ProGenViZ: a novel interactive tool for prokaryotic genome visualization and comparison

Developed Framework

2015

Contents

List of Figures

1 Overview

ProGenViZ is an open-source freely available web tool to compare prokaryotic genomes and HTS contig data that provides an interactive way to explore genomic data and to visualize global and local relationships between genomic regions. Moreover, it provides additional features such as the re-annotation of genes, ordering of contigs against a reference and annotation of contigs by transfer from an annotated sequence.

Throughout the description of the developed framework, italic words will mark commands that can be accessed in the application or some new terms used to identify and explain certain features of the program.

Source code is available at https://github.com/B-UMMI/ProGenViZ and the tool is available at http://darwin.phyloviz.net/ProGenViZ.

2 Developed framework

2.2 Implementation

ProGenViZ was developed using a client-server approach. On the client-side we have the processing of visualization and user interaction through a web browser, while on the server-side we have all the operations leading to the creation of the basic data structures needed for visualization representation.

The Bootstrap framework was used to develop the basic structure of the web application and D3 JavaScript framework to carry out all the operations associated to the creation of visual representations and user interaction.

On the server-side, in order to process genomic data, we used Python scripts to parse all input files and convert them to JavaScript Notation Format (JSON), BLAST to search for genomic sequences, Prodigal to predict prokaryotic gene locations, and MUMmer to order contigs and find single nucleotide variations.

In the following sections we provide a more detailed description of several implementation aspects.

2.2.1 Input processing

ProGenViZ accepts three distinct file formats as input: the GenBank/EMBL format (*.gbk*), the General Feature Format (*.gff*), and the FASTA (*.fasta*) format.

To process each of these formats, we use Python scripts to create two JSON files required to create the genomic data representation and perform other tasks. One of the JSON files as information about the genomic features, while the other as the genomic sequences itself if applicable.

Because the *.gff* format does not contain genomic sequences, we offer an additional option to upload *.gff* and *.fasta* files together. When this happens, we merge the information of the genomic sequences provided by the *.fasta* file with the features provided by the *.gff*.

In the case of *.fasta* files with multiple contig sequences, an additional step is taken in the input processing to add a specific attribute to the JSON file, which uniquely marks each contig. This approach is essential to represent each individual sequence properly in the place reserved for the uploaded file in the visual representation.

2.2.2 Main work area

After the user uploads the first file, they are directed to the main work area. This area is divided into two parts: actions menu and visual representation area.

The actions menu gives access to a group of features that the user can use to explore and extract information of the uploaded files and to control some aspects of the visualization (Figure 2.1-a). The different functions of each action will be described throughout this chapter.

In the second part, the visual representation area, is where the representation of files will be displayed. The way of how genomic data is shown to users is described in the following sections.

2.2.3 Genomic data visualization

To be able to view several complete prokaryotic genomes in a single image we had to create an abstract representation of the genomic sequences and their annotations to reduce the complexity of the visualization (Figure 3.2). To do this we created two levels of abstraction.

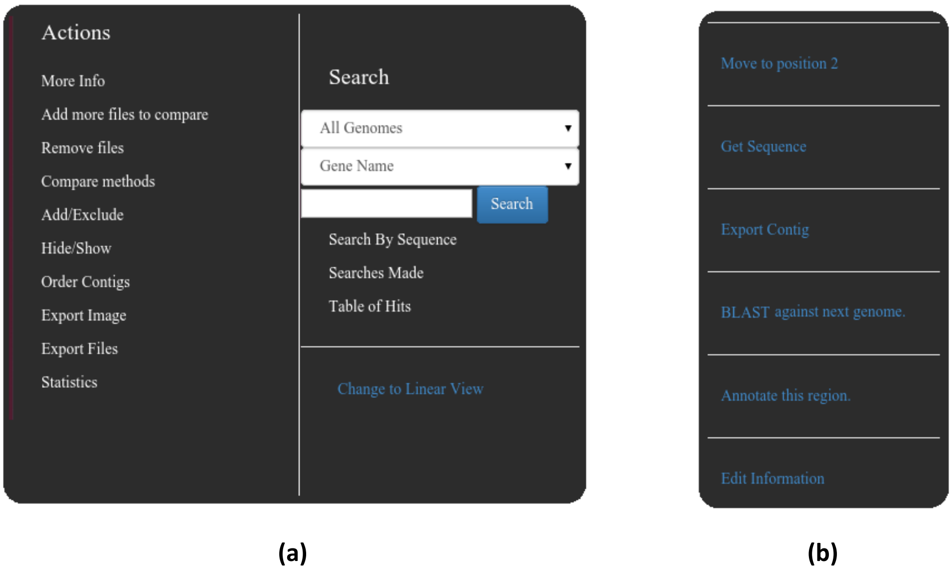


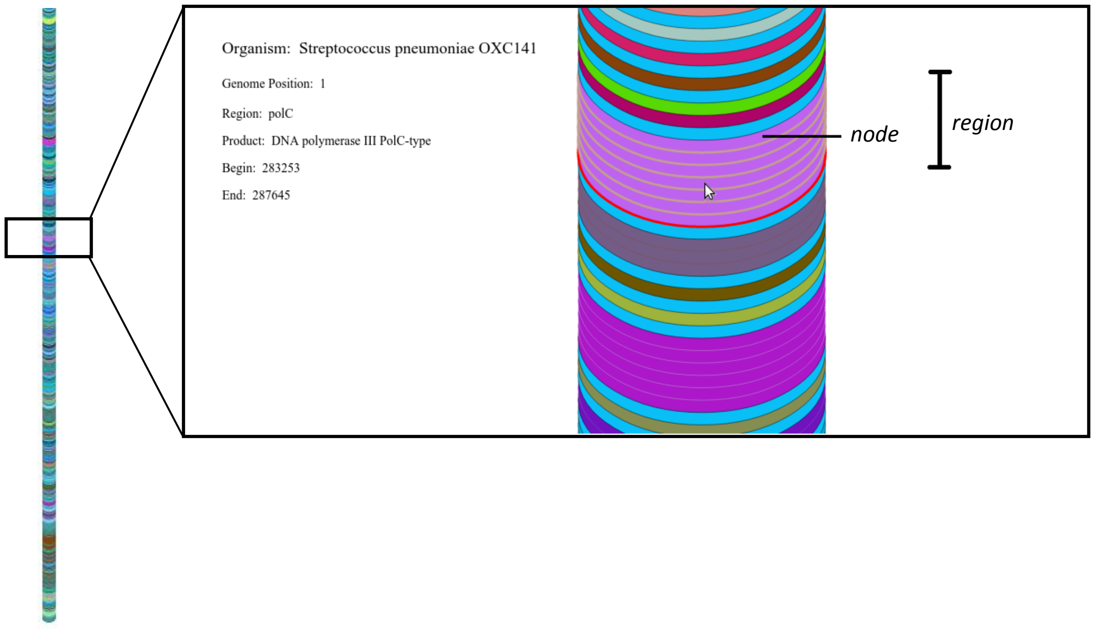
Figure 2.1: The two distinct menus of the application. (a) Actions menu. (b) Right-click menu that is activated when the user interacts with the visual representation.

First we used an approach where we divided all genomic sequences into *regions* according to their annotations. These annotations can be coding sites (CDS) and non-coding sequences that generate products such as tRNA and snRNA. Since not all *regions* of a genome are associated with an annotation, non-annotated *regions* are classified as undefined.

The second abstraction level was the division of all *regions* into intervals of 500 nucleotides, which we define as *nodes*. A *node* is thus the minimal size representation for a *region* in this tool. Therefore a *region* will be represented as many *nodes* as multiples of 500bp corresponding to its size. *Regions* with less than 500bp are still considered a single *node*. It is important to notice that what we achieve is an approximate representation of the length of the genome data and not a real one. *Nodes* are then represented as ellipses in all visual representations that are created.

We used the D3 JavaScript library as framework to develop the two ways to visualize genomic data. D3 allows to create powerful visualization components based on data and here it was used to transform all genomic data into an interactive representation.

In this tool, the representations of the genomic data and relationships between them are based on *graphs*. A *graph* is a representation of a set of objects, usually called *vertices* (singular *vertex*), where the relationships that exist between them are established by *edges* or *links*. In both visual representations developed each vertex has information about a single *node*.



Organism: Streptococcus pneumoniae OXC141

Genome Position: 1

Region: polC

Product: DNA polymerase III PolC-type

Begin: 283253

End: 287645

Figure 2.2: Representation of a part of the *Streptococcus pneumoniae OXC141* genome and definition of *node* and *region*. Different colours represent different products. In the case indicated in the image, the *region* corresponding to PolC gene consists of seven *nodes*, which means that the gene has a length between 3500-3999 base pairs. Mouse-over the region shows the information displayed on the left side of the visual representation.

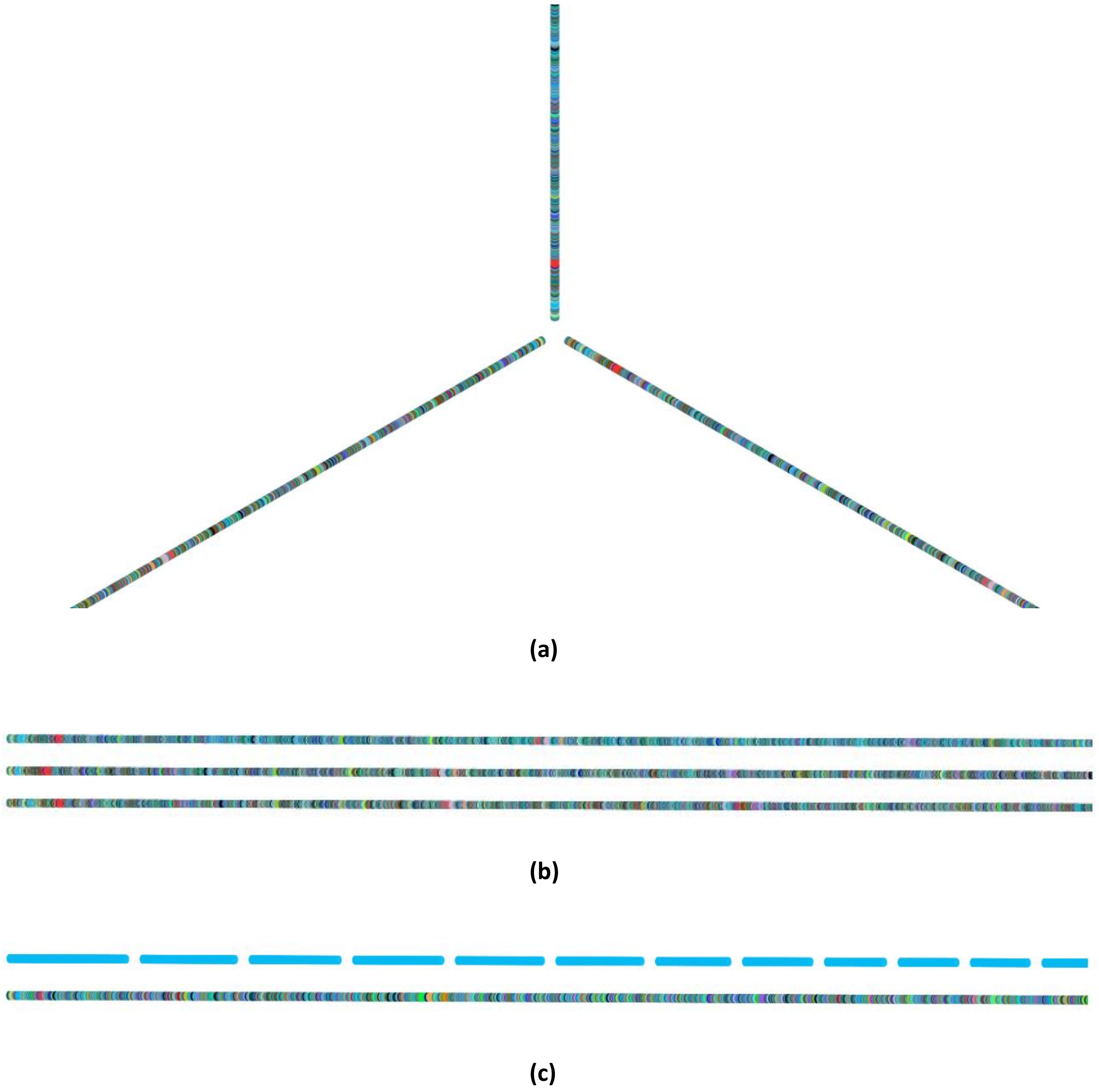
The main visual representation in ProGenViZ is based on Hive Plots (Figure 2.3-a). Hive Plots are characterized by displaying vertices in a linear layout and by clustering different vertices along a radial axis according to some property of the data.

To represent the different genomes in the Hive Plot we grouped the different *nodes* according to the file to which they belong. Depending on the number of files, the radial axis is divided into equal parts and the different linear layouts with all *nodes* belonging to a given uploaded file are disposed counter-clockwise. Near the centre of the display, we have *nodes* corresponding to the initial *regions* of the genome.

On the second visualization mode, the Linear representation (Figure 2.3-b), all *nodes* from different files are shown in a linear layout with a pre-set distance between them. *Nodes* are disposed from left to right and files added later are placed below the ones already displayed forming a stack.

On both visual representations, if a file is classified as having multi-fasta sequences, all sequences are displayed in the same linear layout but separated from each other (Figure 2.2-c) in the same order provided in the file.

The interaction by the user with the visual representation is made through zoom and padding operations, and also by mouse-over interaction with the different *regions*. When the mouse pointer hovers a particular *region*, information about it pops-up in the upper left side of



(a)

(b)

(c)

Figure 2.3: The two visual representations developed. (a) Part of the Hive Plot visual representation with 3 annotated *Streptococcus pneumoniae* genomes. (b) Part of the Linear representation with 3 annotated *Streptococcus pneumoniae* genomes. (c) Differences in the visualization of a file with contigs of *Streptococcus pneumoniae* (in blue) and an annotated genome.

the visualization area and all *regions* sharing the same annotation attribute product get highlighted (Figure 2.2). Also, by right clicking in any of the *regions*, a menu offers a series of operations than can be performed by the user (Figure 2.1-b).

Simple transitions can be made between the two visual representations by the actions menu's "*Change to Hive plot/Linear view*" option provided by the interface. Also, coupled with the ability to move between the two different views, the user has the possibility to reorder the files location in the representation by right clicking with the mouse in any of the represented files and by choosing the desired position. This feature is very important in this tool because it is what enables the comparison of *regions* from one file with any other. Queries on different *regions* are discussed in the next section.

In order to distinguish between different *regions* we designed a colour based scheme to be assigned to them. The undefined *regions* are displayed in blue and different colours are assigned to annotated *regions* according to their products. These colours are randomly generated and their total number equals to the distinct products that exist in the analysis. Moreover, the colour scheme is updated whenever the number of products increases in an analysis, which is usually the case when a new file is added.

The application provides features to highlight or remove certain *regions* of the visual representation. Since a large portion of currently annotated genomes are genes classified as hypothetical proteins, the interface provides the option to highlight or remove from the analysis all genes with products classified as hypotheticals proteins. This is done by selecting the option "*Hide/Show hypothetical proteins*" on the actions menu of the interface to highlight those genes or by selecting the option "*Add/Remove hypothetical proteins*" to remove them from the visual representation. By choosing one of the options, a search is made in client-side for *regions* classified as hypothetical proteins. Those *regions* become red and with larger ellipses if the user choose the option to visualize hypothetical proteins, or are simply removed from the analysis leaving gaps in their location if the user choose the option to remove them.

The interface also offers an option to filter the *regions* that are displayed in the visual representation. The user can perform selections by mouse dragging through the *nodes* and by selecting the "*Use Selection*" option on the mouse right-click menu. Multiple selections can be made simultaneously by pressing the *ctrl k*ey. Selections can also be removed by the user when desired. This option was developed with the purpose of avoiding information overload in the visual representation when the user performs queries, by only displaying the selected *regions* while hiding the remaining sequence.

2.2.4 Querying on genomic data

We developed two distinct approaches to obtain information from the genomic data used for analysis: by providing global statistics, and by querying on specific *regions*.

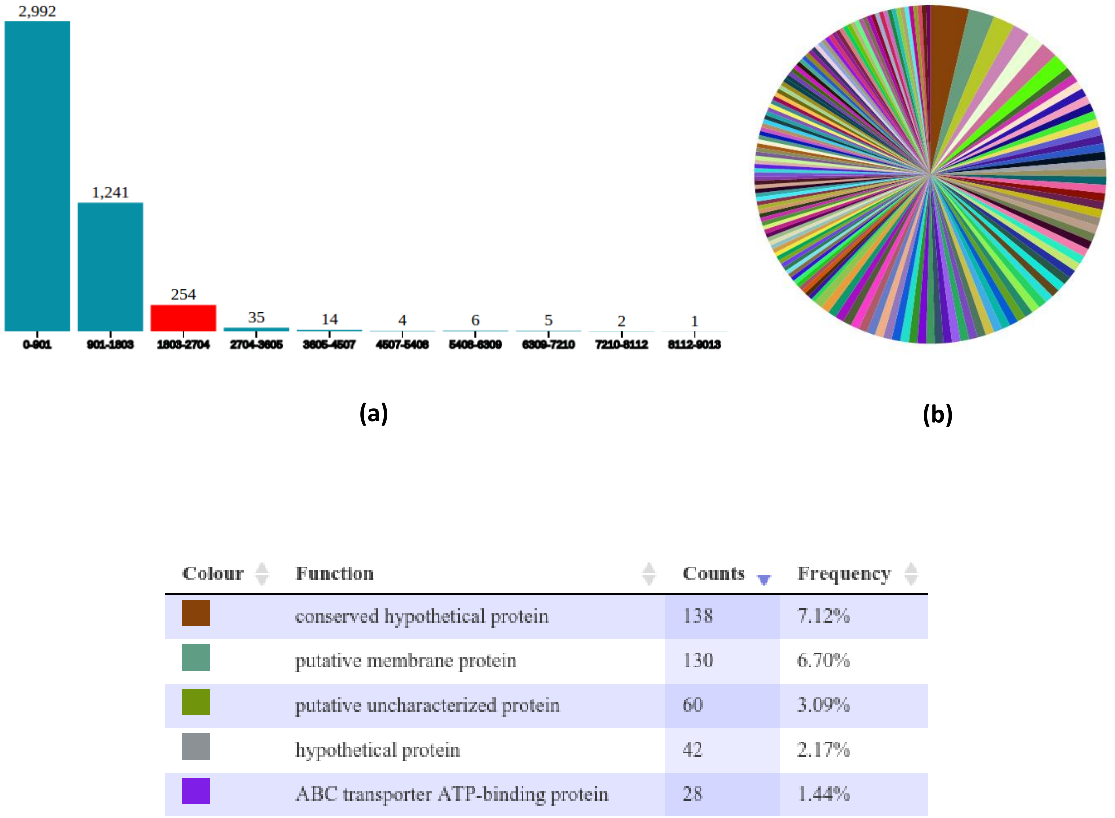
2.2.4.1 Global statistics

For every file uploaded, the user can access to general information regarding the files they uploaded (Figure 2.5-a). They can access to the total size of the genomic sequences and its percentage that is annotated by pressing the "*More Info*" button on the interface's action menu.

Is also presented information about specific genome features such as the number of transposases, the number of insertion sequences (IS), and the overall percentage of annotated genes corresponding to hypothetical proteins. Other statistics could be easily added to the interface as needed.

Other statistics can also be viewed through a representation of the distribution of the *regions'* sizes and products (Figure 2.4). The user can access to this information by selecting the "*Statistics*" button provided by the actions menu. The visual representations of these statistics are made by using a pie chart to show the distribution of *regions'* sizes and a bar chart to show the distribution of products. The colour corresponding to each product is the same as the one defined in the visual representation of genomic data. The pie and bar chart are also accompanied with a table that has information about the different products and their number of appearances in the genomic data.

The user interaction with the pie or bar chart leads to a filtration of data in the other representations. For example, when a range of sizes is chosen in the bar chart, only products in



(a)

(b)

Colour	Function	Counts	Frequency
🟫	conserved hypothetical protein	138	7.12%
🟩	putative membrane protein	130	6.70%
🟩	putative uncharacterized protein	60	3.09%
⬜	hypothetical protein	42	2.17%
🟪	ABC transporter ATP-binding protein	28	1.44%

(c)

Figure 2.4: Visual representations of the *Streptococcus pneumoniae 70585 regions*' product and size distribution. (a) Bar Chart with the *regions* size distribution. The size interval of 1808-2704 is selected. (b) Pie Chart with the products distribution in the selected size interval in the Bar Chart. Each colour represent a different product. (c) Table with the top counts of products in the selected size interval.

the selected size range are shown in the products table and in the pie chart. The reverse situation also occurs when a specific product is selected in the pie chart.

2.2.4.2 Querying on specific genomic *regions*

We developed two distinct approaches to perform queries on specific genomic *regions*: queries on annotations and sequence based queries. We also separated these queries into two different categories: *link queries*, when they establish relationships between *regions* of different files; and *basic queries* when they have only one target *region*.

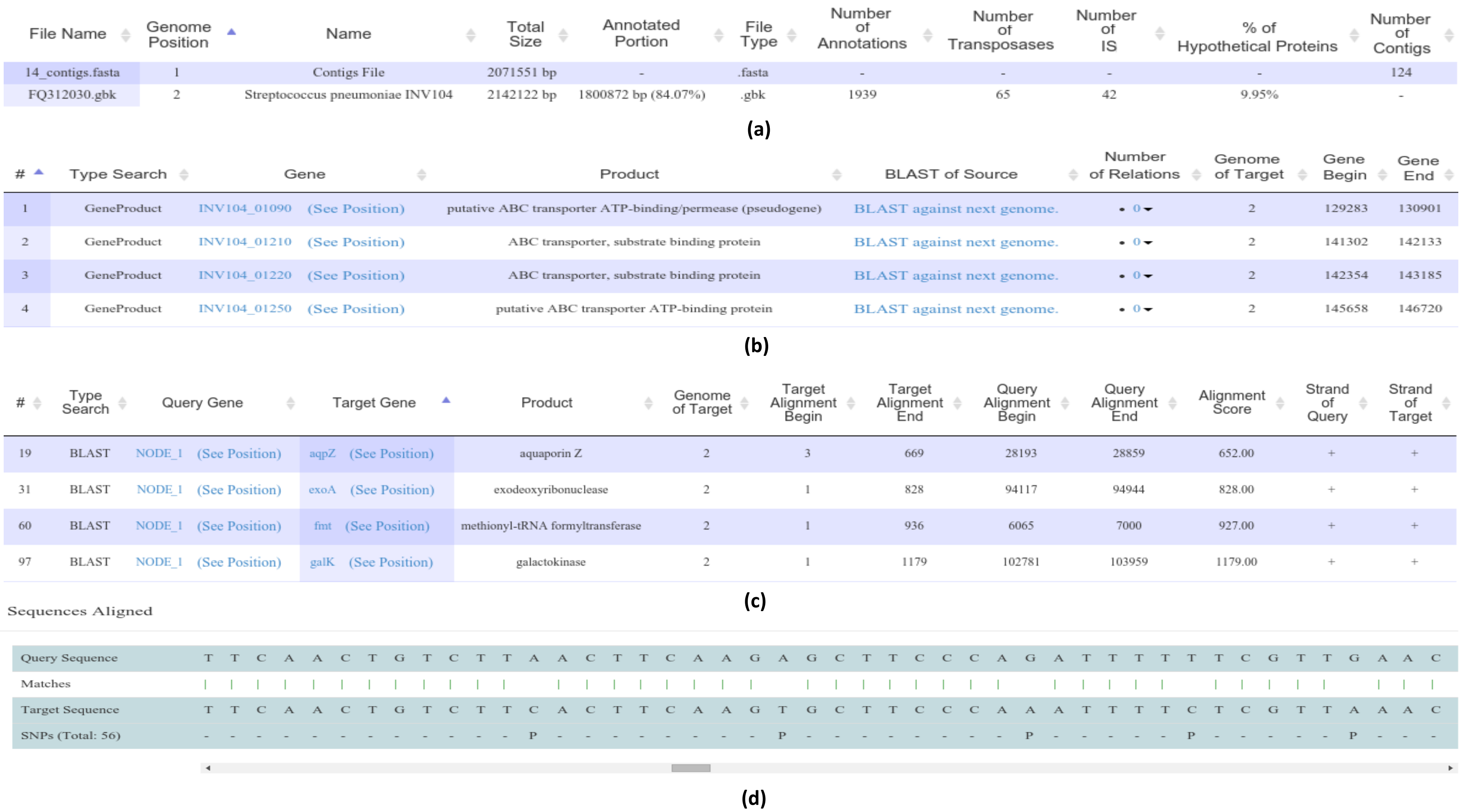
The query system was developed so that the queries are cumulative in the sense that their results can be displayed simultaneously. All queries are also stored in a list and users can remove individual queries from it in order to tailor the final displayed targets to their needs.

Queries on annotations are performed by a client-side search in the annotation attributes name, product, or location. They are made by typing keywords on the search box field on the interface's actions menu and by selecting to option to search in one or in all files represented.

Queries on annotations are considered *link queries* when a comparison method is chosen to establish relationships between annotations of different files. These comparison methods can be by name or product and they are chosen by selecting the "*Comparison methods*" option on the actions menu. When comparing by name, there is a mapping between *regions* of different files with the same name, while the comparison by product finds *regions* of different files with the same product. It should be noted that are only created links for relationships found by a given query between *regions* that are in adjacent positions in the visual representation. This approach is used to avoid overloads when viewing these comparisons. If users want to visualize relationships between annotations of files that are in remote positions in the visual representation, they first have to use the option provided by the program to change the files position in the visual representation.

Sequence based queries are performed using an internal sequence from a *region* or by using an external sequence. Internal sequence queries are considered *link queries* while external sequences queries are considered *basic queries*. In all sequence based queries, all the sequence comparisons are made by BLAST and the user can control the minimum identity and minimum score parameters for a positive match.

Internal sequence queries are made by right clicking on a specific genomic *region* in the representation and by choosing the option "*BLAST against the next position sequences*" on the menu. Because BLAST only performs pairwise comparisons, the *region* chosen by the user is only



File Name	Genome Position	Name	Total Size	Annotated Portion	File Type	Number of Annotations	Number of Transposases	Number of IS	% of Hypothetical Proteins	Number of Contigs
14_contigs.fasta	1	Contigs File	2071551 bp	-	.fasta	-	-	-	-	124
FQ312030.gbk	2	Streptococcus pneumoniae INV104	2142122 bp	1800872 bp (84.07%)	.gbk	1939	65	42	9.95%	-

(a)

#	Type Search	Gene	Product	BLAST of Source	Number of Relations	Genome of Target	Gene Begin	Gene End
1	GeneProduct	INV104_01090 (See Position)	putative ABC transporter ATP-binding/permease (pseudogene)	BLAST against next genome.	0	2	129283	130901
2	GeneProduct	INV104_01210 (See Position)	ABC transporter, substrate binding protein	BLAST against next genome.	0	2	141302	142133
3	GeneProduct	INV104_01220 (See Position)	ABC transporter, substrate binding protein	BLAST against next genome.	0	2	142354	143185
4	GeneProduct	INV104_01250 (See Position)	putative ABC transporter ATP-binding protein	BLAST against next genome.	0	2	145658	146720

(b)

#	Type Search	Query Gene	Target Gene	Product	Genome of Target	Target Alignment Begin	Target Alignment End	Query Alignment Begin	Query Alignment End	Alignment Score	Strand of Query	Strand of Target
19	BLAST	NODE_1 (See Position)	aqpZ (See Position)	aquaporin Z	2	3	669	28193	28859	652.00	+	+
31	BLAST	NODE_1 (See Position)	exoA (See Position)	exodeoxyribonuclease	2	1	828	94117	94944	828.00	+	+
60	BLAST	NODE_1 (See Position)	fmt (See Position)	methionyl-tRNA formyltransferase	2	1	936	6065	7000	927.00	+	+
97	BLAST	NODE_1 (See Position)	galK (See Position)	galactokinase	2	1	1179	102781	103959	1179.00	+	+

(c)

Sequences Aligned

Query Sequence	T	T	C	A	A	C	T	G	T	C	T	T	A	A	C	T	T	C	A	A	G	A	G	C	T	T	C	C	C	A	G	A	T	T	T	T	T	T	C	G	T	T	G	A	A	C
Matches	|	|	|	|	|	|	|	|	|	|	|	|		|	|	|	|	|	|	|	|		|	|	|	|	|	|	|	|		|	|	|	|	|		|	|	|	|	|		|	|	|
Target Sequence	T	T	C	A	A	C	T	G	T	C	T	T	C	A	C	T	T	C	A	A	G	T	G	C	T	T	C	C	C	A	A	A	T	T	T	T	C	T	C	G	T	T	A	A	A	C
SNPs (Total: 56)	-	-	-	-	-	-	-	-	-	-	-	-	P	-	-	-	-	-	-	-	-	P	-	-	-	-	-	-	-	-	P	-	-	-	-	-	P	-	-	-	-	-	P	-	-	-

(d)

Figure 2.5: The different query results tables. (a) More Info table. Has global information about the uploaded files. (b) *Hits Table* with results from queries by annotations. Has information about the results' name, product and position of the file in the visual representation as well as the location of the results in the sequence. (c) *Hits table* with sequence based query results. Shows the alignment location of the query sequence with the reference and its score. (d) HSP aligned table. Show the alignment of a high-similarity pair and their SNPs, at nucleotide level, of any results after a sequence based query.

compared with the genomic sequences that are immediately in the adjacent position in the representation.

Choosing the “*Search by Sequence*” option on the interface gives the option to perform external sequence queries. A file in the visual representation needs to be chosen to act as reference for a BLAST search and an external nucleotide sequence coupled with a name to identify it must be inserted to act as query.

In the next section we describe the developed ways to visualize the results of these queries on specific *regions*.

2.2.5 Visualizing results of queries on specific genomic *regions*: *Hits table* and representation modification

Results obtained after performing queries on specific *regions* are represented in a *Hits table* and by specific modifications in the visual representation of the *nodes*. However, there are some differences in the information that is shown when the user performs *basic* or *link queries*.

The *Hits table* provides text information about the queries results (Figure 2.5-b, c). It can be accessed after performing any query by choosing the “*Hits Table*” option on the actions menu. The table is fully customizable, being possible to organize all columns and filter the entries. This is achieved through the use of the DataTables plugin for JQuery JavaScript library which adds advanced interaction controls to HTML tables.

The *Hit table* also has a feature to allow user interaction between the table and the visual representation. This is achieved by using the "*See position*" field in the *Hits table*, which instantaneously directs the user to the location of the result’s *region* in the visual representation. This operation is carried out by the use of the SVG coordinates to centre the image at the specific point, which leads to a precise focus at the desired *region* in the visualization.

Regarding the visual representation, a new colour is assigned to each query and all query results are represented by a size increment of the ellipses corresponding to the *regions* affected.

When the user performs a *link query*, there are some additional information displayed in both the *Hits table* as in the visual representation. These differences are described below.

2.2.5.1 Link querying: stablishing links and results table modification

When the user performs *link queries*, relationships between different *regions* are displayed through links in the visual representation (Figure 2.6). They are developed by assigning

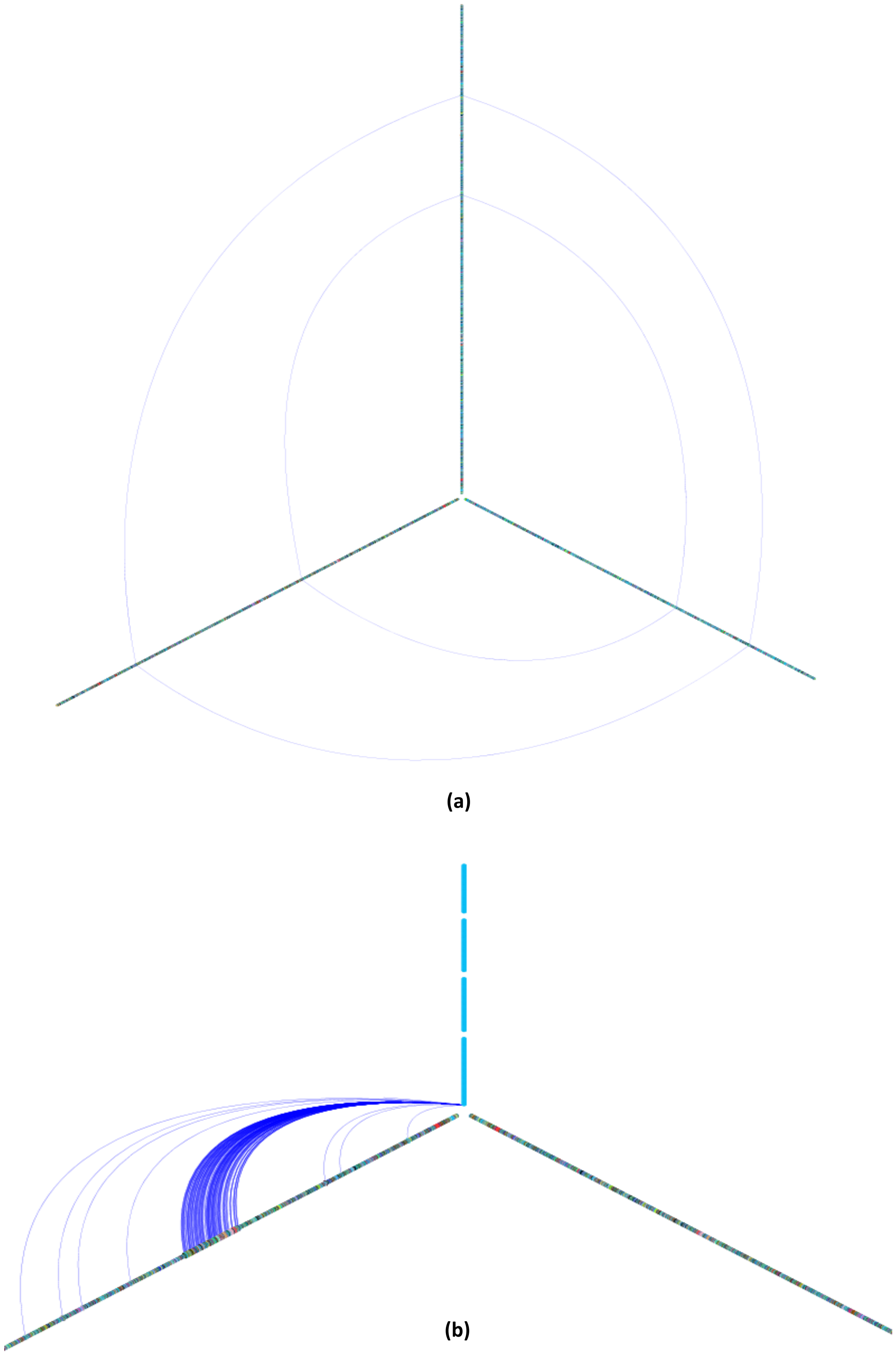


Figure 2.6: Links between *regions* of different files. (a) Established links after choosing the comparison by gene name method and search by name for 2 genes (*aroE*, *xpt*) in 3 *Streptococcus pneumoniae* strains. (b) Part of the visual representation of an internal sequence based query of a contig of *Streptococcus pneumoniae* against an annotated genome. It is possible to verify the localization of the contig in the annotated genome.

the category of source and target to different *regions*, which are used as a start and end point to draw a path line between the two *regions* using D3.

The *Hits table* also as some additional fields that are specific from *link queries* (Figure 2.4-c). For each *link queries* results is shown the information about the name, product and location of the *regions* that functioned as source and target to create the link. In the case of *link queries* involving BLAST alignments, information about the sequence alignment locations and their respective scores is also shown.

2.2.6 Visualizing sequence alignment at nucleotide level

We have already shown two ways to visualize sequence alignment results: by the visual representation in the way of links and by the results in the *Hits table*. Another developed feature allows the representation of BLAST sequence alignments at the nucleotide level and highlights Single Nucleotide Polymorphisms (SNPs) between sequences (Figure 2.5-d).

After the user makes a query that produces results obtained by BLAST, he can choose to visualize the alignment at the nucleotide level by right clicking in one of the results of the *Hits table* and by selecting the option to "*view HSPs aligned*". By choosing this option, the high-similarity pairs (HSPs) obtained by BLAST are shown aligned with the representation of matches and gaps. The number of SNPs and their location is also shown after running the *show-snps* utility of the MUMmer software at the server-side.

2.2.7 Operations with contigs

ProGenViZ provides different features which can be applied to contigs data: order contigs against a reference and contig annotation. These two features are described in detail below.

2.2.7.1 Ordering contig data

Contigs prevenient of assembly of HTS data are usually available in multi-fasta files. One feature that ProGenViZ allows users to do is to load these files and order them against any *.fasta* or *.gbk* file loaded on the interface.

To proceed with the ordering of contigs, the user must choose the "*Order contigs*" option in the actions menu and also a contigs file to act as query and other file to act as reference. After performing all these steps, two *.fasta* files are created. One *.fasta* file with the reference sequence and other with the contigs sequences. They are used as input for the alignment software NUCmer inserted in the MUMmer software package, using the programs' default parameters. The NUCmer results are then filtered through the *delta-filter* algorithm which filters the alignment results of NUCmer, using as default parameters a minimum identity of 98% and a minimum alignment size of 300 nucleotides. These parameters used to filter the alignment results can be changed by the user.

After running the contig ordering option, the resulting order is displayed in the visual representation. It is important to notice that this alignment feature provided by the program aims only to show the relative position between contigs and also act as a filter to display only those contigs that align against the reference sequence under study.

2.2.7.2 Single contig annotation

Contig data is mainly available in multi-fasta format and therefore, no annotation information is provided in this format. In order to allow contig annotations, we implemented an annotation process, which transfers features from a loaded annotated sequence to a target contig (Figure 2.7).

To perform the annotation of a given contig, we use a combination of BLAST to search for homologous genes in an annotated sequence and Prodigal to predict gene locations in the contigs. To annotate a contig the user must first perform an internal sequence query against the annotated reference by right clicking on the contig and choosing the "*BLAST against the next position sequences*" option on the menu. Then, to annotate it, it is necessary to right click on the contig again and choose the "*Annotate this Region*" option. The selection of the annotations that are transferred to the contig is made by combining the results obtained by BLAST and Prodigal. An annotation is transferred only if the start and end positions of an alignment of a *region* from the reference with the contig are the same or include the Prodigal's predicted start and end locations of a CDS in the contig (Figure 2.7-b). Because Prodigal only predicts genes, annotations that do not correspond to a CDS are not passed to the contigs. The annotation procedure can be repeated for each contig loaded in the interface.

2.2.7.3 FASTA file annotation

In addition to the single contig annotation, ProGenViz also offers a feature to annotate *.fasta* files with single sequences or multiple contigs sequences, using an annotated *.gbk* file as reference.

By choosing the "*Annotate*" option on the action menu, the user can then select a query file, a reference file, and an annotation method to perform the annotation. The two possible methods are:

- **BLAST Score Ratio (BSR) method** – In this method, Prodigal is used to predict CDS locations in the query nucleotide sequences to build the query proteome. Then, the BLAST raw score for each reference peptide against itself is stored as a reference score. Each Reference peptide is then compared to each peptide in the query proteome and the BSR is calculated by dividing the best query score by the reference score of that peptide. By default, only the annotations from peptides with a BSR greater than 0.6 are passed to the query *.fasta* file.

- **BLASTN method** – This method consists in the same procedure than the single contig annotation, but only in one step, for an unlimited number of individual sequences in a *.fasta* file. BLASTN of the query sequences against the reference is performed and then Prodigal is used to predict coding sites in the reference sequence. Annotations are transferred by combining the results obtained by BLASTN and Prodigal with the annotation being made only if the start and end positions of an alignment of a *region* from the reference with the contig are the same or include the Prodigal's predicted start and end locations of a CDS in the contig. Because Prodigal only predicts genes, annotations that do not correspond to a CDS are not passed to the contigs.

3.2.8 Editing gene annotations

Designing ways to establish relationships between *regions* of different files also provides the possibility of monitoring the quality of annotations by sequence similarity. In order to be able to change the information of *regions* that are found to be poorly annotated, an option to perform changes in the pre-existing name and product of a *region* was developed. The user can edit an annotation by right clicking on the desired *region* and by selecting the "*Edit information*" option on the menu. The modification of pre-existing annotations is made by modifying the name or product of a *region* in the JSON file with the genomic features. These changes can then be exported using functionalities provided by ProGenViZ.

2.2.9 Exporting Data

There are several different data types that can be exported from ProGenViZ: results presented in tables, images, specific genomic sequences, specific contigs and whole files. The

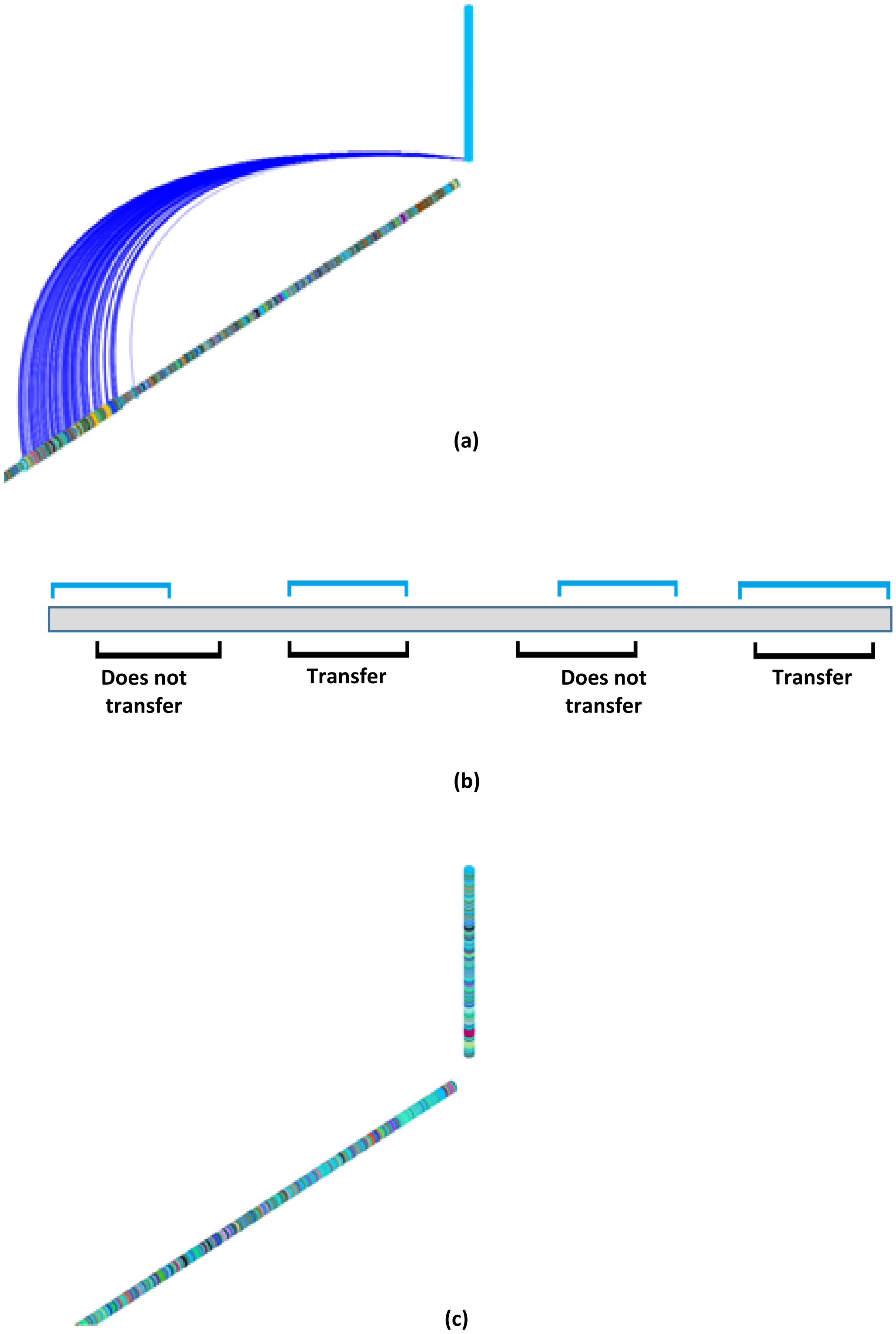


Figure 2.7: The single contig annotation process. (a) Internal sequence query of a contig against a reference. (b) Transfer of annotation by combining the BLAST results and Prodigal. In grey is shown the representation of the contig's sequence. In blue is shown the BLAST alignment result of a *region* from the reference sequence with the contig. In black is represented the CDS predicted by Prodigal in the contig. (c) Annotated contig after choosing the option to *"Annotate this Region"* on the right-click menu.

export functionality is very important in all data analysis software because it enables the use of results obtained from this tool in other software to allow further analysis.

The export of results from tables can be made through an option that is provided on their upper right corner when they are shown. The results can be exported in Comma Separated Values (CSV) and PDF format.

Images can also be exported as the current visual representation. This is done through the "*Export Image*" option of the actions menu. Images can be exported in PNG, high-resolution PNG and PDF formats.

Another option developed is the export of sequences associated with specific *regions*. This export option is made by right clicking on any of the *regions* in the visual representation and by choosing the "*Get Sequence*" option on the menu. Consequently, a *.fasta* file is generated and presented with the sequence of the chosen *region*.

Specific contigs can also be exported. This is made by right clicking on the desired contig and by choosing the "*Export contig*" option in the menu. In that time are generated two files at server-side, one *.fasta* file with the contig sequence and one GFF with its annotations, if any, which the user is given the option to download.

It is also possible to export all information of *regions* and genomic sequences for each file loaded in the interface. If changes occurred in the representation of a file of contigs that led to the annotation of some, the use of this option will only exports the annotated contigs. This type of export is done through the "*Export files*" option in the actions menu and by choosing the location of the file in the visual representation to export. After selecting a file to export we provide a combination of a *.fasta* file with the genomic sequences and a GFF file with the annotations.