GenePainter - Documentation

Authors

Björn Hammesfahr, Florian Odronitz, Stefanie Mühlhausen, Stephan Waack, Martin Kollmar

How to cite

Björn Hammesfahr†, Florian Odronitz†, Stefanie Mühlhausen, Stephan Waack & Martin Kollmar (2013)
GenePainter: a fast tool for aligning gene structures of eukaryotic protein families, visualizing the alignments and mapping gene structures onto protein structures. *BMC Bioinformatics* **14**, 77.

http://www.biomedcentral.com/1471-2105/14/77 Open Access Highly accessed

Contact

Dr. Martin Kollmar
Group Systems Biology of Motor Proteins
Department of NMR-based Structural Biology
Max-Planck-Institute for Biophysical Chemistry
Am Faßberg 11
37077 Göttingen

mako@nmr.mpibpc.mpg.de

Homepage

http://www.motorprotein.de/genepainter.html

Licence

Table of contents

Introduction

The conservation of intron positions comprises information useful for de novo gene prediction, protein sequence alignment improvement, and for analyzing the origin of introns. Here, we present GenePainter, a standalone tool for mapping gene structures onto protein multiple sequence alignments (MSA). Gene structures, as obtained for example by using WebScipio (http://www.webscipio.org), are aligned with respect to the exact positions of the introns (down to nucleotide level) and intron phase. The output can be visualized in various formats, ranging from plain text to complex graphical figures.

Installation

Unpack

Use one of the following methods, depending on the archive file type:

```
$ unzip gene_painter.zip
$ tar -xzf gene_painter.tgz
```

Compilation

No compilation required.

Ruby version

Ruby version 1.9.2 or higher is required. If necessary, consider using Ruby Version Manager (https://rvm.io/; RVM) to install and work with multiple ruby environments on your machine.

Usage

```
$ ruby gene_painter.rb -i <alignment> -p <yaml-files> \
[<options>]
```

Option	Description
-i	Multiple protein sequence alignment in FASTA format.
-p	Path to the directory containing the gene structures in YAML-format.

A more detailed description of the MSA and the YAML files as well as the incorporation of both by GenePainter is given in the Input section.

Ruby interpreter

Invoke GenePainter via one of the following options:

1. As a script

```
$ ruby gene_painter.rb
```

2. As a program

```
$ ./gene_painter.rb
```

Important note

GenePainter assumes your ruby interpreter to be located at `/usr/local/bin/ruby`. This assumption is coded in the first line of `gene_painter.rb` (Shebang). To meet your local requirements, please edit the Shebang and change the specified path to the correct one. For example, with the Ruby interpreter located at `/usr/bin/ruby`, the Shebang should look like this: `#!/usr/bin/ruby`.

Example

```
$ ruby gene_painter.rb \
-i test_data/coronin_alignment.fas  -o coronin \
-p test_data/coronin_genes -svg 1000 500 extended \
-pdb test_data/2AQ5.pdb -pdb_prot HsCoro1A
```

Options

Option	Description
-o <project_name>	Base name of the output file(s) (default: 'genepainter').
-a	Output the alignment file with additional lines containing intron phases.
-n	Mark introns by intron phase instead of the vertical bar "|".
-phylo	For phylogenetic analysis: Mark exons and introns by "0" and "1", respectively.
-s	Mark exons by " " instead of "-" and introns by "|".
-svg width height	Create an SVG-file of size <width> x <height>.
[extended |	Use this parameter to create a more detailed svg.
normal |	Use this parameter to create the normal svg (default).

Option	Description
reduced]	Use this parameter to create an svg focused on common introns.
-start X	Alignment position to start (default: position 1).
-stop Y	Alignment position to stop (default: last position).
-pdb file.pdb [chain]	Two scripts for execution in PyMol will be provided. In color_exons.py the consensus exons are colored and in color_splicesites.py the splice junctions of the consensus exons are marked for <chain> (default: chain A).
-pdb_prot prot_name	Use protein <prot_name> as reference for alignment with the pdb sequence Default: First protein in <alignment>.
-f	Force alignment between pdb and first protein sequence of the MSA or protein <prot_name> (if specified). This ignores the default that intron positions will only be mapped if the alignment score > 70%.
-consensus value	Color only intron positions conserved in <value> percent of all genes (default: 80%).
-ref_prot_struct	Color only the intron positions occurring in the gene of the reference protein. May be combined with "-consensus".
-penalize_endgaps	Penalize gaps at the end of the alignment (behaves like the standard Needleman-Wunsch algorithm). Default: gaps at the end of the alignment are not penalized.

Input

GenePainter expects two types of input:

1. a FASTA-formatted multiple sequence alignment (MSA);
2. a folder containing gene structures in YAML format as specified by WebScipio.

GenePainter combines information from the alignment with gene structures. Therefore, the protein names from the MSA are matched with YAML filenames. Only those genes, which can be matched (i.e. protein name equals the YAML filename), will be analysed.

Multiple protein sequence alignment

This file must be a multiple protein sequence alignment, in which all sequences are of same length. Protein sequences are matched with the gene structures on the basis of the FASTA header and file names, respectively. To this end, the FASTA header must be exactly like the corresponding YAML filename for each gene, which should be included in the analysis. For this reason, FASTA headers must not contain any blanks or special characters.

YAML

For the analysis, GenePainter needs gene structure information for each gene. This information must be stored in a specific file format, the YAML-format as generated for example by WebScipio (http://www.webscipio.org). A minimal working example is shown in Figure 1. Moreover, all gene structures should be located in the same directory.

The most convenient way to obtain YAML-formatted files is to use the WebScipio web interface for gene reconstruction and to download the resulting YAML files. For automation of the YAML generation, several scriptable alternatives exist. First, WebScipio can be accessed by the web service API. This can be done within any software program. In the GenePainter package, a script, `gene_scan.rb`, is included for accessing WebScipio through the web service and storing the resulted YAML file locally. A brief introduction into the usage of the web service can be found at the WebScipio homepage (http://www.webscipio.org/webscipio/web_service). Second, one can also download the Scipio command line tool from the web interface.

Usage of `gene_scan.rb`

```
$ ruby tools/gene_scan.rb \
'Species name' \
fasta_sequence.fas \
gene_structure.yaml
```

Arguments

Option	Description
Species name	A list of possible species names can be found on http://www.diark.org or http://www.diark.org/api_species.xml
fasta_sequence.fas	A file containing the query sequence in fasta format
gene_structure.yaml	Name of the output file

Structure of YAML files

Scipio and WebScipio store gene structure information in YAML format. This format comprises a collection of key – value pairs, an associative array. However, the accurate gene structure representation requires more keys than necessary for the alignment of the gene structures. Thus, GenePainter ignores some data included in the YAML files. Accordingly, these additional keys need not be included in manually reconstructed YAML files. A minimal working example YAML file is defined in Figure 1. An exhaustive description of all keys used by WebScipio can be found at the WebScipio homepage (http://www.webscipio.org/help/scipio#description).

Figure 1 - Excerpt from the YAML file describing HsCoro1A. All exon and intron descriptions within the very first and last ones have been omitted (marked in yellow). Blank lines were added to separate exon and intron descriptions. Additionally, green boxes highlight exons and blue boxes highlight introns. Only those key – value pairs, which are relevant for GenePainter are shown. The original YAML file is part of the test data included in the package.

The list of exons and introns ("matchings") must start with the keyword "`- matchings:`". The order of keys describing the exons and introns is not important. Mandatory keys are listed in the following tables:

YAML keys	Description
ID	The id of the BLAT hit.
status	Might be "auto", "complete", "partial", "incomplete" or "manual". "complete" means that Scipio had no problems locating the query; "partial" means that the hit is on one of multiple targets each matching a part of the query; "incomplete" means that Scipio could not completely match the query sequence to the target; another reason for "incomplete" is a missing stop-codon in the target sequence following the last amino acid of the query

YAML keys	Description
	sequence. "manual" can be entered if the output was modified by hand.
type	"intron", "intron?", "exon", or "gap". "intron?" is used for uncertain introns (unusual splice patterns found)
nucl_start	Location in the query (in nucleotide coordinates).
dna_start	Location in the target.
dna_end	Location in the target.
seq	DNA sequence of the feature.

YAML keys that appear only in exons	Description
nucl_end	Location in the query (in nucleotide coordinates).
prot_start	The location transformed into residue coordinates rather than nucleotides. A remainder of 1 is rounded down, and 2 is rounded up.
prot_end	The location transformed into residue coordinates rather than nucleotides. A remainder of 1 is rounded down, and 2 is rounded up.
translation	Translation of the aligned part of the DNA sequence.

Meaning of the parameters

The following figures illustrate some of GenePainters output formats and options. All figures were generated with test data comprising coronin genes as included in the archive `gene_painter.zip`.

Text-based output

The basic output format is a plain text-file where exons are represented as minus signs and introns as vertical bars (Figure 5A). A more detailed output including intron phases can be obtained by using the `-n` option (Figure 5C). By using the `-s` option (Figure 5B), only introns are represented by "|". Moreover, intron phases can be included as additional lines in the given alignment (Figure 6A;

option `-a`), or an alignment based on the presence `1` and absence `0` of introns for further phylogenetic analyses can be generated (Figure 6B; option `–phylo`).

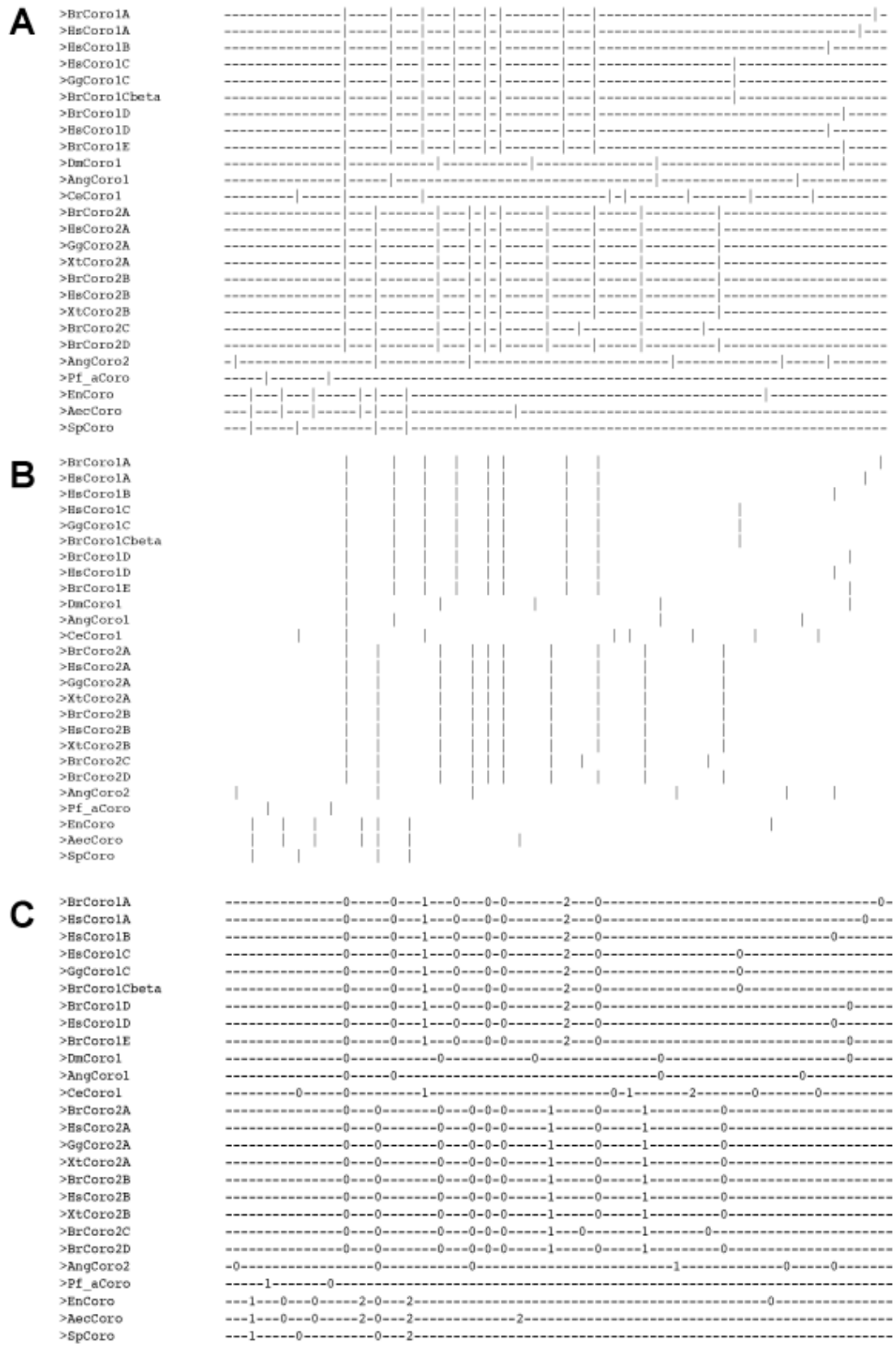

Figure 2 - Basic output formats of GenePainter.

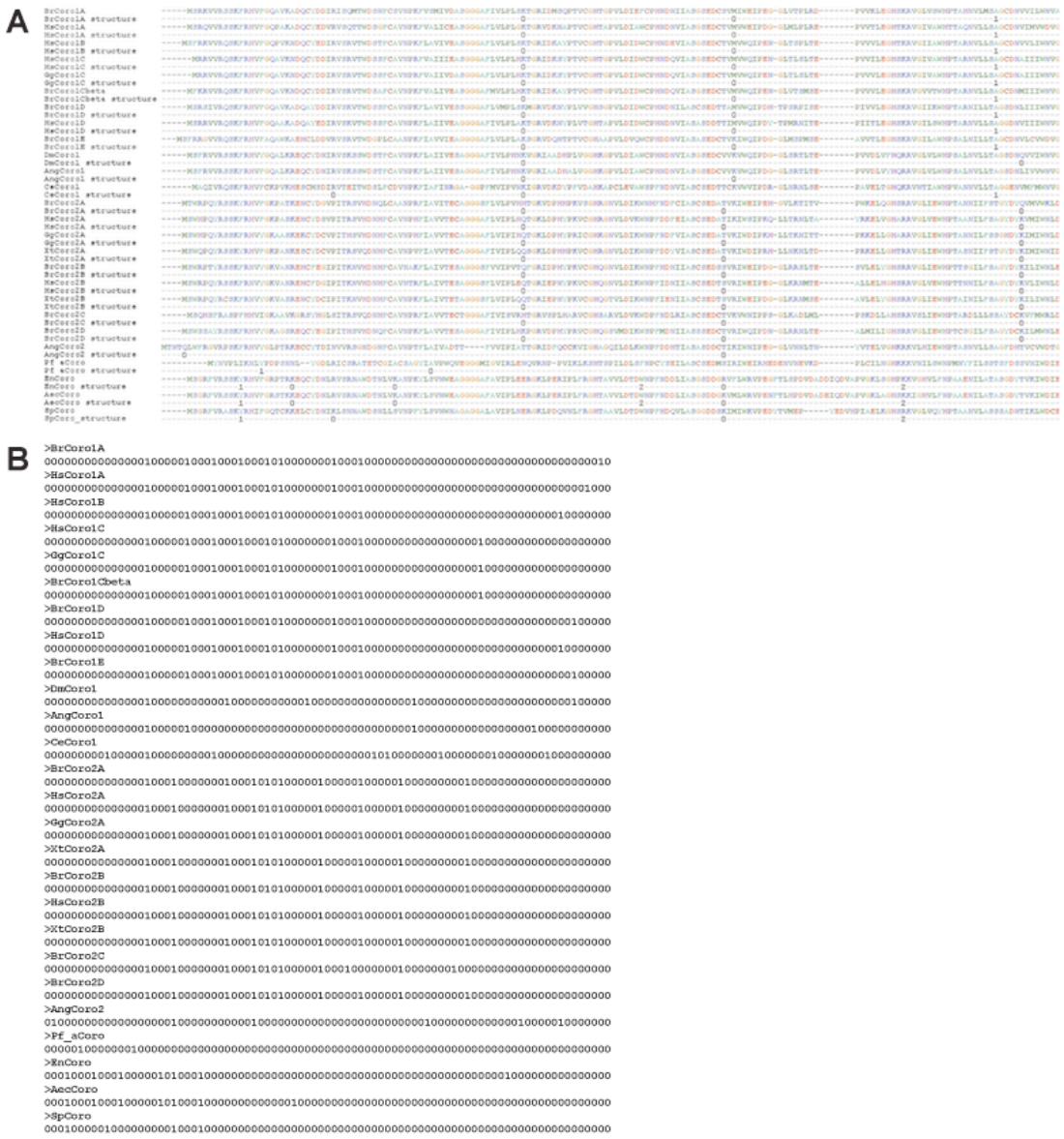

Figure 3 - Intron phases included in given alignment.

Graphical output

Apart from text based output formats, a graphical output can be generated. To this end, SVG parameters need to be set with `-svg <width> <height> [extended|normal|reduced]`. Figure 7A illustrates a detailed SVG of size 1000 x 500 pixel (`extended` is set). In contrast, the basic SVG (same size, but with `normal` set) is shown in Figure 7B. In Figure 7C the `reduced` SVG is shown, which highlights common introns. In order to restrict the algorithm to a certain part of the alignment, first and last positions to be considered can be set by `-start` and `-stop`, respectively. Figure 8 pictures a detailed SVG based on only part of the given alignment (domain of interest, alignment positions `1-612`).

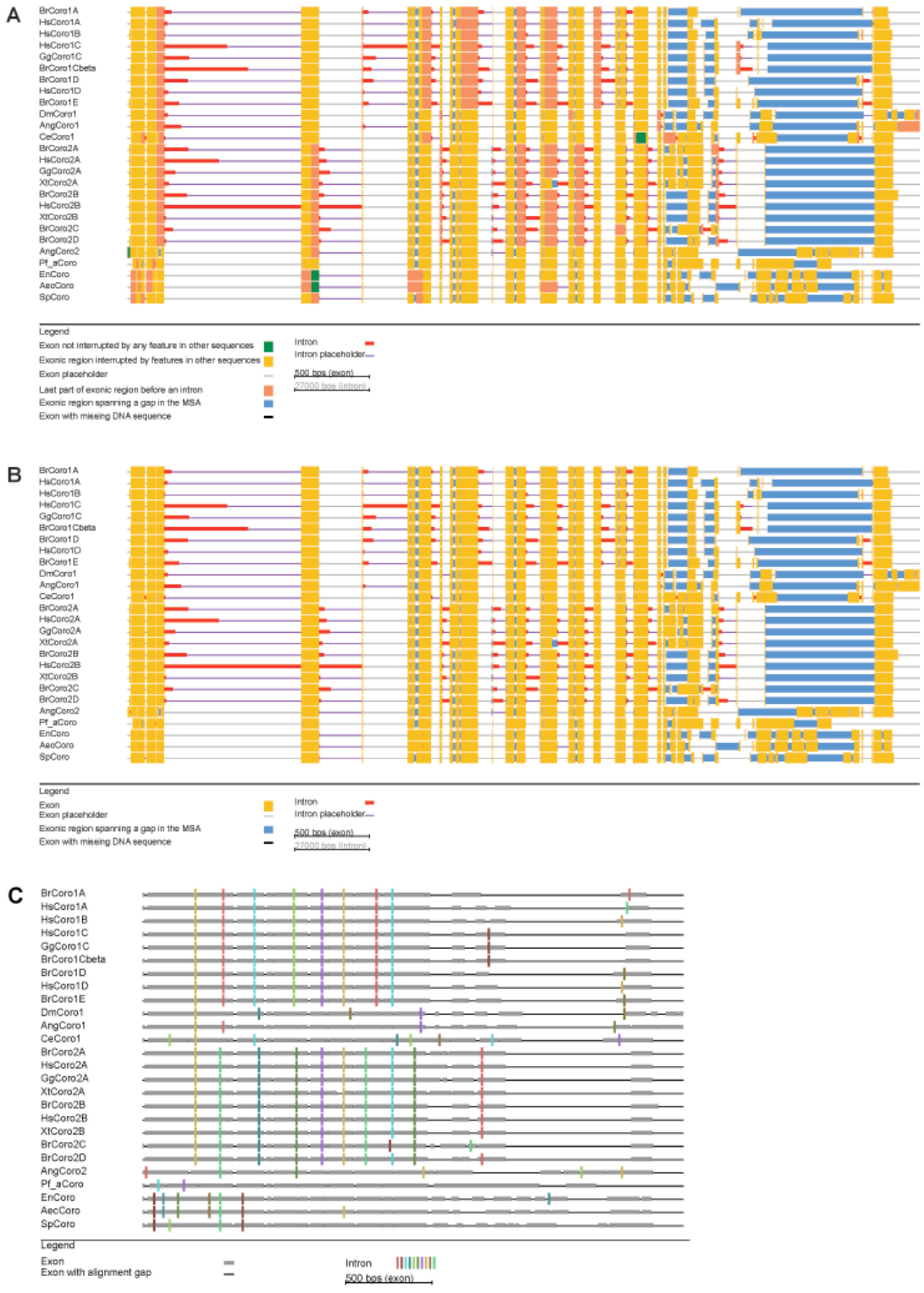

Figure 4 - Graphical output of GenePainter.

Figure 5 - Detailed graphical output covering only domain of interest.

Gene structures mapped to protein structures

Additionally, if a pdb file is specified via `-pdb`, intron positions and phases are mapped onto protein structure. Figure 9A demonstrates mapping of the exons of the human coronin HsCoro1A gene (`-pdb_prot HsCoro1A`) onto the protein structure of mouse coronin MmCoro1A (the pdb file is part of the test data set, `-pdb test_data/2Aq5.pdb`). While for this figure all exons that are conserved in at least 80% of all proteins are considered (default), Figure 9B displays all exons present in the reference sequence (`-ref_prot_struct`). Accordingly, splice sites are shown in Figures 9C and 9D. In this output, attention is drawn to intron phases. A three-color scheme and numbers denote phases.

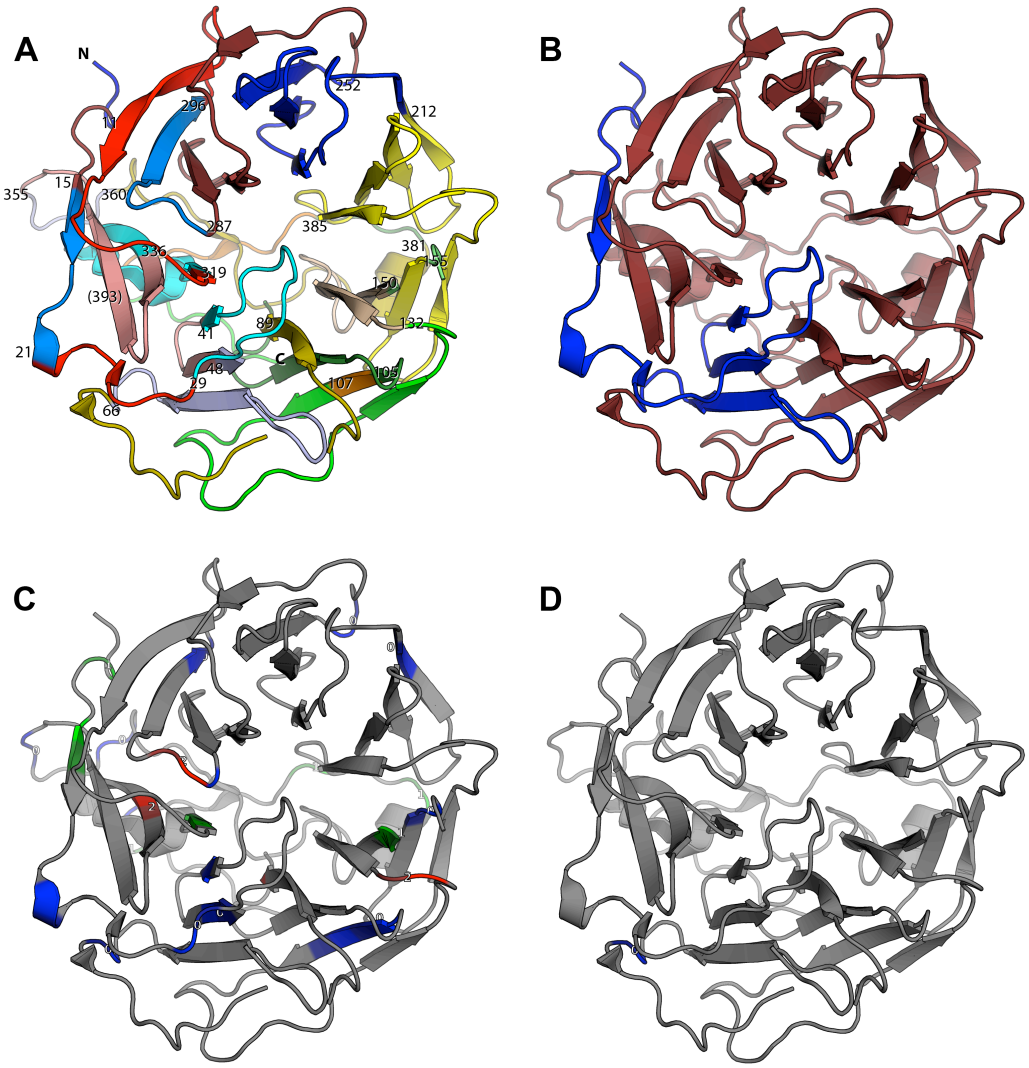

Figure 6 - Mapping of conserved exons (9A, 9B) and intron position and phase (9C, 9D) onto protein structure.

Part of the underlying algorithm is the calculation of a global alignment between reference and pdb sequence. Although this alignment is a simple implementation of the Needleman-Wunsch algorithm, some adjustments are done. In detail, gaps at the end of the alignment are not penalized. This is particular useful, as pdb sequence and reference sequence may vary in length. Alignments with and without free end gaps are opposed in Figure 10.

```
HsCoro1A:
MSRQVVRSSKFRHVFGQPAKADQCYEDVRVSQTTWDSGFCAVNPKFVALICEASGGGAFLVLPLGKTGRVDKNAPTVCGHTAPVLDIAWCPHNDNVIASGSEDCTVMVWEIPDGGLM
LPLREPVVTLEGHTKRVGIVAWHTTAQNVLLSAGCDNVIMVWDVGTGAAMLTLGPEVHPDTIYSVDWSRDGGLICTSCRDKRVRIIEPRKGTVVAEKDRPHEGTRPVRAVFVSEGKI
LTTGFSRMSERQVALWDTKHLEEPLSLQELDTSSGVLLPFFDPDTNIVYLCGKGDSSIRYFEITSEAPFLHYLSMFSSKESQRGMGYMPKRGLEVNKCEIARFYKLHERRCEPIAMT
VPRKSDLFQEDLYPPTAGPDPALTAEEWLGGRDAGPLLISLKDGYVPPKSRELRVNRGLDTGRRRAAPEASGTPSSDAVSRLEEEMRKLQATVQELQKRLDRLEETVQAK

2AQ5.pdb, global end gap free alignment
-------SSKFRHVFGQPAKADQCYEDVRVSQTTWDSGFCAVNPKFMALI-EASGGGAFLVLPLGKTGRVDKNVPLV-GHTAPVLDIAW-PHNDNVIASGSEDCTVMVWEIPDGGLV
LPLREPVITLEGHTKRVGIVAWHPTAQNVLLSAG-DNVILVWDVGTGAAVLTLGPDVHPDTIYSVDWSRDGALICTSCRDKRVRVIEPRKGTVVAEKDRPHEGTRPVHAVFVSEGKI
LTTGFSRMSERQVALWDTKHLEEPLSLQELDTSSGVLLPFFDPDTNIVYLCGKGDSSIRYFEITSEAPFLHYLSMFSSKESQRGMGYMPKRGLEVNK-EIARFYKLHERKCEPIAMT
VPRKSDLFQEDLYPPTAGPDPALTAEEWLGGRDAGPLLISLKDGYVPPKSR------------------------------------------------------------

2AQ5.pdb, global alignement without end gap free option
-------SSKFRHVFGQPAKADQCYEDVRVSQTTWDSGFCAVNPKFMALI-EASGGGAFLVLPLGKTGRVDKNVPLV-GHTAPVLDIAW-PHNDNVIASGSEDCTVMVWEIPDGGLV
LPLREPVITLEGHTKRVGIVAWHPTAQNVLLSAG-DNVILVWDVGTGAAVLTLGPDVHPDTIYSVDWSRDGALICTSCRDKRVRVIEPRKGTVVAEKDRPHEGTRPVHAVFVSEGKI
LTTGFSRMSERQVALWDTKHLEEPLSLQELDTSSGVLLPFFDPDTNIVYLCGKGDSSIRYFEITSEAPFLHYLSMFSSKESQRGMGYMPKRGLEVNK-EIARFYKLHERKCEPIAMT
VPRKSDLFQEDLYPPTAGPDPALTAEEWLGGRDAGPLLISLKDGYVPPK------------------------------S---------------------R--------
```

Figure 7 - Aligned sequences.