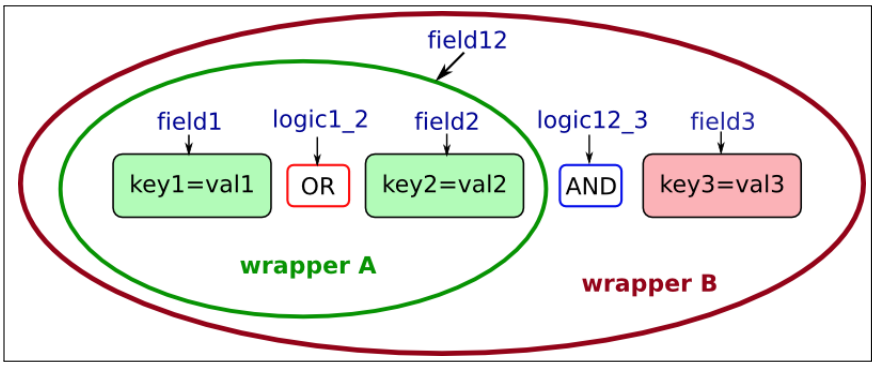


Figure 6.8: Decomposition of a *where clause* sentence. The boxes described by key_n=val_n are simple predicates, the wrappers A and B outline the complex predicates.

6.4.2 Layout definition

The second child of QUERY node is QLAYOUT. In this section, it can be possible to specify all the graphical and descriptive features of the web page. There is the possibility to set a title, an optional subtitle and a brief description about what page is used for (Fig. 6.10). Moreover, different css templates can be chosen adjusting the PTHEME tag, allowing to change also the visual properties when the database interface is configured for different projects. At present, QLAYOUT is formed by a minimal set of child nodes. In future, it is hopeful to extend the configur-

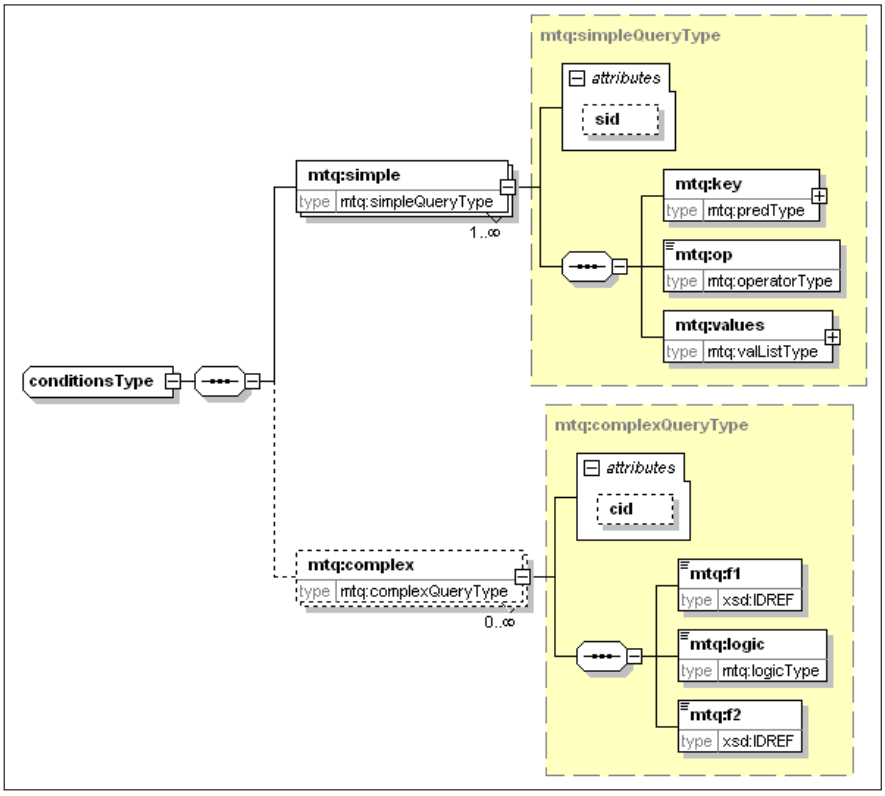


Figure 6.9: The XSD representation of SQL *where clause*. *sid* or *cid* data types are the only admitted values for F1 and F2, that are COMPLEX child nodes.

ing possibilities, providing for an extremely fine-grain web style management.

6.4.3 Database definition

QDATABASE node refers to the database connection. The web server needs some information to link to MySQL daemon, as the database location, the username and password for accessing a particular database. These data are used in every step involving a database connection, that is the querying processes that are the basement of the database interface. This node offers the possibility to set-up all the parameters necessary for database connection (Fig. 6.10).

6.5 FUTURE PERSPECTIVES

The database interface project was born from the need of the Grape consortium people to access the annotation data stored in Grape database. The first attempts consisted of a simple interface, where a fixed hard-coded database queries allowed to collect a limited number of data. The increasing requests of interface extensions required large amount of time to modify the software code, and nobody else than the code developer was able to change it. These facts brought to the decision to adopt

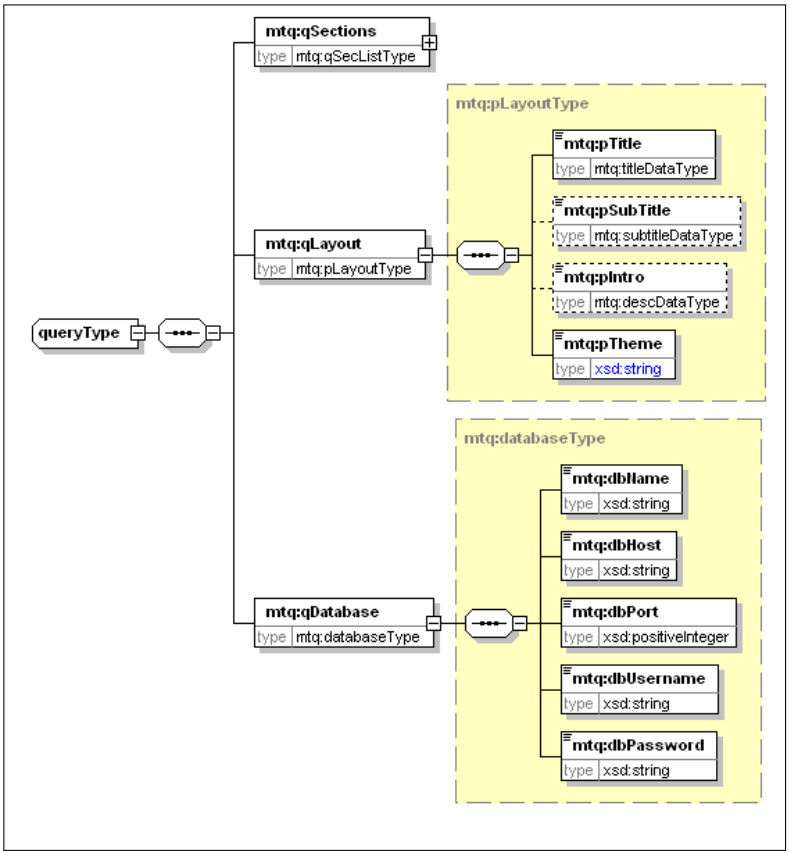


Figure 6.10: XSD structure of QLAYOUT and QDATABASE nodes.

a different approach and to study an agile system to deal with genome databases. The described database interface is the result of this need, but it is at its earlier release. A lot of work has to be done to reach a strong and solid solution. The further developments could follow three directions:

- XSD REFINEMENT: the XML schema seems to be complete, at least to define the queries used in the Grape database. However, the goal is to design a schema really general, that comprises all SQL situations. Thus, the actual XSD needs to be tested for all kind of SQL statement and, if necessary, to be extended.

- SOFTWARE ADAPTATION: the actual release of interface software bases on a previous XSD version compared with that described above. This old version is simpler and deals with minimal queries. The next step is represented by the adaptation of software code making it able to read the new XML schema. Moreover, it is desirable to completely separate the *formal* features from the *content* ones, e.g. managing the whole graphical aspects from XML file.

- WEB SERVICES IMPLEMENTATION: the database interface has been developed as a web-system. All the querying procedures are processed by a web-server and the results are published as HTML pages. This implementation approach

can represent a limit, especially when a great number of analyses has to be performed. In particular, the construction of workflows, that recursively and automatically execute some kinds of analysis, is avoided with this web implementation, that rather requires the interactive user contribution. In these cases, a system uncoupled with web environment could be more useful. The *web services* technology considers these possibilities of automation. Web services are software systems that allow the interoperability and the communication between software applications. A dedicated message system makes possible the data-exchange through the HTTP protocol between software applications written with different programming languages, realizing an efficient request-response procedure.
In this case, with a web-services approach the human interaction will be minimized and the query procedures could be iterated and processed by means of software scripts or workflow applications as Taverna [48].

A

XML

XML (eXtensible Markup Language) is a meta-language that defines and describes the *structure* of a document or information. In other words, XML is a set of syntactic rules to electronically encode documents. These rules are specified and maintained by W3C, that is the main international standards organization for the World Wide Web[1].
XML is particularly suitable for applications where data consistency and structure are essential: it was designed to transport and store data. Conceptually, XML is similar to HTML (used for viewing web pages), but XML is more content-oriented: XML was designed to carry and store data, not to display data.
XML documents are textual files, made up of storage units called entities. They contain *content* data, that represent the information to be stored in the document, and *tags*, that are markup constructs necessary for the description of the document content and logical structure. XML has not predefined tags, and one must define its own tags to built new languages.
The box below gives an example of an XML document that describes an hypothetical biological sequence.

```
1 <?xml version="1.0" encoding="UTF–8"?>
2 <sequence>
3         <name id="scf1_2">JGVv1.2</name>
4         <chromosome>14</chromosome>
5         <type>gene</type>
6 </sequence>
```

The first line is constituted by the mandatory XML declaration. The remaining lines describe the XML elements. An element is a logical component of a document and it is enclosed by a *start-tag* and an *end-tag*. The former is enclosed by "<" and ">", while the latter is enclosed by "</" and ">". Each element can include a *content* or other elements (*child elements*). The second line represent the *root* element **sequence** that contains other elements: **name**, **chromosome** and **type**. The markup construct *type* at line 5 constitutes the **type** element and its content is *gene*. A further markup construct is represented by *attribute*, that is a key-value pair present within a start-tag (attribute "id" in line 3).
Therefore, trough XML it is possible to encode any type of information, because all tags can be freely defined.

1 http://www.w3.org/

An XML document has to be *well-formed* and *valid*. It is well-formed if it satisfies a list of syntax rules, e.g. only one root element is admitted or start/end-tags have to be correctly nested. It is valid if it is complying to an associated XSD (Xml Schema Definition). XSD is a language that describes XML, defining the permitted elements, the hierarchical relations between elements and also the data-type associated to each element content. In the box above, the XSD (not showed) specifies that **sequence** element can have **name**, **chromosome** and **type** children. But **chromosome** element can not be child of **type** element.
An XML document can be represented as a tree, where elements represent tree nodes and the edges are the hierarchical relationships between nodes (Fig. A.1).

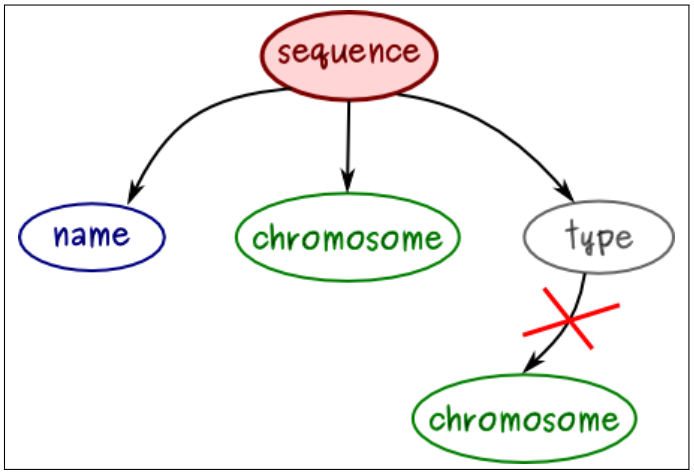


Figure A.1: XML tree representing the XML document described in the box. The XSD associated to the XML does not allow the type→chromosome relation.

BIBLIOGRAPHY

[1] Analysis of the genome sequence of the flowering plant Arabidopsis thaliana. *Nature*, 408(6814):796–815, 2000. (Cited on page 3.)

[2] The map-based sequence of the rice genome. *Nature*, 436(7052):793–800, 2005. (Cited on page 3.)

[3] The Universal Protein Resource (UniProt) 2009. *Nucleic Acids Res*, 37(Database issue):D169–74, 2009. (Cited on page 49.)

[4] J. E. Allen, W. H. Majoros, M. Pertea, and S. L. Salzberg. JIGSAW, GeneZilla, and GlimmerHMM: puzzling out the features of human genes in the ENCODE regions. *Genome Biol*, 7 Suppl 1:S9.1–13, 2006. (Cited on pages 13 and 24.)

[5] J. E. Allen, M. Pertea, and S. L. Salzberg. Computational gene prediction using multiple sources of evidence. *Genome Res*, 14(1):142–8, 2004. (Cited on page 21.)

[6] J. E. Allen and S. L. Salzberg. JIGSAW: integration of multiple sources of evidence for gene prediction. *Bioinformatics*, 21(18):3596–603, 2005. (Cited on pages 21 and 24.)

[7] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J Mol Biol*, 215(3):403–10, 1990. (Cited on page 18.)

[8] W. J. Ansorge. Next-generation DNA sequencing techniques. *N Biotechnol*, 25(4):195–203, 2009. (Cited on page 34.)

[9] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, 25(1):25–9, 2000. (Cited on page 56.)

[10] S. Avraham, C. W. Tung, K. Ilic, P. Jaiswal, E. A. Kellogg, S. McCouch, A. Pujar, L. Reiser, S. Y. Rhee, M. M. Sachs, M. Schaeffer, L. Stein, P. Stevens, L. Vincent, F. Zapata, and D. Ware. The Plant Ontology Database: a community resource for plant structure and developmental stages controlled vocabulary and annotations. *Nucleic Acids Res*, 36(Database issue):D449–54, 2008.

[11] D. L. Baillie and A. M. Rose. WABA success: a tool for sequence comparison between large genomes. *Genome Res*, 10(8):1071–3, 2000. (Cited on page 18.)

[12] D. Barrell, E. Dimmer, R. P. Huntley, D. Binns, C. O'Donovan, and R. Apweiler. The GOA database in 2009–an integrated Gene Ontology Annotation resource. *Nucleic Acids Res*, 37(Database issue):D396–403, 2009. (Cited on page 56.)

[13] D. R. Bentley et al. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, 456(7218):53–9, 2008. (Cited on page 35.)

[14] H. M. Berman, T. N. Bhat, P. E. Bourne, Z. Feng, G. Gilliland, H. Weissig, and J. Westbrook. The Protein Data Bank and the challenge of structural genomics. *Nat Struct Biol*, 7 Suppl:957–9, 2000. (Cited on page 62.)

[15] E. Birney, M. Clamp, and R. Durbin. GeneWise and Genomewise. *Genome Res*, 14(5):988–95, 2004. (Cited on page 18.)

[16] E. Blanco, G. Parra, and R. Guigo. Using geneid to identify genes. *Curr Protoc Bioinformatics*, Chapter 4:Unit 4.3, 2007. (Cited on page 14.)

[17] Terry Brown. *Genomes 3*. Garland Science, third edition, May 2006. (Cited on page 60.)

[18] M. Brudno, C. B. Do, G. M. Cooper, M. F. Kim, E. Davydov, E. D. Green, A. Sidow, and S. Batzoglou. LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. *Genome Res*, 13(4):721–31, 2003. (Cited on page 18.)

[19] K. Bryson, L. J. McGuffin, R. L. Marsden, J. J. Ward, J. S. Sodhi, and D. T. Jones. Protein structure prediction servers at University College London. *Nucleic Acids Res*, 33(Web Server issue):W36–8, 2005. (Cited on page 62.)

[20] C. Burge and S. Karlin. Prediction of complete gene structures in human genomic DNA. *J Mol Biol*, 268(1):78–94, 1997. (Cited on page 13.)

[21] M. Burset and R. Guigo. Evaluation of gene structure prediction programs. *Genomics*, 34(3):353–67, 1996.

[22] M. Burset, I. A. Seledtsov, and V. V. Solovyev. Analysis of canonical and non-canonical splice sites in mammalian genomes. *Nucleic Acids Res*, 28(21):4364–75, 2000. (Cited on page 12.)

[23] D. Campagna, A. Albiero, A. Bilardi, E. Caniato, C. Forcato, S. Manavski, N. Vitulo, and G. Valle. PASS: a program to align short sequences. *Bioinformatics*, 25(7):967–8, 2009. (Cited on page 37.)

[24] P. Carninci. Tagging mammalian transcription complexity. *Trends Genet*, 22(9):501–10, 2006.

[25] P. Carninci. Constructing the landscape of the mammalian transcriptome. *J Exp Biol*, 210(Pt 9):1497–506, 2007.

[26] P. Carninci and Y. Hayashizaki. Noncoding RNA transcription beyond annotated genes. *Curr Opin Genet Dev*, 17(2):139–44, 2007.

[27] S. Cawley, S. Bekiranov, H. H. Ng, P. Kapranov, E. A. Sekinger, D. Kampa, A. Piccolboni, V. Sementchenko, J. Cheng, A. J. Williams, R. Wheeler, B. Wong, J. Drenkow, M. Yamanaka, S. Patel, S. Brubaker, H. Tammana, G. Helt, K. Struhl, and T. R. Gingeras. Unbiased mapping of transcription factor binding sites along human chromosomes 21 and 22 points to widespread regulation of noncoding RNAs. *Cell*, 116(4):499–509, 2004.

[28] S. E. Cawley, A. I. Wirth, and T. P. Speed. Phat–a gene finding program for Plasmodium falciparum. *Mol Biochem Parasitol*, 118(2):167–74, 2001. (Cited on page 13.)

[29] J. I. Clark, C. Brooksbank, and J. Lomax. It's all GO for plant scientists. *Plant Physiol*, 138(3):1268–79, 2005.

[30] CLC bio. Signal peptides. *Bioinformatics explained*, 2006.

[31] CLC bio. HMMER. *Bioinformatics explained*, 2007. (Cited on page 51.)

[32] C. Cole, J. D. Barber, and G. J. Barton. The Jpred 3 secondary structure prediction server. *Nucleic Acids Res*, 36(Web Server issue):W197–201, 2008. (Cited on page 62.)

[33] S. R. Davies, L. W. Chang, D. Patra, X. Xing, K. Posey, J. Hecht, G. D. Stormo, and L. J. Sandell. Computational identification and functional validation of regulatory motifs in cartilage-expressed genes. *Genome Res*, 17(10):1438–47, 2007. (Cited on page 52.)

[34] A. L. Delcher, D. Harmon, S. Kasif, O. White, and S. L. Salzberg. Improved microbial gene identification with GLIMMER. *Nucleic Acids Res*, 27(23):4636–41, 1999.

[35] A. L. Delcher, A. Phillippy, J. Carlton, and S. L. Salzberg. Fast algorithms for large-scale genome alignment and comparison. *Nucleic Acids Res*, 30(11):2478–83, 2002. (Cited on page 18.)

[36] A. L. Delcher, S. L. Salzberg, and A. M. Phillippy. Using MUMmer to identify similar regions in large sequence sets. *Curr Protoc Bioinformatics*, Chapter 10:Unit 10.3, 2003. (Cited on page 18.)

[37] F. Denoeud, J. M. Aury, C. Da Silva, B. Noel, O. Rogier, M. Delledonne, M. Morgante, G. Valle, P. Wincker, C. Scarpelli, O. Jaillon, and F. Artiguenave. Annotating genomes with massive-scale RNA sequencing. *Genome Biol*, 9(12):R175, 2008. (Cited on page 26.)

[38] S. R. Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9):755–63, 1998. (Cited on page 52.)

[39] O. Emanuelsson, S. Brunak, G. von Heijne, and H. Nielsen. Locating proteins in the cell using TargetP, SignalP and related tools. *Nat Protoc*, 2(4):953–71, 2007. (Cited on page 58.)

[40] O. Emanuelsson, H. Nielsen, S. Brunak, and G. von Heijne. Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. *J Mol Biol*, 300(4):1005–16, 2000. (Cited on page 59.)

[41] R. D. Finn, J. Tate, J. Mistry, P. C. Coggill, S. J. Sammut, H. R. Hotz, G. Ceric, K. Forslund, S. R. Eddy, E. L. Sonnhammer, and A. Bateman. The Pfam protein families database. *Nucleic Acids Res*, 36(Database issue):D281–8, 2008. (Cited on page 52.)

[42] L. Florea, G. Hartzell, Z. Zhang, G. M. Rubin, and W. Miller. A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Res*, 8(9):967–74, 1998. (Cited on page 15.)

[43] D. Gilbert. Shopping in the genome market with EnsMart. *Brief Bioinform*, 4(3):292–6, 2003. (Cited on page 77.)

[44] R. Guigo, P. Flicek, J. F. Abril, A. Reymond, J. Lagarde, F. Denoeud, S. Antonarakis, M. Ashburner, V. B. Bajic, E. Birney, R. Castelo, E. Eyras, C. Ucla, T. R. Gingeras, J. Harrow, T. Hubbard, S. E. Lewis, and M. G. Reese. EGASP: the human ENCODE Genome Annotation Assessment Project. *Genome Biol*, 7 Suppl 1:S2.1–31, 2006. (Cited on page 14.)

[45] S. Guindon and O. Gascuel. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst Biol*, 52(5):696–704, 2003. (Cited on page 60.)

[46] M. A. Harris et al. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res*, 32(Database issue):D258–61, 2004.

[47] P. Horton, K. J. Park, T. Obayashi, N. Fujita, H. Harada, C. J. Adams-Collier, and K. Nakai. WoLF PSORT: protein localization predictor. *Nucleic Acids Res*, 35(Web Server issue):W585–7, 2007. (Cited on page 59.)

[48] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. R. Pocock, P. Li, and T. Oinn. Taverna: a tool for building and running workflows of services. *Nucleic Acids Res*, 34(Web Server issue):W729–32, 2006. (Cited on page 91.)

[49] N. Hulo, A. Bairoch, V. Bulliard, L. Cerutti, B. A. Cuche, E. de Castro, C. Lachaize, P. S. Langendijk-Genevaux, and C. J. Sigrist. The 20 years of PROSITE. *Nucleic Acids Res*, 36(Database issue):D245–9, 2008. (Cited on page 52.)

[50] S. Hunter et al. InterPro: the integrative protein signature database. *Nucleic Acids Res*, 37(Database issue):D211–5, 2009. (Cited on page 61.)

[51] I. Iliopoulos, S. Tsoka, M. A. Andrade, A. J. Enright, M. Carroll, P. Poullet, V. Promponas, T. Liakopoulos, G. Palaios, C. Pasquier, S. Hamodrakas, J. Tamames, A. T. Yagnik, A. Tramontano, D. Devos, C. Blaschke, A. Valencia, D. Brett, D. Martin, C. Leroy, I. Rigoutsos, C. Sander, and C. A. Ouzounis. Evaluation of annotation strategies using an entire genome sequence. *Bioinformatics*, 19(6):717–26, 2003. (Cited on pages 48 and 55.)

[52] O. Jaillon, J. M. Aury, B. Noel, A. Policriti, C. Clepet, A. Casagrande, N. Choisne, S. Aubourg, N. Vitulo, C. Jubin, A. Vezzi, F. Legeai, P. Hugueney, C. Dasilva, D. Horner, E. Mica, D. Jublot, J. Poulain, C. Bruyere, A. Billault, B. Segurens, M. Gouyvenoux, E. Ugarte, F. Cattonaro, V. Anthouard, V. Vico, C. Del Fabbro, M. Alaux, G. Di Gaspero, V. Dumas, N. Felice, S. Paillard, I. Juman, M. Moroldo, S. Scalabrin, A. Canaguier, I. Le Clainche, G. Malacrida, E. Durand, G. Pesole, V. Laucou, P. Chatelet, D. Merdinoglu, M. Delledonne, M. Pezzotti, A. Lecharny, C. Scarpelli, F. Artiguenave, M. E. Pe, G. Valle, M. Morgante, M. Caboche, A. F. Adam-Blondon, J. Weissenbach, F. Quetier, and P. Wincker. The grapevine genome sequence suggests ancestral hexaploidization in major angiosperm phyla. *Nature*, 449(7161):463–7, 2007. (Cited on pages 3 and 41.)

[53] G. E. Jordan and W. H. Piel. PhyloWidget: web-based visualizations for the tree of life. *Bioinformatics*, 24(14):1641–2, 2008. (Cited on page 61.)

[54] D. Kampa, J. Cheng, P. Kapranov, M. Yamanaka, S. Brubaker, S. Cawley, J. Drenkow, A. Piccolboni, S. Bekiranov, G. Helt, H. Tammana, and T. R. Gingeras. Novel RNAs identified from an in-depth analysis of the transcriptome of human chromosomes 21 and 22. *Genome Res*, 14(3):331–42, 2004.

[55] M. Kanehisa and S. Goto. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res*, 28(1):27–30, 2000. (Cited on page 54.)

[56] M. Kanehisa, S. Goto, M. Furumichi, M. Tanabe, and M. Hirakawa. KEGG for representation and analysis of molecular networks involving diseases and drugs. *Nucleic Acids Res*, 38(Database issue):D355–60, 2010.

[57] M. Kanehisa, S. Goto, M. Hattori, K. F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, and M. Hirakawa. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res*, 34(Database issue):D354–7, 2006.

[58] P. Kapranov, J. Cheng, S. Dike, D. A. Nix, R. Duttagupta, A. T. Willingham, P. F. Stadler, J. Hertel, J. Hackermuller, I. L. Hofacker, I. Bell, E. Cheung, J. Drenkow, E. Dumais, S. Patel, G. Helt, M. Ganesh, S. Ghosh, A. Piccolboni, V. Sementchenko, H. Tammana, and T. R. Gingeras. RNA maps reveal new RNA classes and a possible function for pervasive transcription. *Science*, 316(5830):1484–8, 2007.

[59] P. Kapranov, A. T. Willingham, and T. R. Gingeras. Genome-wide transcription and the implications for genomic organization. *Nat Rev Genet*, 8(6):413–23, 2007.

[60] A. Kasprzyk, D. Keefe, D. Smedley, D. London, W. Spooner, C. Melsopp, M. Hammond, P. Rocca-Serra, T. Cox, and E. Birney. EnsMart: a generic system for fast and flexible access to biological data. *Genome Res*, 14(1):160–9, 2004. (Cited on page 77.)

[61] W. J. Kent. BLAT–the BLAST-like alignment tool. *Genome Res*, 12(4):656–64, 2002. (Cited on page 15.)

[62] I. Korf. Gene finding in novel genomes. *BMC Bioinformatics*, 5:59, 2004. (Cited on page 14.)

[63] I. Korf, P. Flicek, D. Duan, and M. R. Brent. Integrating genomic homology into gene structure prediction. *Bioinformatics*, 17 Suppl 1:S140–8, 2001.

[64] A. Krogh, B. Larsson, G. von Heijne, and E. L. Sonnhammer. Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *J Mol Biol*, 305(3):567–80, 2001. (Cited on page 59.)

[65] S. Kurtz, A. Phillippy, A. L. Delcher, M. Smoot, M. Shumway, C. Antonescu, and S. L. Salzberg. Versatile and open software for comparing large genomes. *Genome Biol*, 5(2):R12, 2004. (Cited on page 19.)

[66] I. Letunic, T. Doerks, and P. Bork. SMART 6: recent updates and new developments. *Nucleic Acids Res*, 37(Database issue):D229–32, 2009. (Cited on page 52.)

[67] R. Li, Y. Li, K. Kristiansen, and J. Wang. SOAP: short oligonucleotide alignment program. *Bioinformatics*, 24(5):713–4, 2008. (Cited on page 37.)

[68] W. Li and A. Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–9, 2006. (Cited on page 60.)

[69] H. Lin, Z. Zhang, M. Q. Zhang, B. Ma, and M. Li. ZOOM! Zillions of oligos mapped. *Bioinformatics*, 24(21):2431–7, 2008. (Cited on page 37.)

[70] A. V. Lukashin and M. Borodovsky. GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Res*, 26(4):1107–15, 1998.

[71] W. H. Majoros, M. Pertea, and S. L. Salzberg. TigrScan and GlimmerHMM: two open source ab initio eukaryotic gene-finders. *Bioinformatics*, 20(16):2878–9, 2004. (Cited on page 13.)

[72] W. H. Majoros, M. Pertea, and S. L. Salzberg. Efficient implementation of a generalized pair hidden Markov model for comparative gene finding. *Bioinformatics*, 21(9):1782–8, 2005. (Cited on page 21.)

[73] William H. Majoros. *Methods for Computational Gene Prediction*. Cambridge University Press, September 2007. (Cited on pages 14 and 26.)

[74] E. R. Mardis. The impact of next-generation sequencing technology on genetics. *Trends Genet*, 24(3):133–41, 2008. (Cited on page 34.)

[75] M. Margulies et al. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–80, 2005. (Cited on page 34.)

[76] A. C. Martin, C. A. Orengo, E. G. Hutchinson, S. Jones, M. Karmirantzou, R. A. Laskowski, J. B. Mitchell, C. Taroni, and J. M. Thornton. Protein folds and functions. *Structure*, 6(7):875–84, 1998. (Cited on page 62.)

[77] O. Morozova and M. A. Marra. Applications of next-generation sequencing technologies in functional genomics. *Genomics*, 92(5):255–64, 2008. (Cited on page 34.)

[78] R. Mott. EST_GENOME: a program to align spliced DNA sequences to unspliced genomic DNA. *Comput Appl Biosci*, 13(4):477–8, 1997. (Cited on page 15.)

[79] L. A. Mueller et al. A snapshot of the emerging tomato genome sequence. *The Plant Genome*, 2(1):78–92, 2009.

[80] U. Nagalakshmi, Z. Wang, K. Waern, C. Shou, D. Raha, M. Gerstein, and M. Snyder. The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science*, 320(5881):1344–9, 2008. (Cited on page 34.)

[81] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–53, 1970.

[82] G. Parra, P. Agarwal, J. F. Abril, T. Wiehe, J. W. Fickett, and R. Guigo. Comparative gene prediction in human and mouse. *Genome Res*, 13(1):108–17, 2003.

[83] G. Parra, E. Blanco, and R. Guigo. GeneID in Drosophila. *Genome Res*, 10(4):511–5, 2000. (Cited on page 14.)

[84] G. Parra, K. Bradnam, and I. Korf. CEGMA: a pipeline to accurately annotate core genes in eukaryotic genomes. *Bioinformatics*, 23(9):1061–7, 2007.

[85] M. Pertea, X. Lin, and S. L. Salzberg. GeneSplicer: a new computational method for splice site prediction. *Nucleic Acids Res*, 29(5):1185–90, 2001. (Cited on page 43.)

[86] M. Pop and S. L. Salzberg. Bioinformatics challenges of new sequencing technology. *Trends Genet*, 24(3):142–9, 2008. (Cited on page 34.)

[87] P. Rice, I. Longden, and A. Bleasby. EMBOSS: the European Molecular Biology Open Software Suite. *Trends Genet*, 16(6):276–7, 2000.

[88] S. M. Rumble, P. Lacroute, A. V. Dalca, M. Fiume, A. Sidow, and M. Brudno. SHRiMP: accurate mapping of short color-space reads. *PLoS Comput Biol*, 5(5):e1000386, 2009. (Cited on page 37.)

[89] J. Schultz, F. Milpetz, P. Bork, and C. P. Ponting. SMART, a simple modular architecture research tool: identification of signaling domains. *Proc Natl Acad Sci U S A*, 95(11):5857–64, 1998.

[90] S. Schulze-Kremer. Ontologies for molecular biology and bioinformatics. *In Silico Biol*, 2(3):179–93, 2002.

[91] S. Schwartz, W. J. Kent, A. Smit, Z. Zhang, R. Baertsch, R. C. Hardison, D. Haussler, and W. Miller. Human-mouse alignments with BLASTZ. *Genome Res*, 13(1):103–7, 2003. (Cited on page 18.)

[92] J. Shendure, G. J. Porreca, N. B. Reppas, X. Lin, J. P. McCutcheon, A. M. Rosenbaum, M. D. Wang, K. Zhang, R. D. Mitra, and G. M. Church. Accurate multiplex polony sequencing of an evolved bacterial genome. *Science*, 309(5741):1728–32, 2005. (Cited on page 36.)

[93] S. A. Simon, J. Zhai, R. S. Nandety, K. P. McCormick, J. Zeng, D. Mejia, and B. C. Meyers. Short-read sequencing technologies for transcriptional analyses. *Annu Rev Plant Biol*, 60:305–33, 2009. (Cited on page 34.)

[94] D. Smedley, S. Haider, B. Ballester, R. Holland, D. London, G. Thorisson, and A. Kasprzyk. BioMart–biological queries made easy. *BMC Genomics*, 10:22, 2009. (Cited on page 77.)

[95] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–7, 1981.

[96] M. Stanke, M. Diekhans, R. Baertsch, and D. Haussler. Using native and syntenically mapped cDNA alignments to improve de novo gene finding. *Bioinformatics*, 24(5):637–44, 2008.

[97] M. Stanke and B. Morgenstern. AUGUSTUS: a web server for gene prediction in eukaryotes that allows user-defined constraints. *Nucleic Acids Res*, 33(Web Server issue):W465–7, 2005. (Cited on page 21.)

[98] M. Stanke, A. Tzvetkova, and B. Morgenstern. AUGUSTUS at EGASP: using EST, protein and genomic alignments for improved gene prediction in the human genome. *Genome Biol*, 7 Suppl 1:S11.1–8, 2006.

[99] L. Stein. Genome annotation: from sequence to biology. *Nat Rev Genet*, 2(7):493–503, 2001. (Cited on page 14.)

[100] L. D. Stein, C. Mungall, S. Shu, M. Caudy, M. Mangone, A. Day, E. Nickerson, J. E. Stajich, T. W. Harris, A. Arva, and S. Lewis. The generic genome browser: a building block for a model organism system database. *Genome Res*, 12(10):1599–610, 2002. (Cited on page 20.)

[101] B. E. Suzek, H. Huang, P. McGarvey, R. Mazumder, and C. H. Wu. UniRef: comprehensive and non-redundant UniProt reference clusters. *Bioinformatics*, 23(10):1282–8, 2007. (Cited on page 49.)

[102] The Plant Ontology Consortium. The plant ontology consortium and plant ontologies. *Comp Funct Genomics*, 3(2):137–42, 2002. (Cited on page 57.)

[103] J. D. Thompson, T. J. Gibson, and D. G. Higgins. Multiple sequence alignment using ClustalW and ClustalX. *Curr Protoc Bioinformatics*, Chapter 2:Unit 2.3, 2002. (Cited on page 60.)

[104] G. A. Tuskan et al. The genome of black cottonwood, Populus trichocarpa (Torr. & Gray). *Science*, 313(5793):1596–604, 2006. (Cited on page 3.)

[105] G. E. Tusnady and I. Simon. Principles governing amino acid composition of integral membrane proteins: application to topology prediction. *J Mol Biol*, 283(2):489–506, 1998. (Cited on page 59.)

[106] G. E. Tusnady and I. Simon. The HMMTOP transmembrane topology prediction server. *Bioinformatics*, 17(9):849–50, 2001.

[107] R. Velasco et al. A high quality draft consensus sequence of the genome of a heterozygous grapevine variety. *PLoS One*, 2(12):e1326, 2007. (Cited on page 4.)

[108] K. V. Voelkerding, S. A. Dames, and J. D. Durtschi. Next-generation sequencing: from basic research to diagnostics. *Clin Chem*, 55(4):641–58, 2009. (Cited on page 34.)

[109] C. Wei and M. R. Brent. Using ESTs to improve the accuracy of de novo gene prediction. *BMC Bioinformatics*, 7:327, 2006. (Cited on pages 14 and 21.)

[110] S. J. Wheelan, D. M. Church, and J. M. Ostell. Spidey: a tool for mRNA-to-genomic alignments. *Genome Res*, 11(11):1952–7, 2001. (Cited on page 15.)

[111] D. L. Wheeler et al. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res*, 36(Database issue):D13–21, 2008. (Cited on page 49.)

[112] B. T. Wilhelm, S. Marguerat, S. Watt, F. Schubert, V. Wood, I. Goodhead, C. J. Penkett, J. Rogers, and J. Bahler. Dynamic repertoire of a eukaryotic transcriptome surveyed at single-nucleotide resolution. *Nature*, 453(7199):1239–43, 2008. (Cited on page 34.)

[113] T. D. Wu and C. K. Watanabe. GMAP: a genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics*, 21(9):1859–75, 2005. (Cited on page 15.)