

# GenePainter 2 - Documentation

---

## Authors

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

## How to cite

### **Stefanie Mühlhausen, Marcel Hellkamp & Martin Kollmar (2014)**

GenePainter 2.0 resolves the taxonomic distribution of intron positions. unpublished.

and

### **Björn Hammesfahr<sup>†</sup>, Florian Odronitz<sup>†</sup>, 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>

## 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](mailto:mako@nmr.mpibpc.mpg.de)

## Homepage

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

## Licence

*Licence:* GenePainter can be downloaded and used under a GNU General Public License.

*Any restrictions to use by non-academics:* Using GenePainter by non-academics requires permission.

## Table of contents

<b>Authors</b> .....	<b>1</b>
How to cite .....	1
Contact .....	1
Homepage.....	1
<b>Licence</b> .....	<b>1</b>
<b>Table of contents</b> .....	<b>2</b>
<b>Introduction</b> .....	<b>3</b>
<b>Installation</b> .....	<b>3</b>
Unpack .....	3
Compilation.....	3
Ruby version .....	3
<b>Usage</b> .....	<b>3</b>
Ruby interpreter .....	4
Using GenePainter under windows.....	4
<b>Example</b> .....	<b>4</b>
<b>Options</b> .....	<b>4</b>
Text-based output format.....	4
Graphical output format.....	5
Taxonomy and statistics.....	5
Data and output selection options .....	6
General options .....	7
<b>Input</b> .....	<b>7</b>
Multiple protein sequence alignment.....	7
Gene structure files .....	7
<b>Meaning of the parameters</b> .....	<b>11</b>
Text-based output.....	11
Graphical output.....	14
Gene structures mapped to protein structures.....	15
Intron gain and loss over taxonomy .....	16
Taxonomy and statistics.....	17
Data and output selection.....	19

## 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. GenePainter is 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 genePainter.zip
$ tar -xzf genePainter.tgz
```

### Compilation

No compilation required.

### Ruby version

Ruby version 2.0 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 genePainter.rb -i <alignment> -p <yaml-files> \
[<options>]
```

Option	Description
<code>--input &lt;file&gt;</code> <b>-i</b>	Multiple protein sequence alignment in FASTA format.
<code>--path &lt;path&gt;</code> <b>-p</b>	Path to the directory containing the gene structures in YAML- or GFF-format.

A more detailed description of the MSA and the YAML/GFF 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 genePainter.rb
```

2. As a program

```
$ ./genePainter.rb
```

### Important note

GenePainter assumes that `/usr/bin/env ruby` points to the RVM ruby interpreter. While this will work on most UNIX systems, it might not work on windows machines. In case you encounter any `"/usr/bin/env ruby: No such file or directory"` errors, please edit the very first line of script `genePainter.rb` 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`.

## Using GenePainter under windows

GenePainter was developed for UNIX and Mac. To use GenePainter under Windows, you might consider installing Cygwin, which comes with a bash and appropriate GNU tools. Please make sure to have GNU tools `grep` and `tar` installed and in your executable path when using GenePainter with its taxonomy options. If not, you should unpack the NCBI taxonomy dump yourself and add the parameter `--no-grep` to read the taxonomy dump into memory instead of grepping its content. In order to use GenePainter with the parameter `--tree`, please make sure to have `python` installed and in your executable path.

## Example

```
$ ruby genePainter.rb \  
-i example/coronin_alignment.fas -o coronin \  
-p example/coronin_genes --svg 500 1000 \  
--pdb example/2AQ5.pdb --pdb-ref-prot HsCoro1A \  
--intron-phase
```

## Options

### Text-based output format

In the standard output format, exons are marked by `"-"` and introns by `"|"`.

Option	Description
<b>--intron-phase</b>	Mark introns by intron phase instead of by " ".
<b>--phylo</b>	For phylogenetic analysis: Mark exons by "0" and introns by "1"
<b>--spaces</b>	Mark exons by blank (" ") instead of "-" and introns by " ".
<b>--no-standard-output</b>	Suppress standard output-format. Default: standard output is always generated
<b>--alignment</b>	Output the alignment file with additional lines containing intron phases.
<b>--consensus &lt;N&gt;</b>	Additional line marking introns occurring in N percent of all genes. Specify N as decimal number between 0 and 1.
<b>--merge</b>	Additional line marking all introns present in any gene.

### Graphical output format

Option	Description
<b>--svg</b>	Create an svg-file of all gene structures
<b>--svg-format &lt;FORMAT&gt;</b>	Format is one of "normal", "reduced" and "both". Use "normal" to create a detailed svg (default). Use "reduced" to create an svg focused on common introns. Use "both" to create both svg files.
<b>--pdb &lt;FILE&gt;</b>	Two scripts for execution in PyMol will be provided. The scripts will colour the consensus exons and splice junctions of consensus exons. Combine with "--consensus", "--merge" or "--pdb-ref-prot-struct" to specify, which introns should be considered.
<b>--pdb-chain &lt;CHAIN&gt;</b>	Mark gene structure for chain CHAIN. Default: chain A
<b>--pdb-ref-prot-struct</b>	Colour only the intron positions occurring in the gene specified by "--pdb-ref-prot".
<b>--pdb-ref-prot &lt;PROT&gt;</b>	Use protein PROT as reference for alignment with pdb sequence. Default: first protein in the input alignment
<b>--tree</b>	Generate newick tree file and SVG representation

### Taxonomy and statistics

Option	Description
<b>--statistics</b>	Provides a plain text file with information about

	each intron. Specify "--taxonomy" and "--taxonomy-to-fasta" to include taxonomic information.
<b>--taxonomy &lt;FILE&gt;</b>	FILE is database dump of NCBI taxonomy .
<b>--taxonomy-to-fasta &lt;FILE&gt;</b>	Text-file defining the matching between fasta header and NCBI species Format: Gene1[,Gene2]:Species
<b>--taxonomy-common-to &lt;X,Y,Z&gt;</b>	Mark introns common to taxa X,Y,Z in an additional line. List must consist of at least one NCBI taxon.
<b>--[no-]exclusively-in-taxa</b>	Mark introns occurring (not) exclusively in taxa specified by "--taxonomy-common-to".
<b>--introns-per-taxon</b>	Mark all introns newly gained in each <i>last common ancestor</i> of supplied species by intron phase. Provides a plain text file.
<b>--no-grep</b>	Read the NCBI taxonomy dump into RAM. This will require some hundred MBs of RAM additionally. Default: taxonomy dump is parsed with 'grep' calls.

### Data and output selection options

Option	Description
<b>--data-selection</b>	Use only a selection of all data, i.e. gene structures. All output is based on the intron positions occurring in the selection.
<b>--analyse-data-selection-only</b>	Ignore all data that are not in selection. Effects only taxonomy options: If not specified, all data are used for taxonomic analysis.
<b>--output-selection</b>	Text-based and graphical output contains only selected data. Analysis of intron positions is based on all data.

### Selection criteria for data and output selection

Option	Description
<b>---selection-based-on-regex &lt;"REGEX"&gt;</b>	Specify a regular expression (without the enclosing "/" and any modifiers). It will be applied on gene structure names.
<b>---selection-based-on-list &lt;X,Y,Z&gt;</b>	List of gene names to be used.
<b>---selection-based-on-species &lt;"SPECIES"&gt;</b>	Use all genes associated with that species. Specify also --taxonomy-to-fasta to map gene structure file names to species names

## General options

Option	Description
<b>--outfile &lt;FILE&gt;</b>	<b>-o</b> Base name of the output file(s). Default: "genepainter"
<b>--range &lt;START,STOP&gt;</b>	Restrict genes to alignment range START-STOP
<b>--[no-]delete-range</b>	Delete everything within (outside) the specified range.
<b>--keep-common-gaps</b>	Keep alignment gaps common to all sequences.
<b>--no-best-position-introns</b>	Plot introns at alignment gap always onto beginning of the gap. Default: Align introns separated by gaps.
<b>--[no-]separate-introns- in-textbased-output</b>	(Not) Separate each consecutive pair of introns by an exon placeholder in text-based output formats. Default: Separate introns unless the output lines get too long.
<b>--help</b>	<b>-h</b> Show help message.

## 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 or in GFF v.3 format.

GenePainter combines information from the alignment with the gene structures. Therefore, the protein names from the MSA must match to the YAML or GFF filenames. Only those genes, which can be matched (i.e. protein name equals the YAML or GFF 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 or GFF filename for each gene, which should be included in the analysis. For this reason, FASTA headers must not contain any blanks or special characters.

### Gene structure files

For the analysis, GenePainter needs gene structure information for each gene. This information must be stored either in the YAML-format as generated by

WebScipio (<http://www.webscipio.org>) or in GFF v.3 format. Moreover, all gene structures must be located in the same directory.

## GFF

GFF3 formatted gene structures are parsed for “CDS” features. All “CDS” features are considered, which are linked via the attribute column (*parent tag*) to the first “mRNA” feature specified. If no mRNA feature is given in the GFF file, all CDS features are parsed. GenePainter uses column 1 (*seqid*), columns 4 and 5 (*start* and *end of feature*) and column 8 (*phase*). All other features and columns are ignored. GFF files generated by WebScipio can also be used as input, although they do not strictly follow GFF3 conventions.

## YAML

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 its web service API. This can be done within any software program. In the GenePainter package, scripts are included for querying WebScipio with genes belonging to a single species (`generate_yaml_for_species.rb`) or with genes belong to different species (`generate_yaml_for_multiple_species.rb`). Both scripts access WebScipio through the web service and store the resulting YAML files locally. A brief introduction to the usage of the web service can be found at the WebScipio homepage ([http://www.webscipio.org/webscipio/web\\_service](http://www.webscipio.org/webscipio/web_service)). Second, Scipio can be used locally, which requires further software (BLAT, Bioperl, YAML Perl module) and respective genome assembly files.

A list of all species available can be found at  
[http://www.webscipio.org/webscipio/genome\\_list](http://www.webscipio.org/webscipio/genome_list)

### Usage of `generate_yaml_for_species.rb`

```
$ ruby tools/ generate_yaml_for_species.rb \  
-s 'Species name' -i fasta_sequence.fas
```

### Mandatory Arguments

Option		Description
<code>--species</code> <code>&lt;'species_name'&gt;</code>	<code>-s</code>	Species encoding the specified protein(s). Species should be wrapped with " ' " to preserve spaces.
<code>--input &lt;file&gt;</code>	<code>-i</code>	Path to fasta-formatted protein sequence(s). Might be a multiple sequence alignment of sequences encoded by same species.

### Options

Option		Description
<code>--outfile</code>	<code>-o</code>	Name of the YAML output file. ONLY used if the

<b>&lt;file_name&gt;</b>		fasta file contains only one sequence. Default: Use fasta-header of the input protein sequence.
<b>--help</b>	<b>-h</b>	Show help message.

### Usage of generate\_yaml\_for\_multiple\_species.rb

```
$ ruby tools/ generate_yaml_for_multiple_species.rb \
-s example/fastahaders2species.txt -i fasta_sequence.fas
```

### Mandatory Arguments

Option		Description
<b>--species-to-fasta &lt;file&gt;</b>	<b>-s</b>	Text-based file mapping fasta header to species names. Mandatory line-format: Fastaheader1[,Fastaheader2]:Species
<b>--input &lt;file&gt;</b>	<b>-i</b>	Path to fasta-formatted multiple sequence alignment.

### Options

Option		Description
<b>--help</b>	<b>-h</b>	Show help message.

YAML files will be named like the corresponding fasta-headers. YAMLs are only generated for those sequences, for which a species is specified.

### 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>).

```

1  ---
2  - matchings:
3      type: exon
4      nucl_start: 0
5      nucl_end: 198
6      dna_start: -72598366
7      dna_end: -72598168
8      prot_start: 0
9      prot_end: 66
10     translation: MSRQVVRSSKFRHVFVGGQPAKADQCYEDVRVVSQTWDSGFCVAVNPKFVALICEASGGGAFLVPLGK
11     seq: atgagccggcagggtggctccgctccagcaagtccgccacgtgttggacagccggccaaggccgaccagtgctatgaagatgtgcgcgctc
12
13     type: intron
14     nucl_start: 198
15     - dna_start: -72598168
16     dna_end: -72596978
17     seq: atqaaccctaaqacccctaaqaaaaaacactcctccaccaqccatgaactctatqcaagtccctaattagatqacagtaccagaaccttcc
18     ...
19     type: intron
20     nucl_start: 1065
21     - dna_start: -72594999
22     dna_end: -72594999
23     seq: tcggacctgttccaggaggacctgtaccacccaccgcaggcccgaccctgcctcacggctgaggagtggctgggggggtcgggatgctg
24
25     type: exon
26     nucl_start: 1281
27     nucl_end: 1383
28     dna_start: -72594716
29     dna_end: -72594614
30     prot_start: 427
31     prot_end: 461
32     translation: DAVSRLEEEMRKLQATVQELQKRLDRLEETVQAK
33     seq: gatgccgtgtctcggctggaggaggagatgcggaagctccaggccacgggtgcaggagctccagaagcgcttgacaggctggaggagacag
34 ID: 720
35 status: auto

```

**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 for successful incorporation in GenePainter are listed in the following tables:

YAML keys	Description
<b>type</b>	"intron", "intron?", "exon", or "gap". "intron?" is used for uncertain introns (unusual splice patterns found)
<b>nucl_start</b>	Location in the query (in nucleotide coordinates).
<b>seq</b>	DNA sequence of the feature.

YAML keys that appear only in exons	Description
<b>nucl_end</b>	Location in the query (in nucleotide coordinates).

## Structure of file mapping fasta header to species names

```

1 HsCoro1A, HsCoro1B, HsCoro1C, HsCoro1D, HsCoro2A, HsCoro2B: Homo sapiens
2 GgCoro1C: Gallus gallus
3 GgCoro2A: Gallus gallus
4 DmCoro1: Drosophila melanogaster
5 ...
6

```

**Figure 2** - For taxonomy options as well as to generate YAML files, a mapping between genes described in MSA and corresponding species must be established. To this end, fasta header (separated by “,”) are mapped to species names. Fasta header and species names are separated by “:”.

Genes that are linked to the corresponding species are considered for taxonomic computations or generation of YAML files, respectively. All other genes are omitted from these computations. Species names must match NCBI taxonomy.

## Meaning of the parameters

The following figures illustrate some of GenePainters output formats and options. All figures were generated with the test data comprising coronin genes as included in the archive `genePainter.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 3A). Using the `--spaces` option (Figure 3B), shows only introns. A more detailed output including intron phases can be obtained by using the `--intron-phase` option (Figure 3C). Options `--merge` and `--consensus` are applied to all generated output.

In all text-based output formats, consecutive introns are by default (or by setting `--separate-introns-in-textbased-output`) separated by exon placeholders (“-”). By setting the option `--no-separate-introns-in-textbased-output`, delimiters between consecutive introns are omitted (Figure 4).

When plotting intron positions onto aligned protein sequences, introns of different sequences might be at same position except for gaps present in one of the sequences. By default, such introns are aligned. Set `--no-best-position-intron` to turn this behaviour off (Figure 5).

Moreover, intron phases can be included as additional lines in the given alignment (option `--alignment`). Common gaps are removed by default. To keep them, please specify `--keep-common-gaps` (Figure 6A and B). If two or three distinct introns are at the same amino acid alignment position, they will be displayed as “?” instead of listing their intron phase (because of space limitations; figure not shown). Option `--phylo` generates an alignment based on the presence (1) and absence (0) of introns for further phylogenetic analyses (Figure 6C).



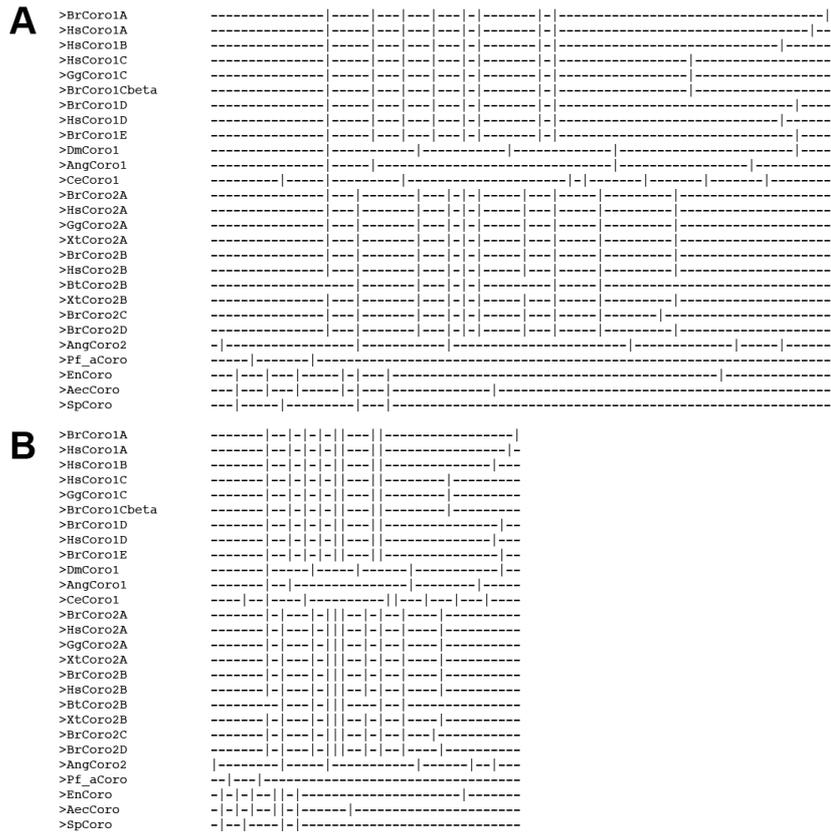


Figure 4 - Effect of `--no-separate-introns-in-textbased-output` option.

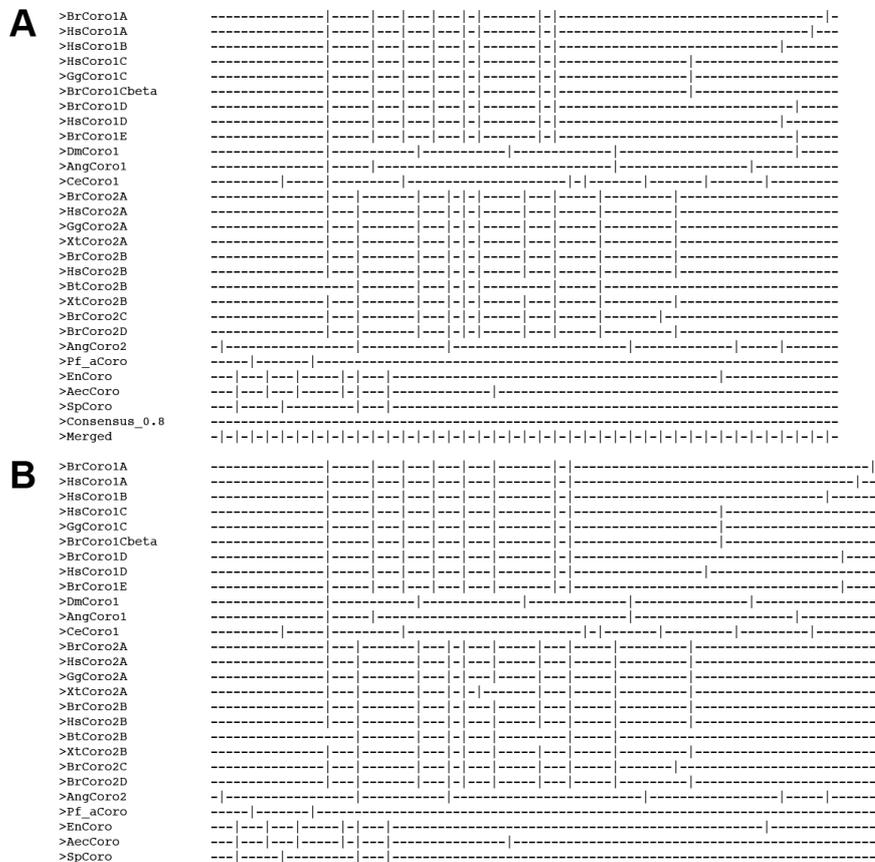


Figure 5 - Effect of `--no-best-position-introns` option.



within format reduced (Figure 7). Both formats can be requested at the same time by specifying both.

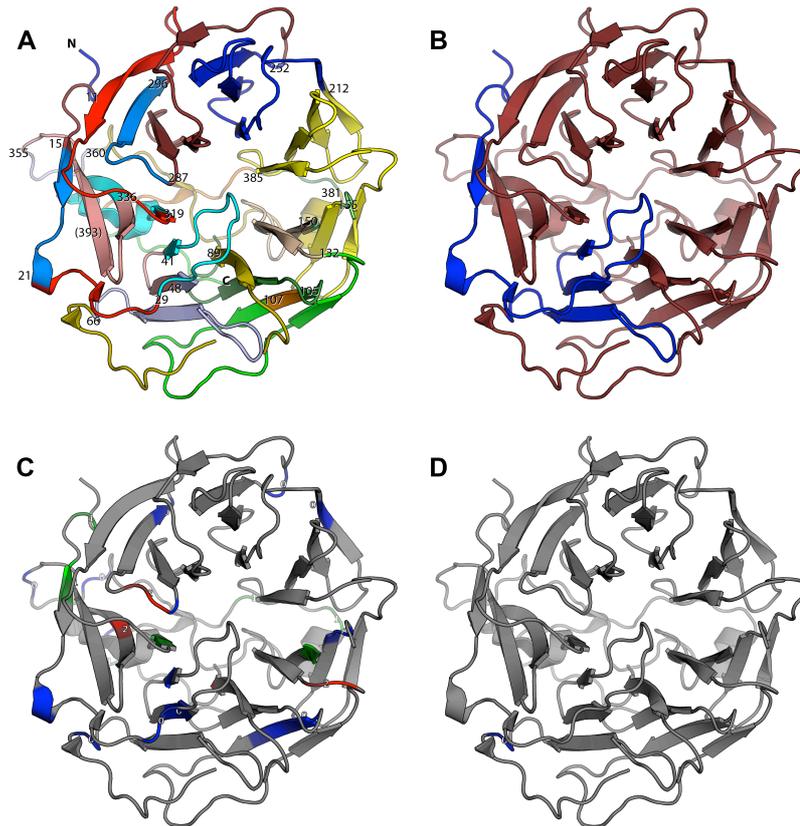


**Figure 7** - Graphical output options.

## Gene structures mapped to protein structures

Additionally, if a pdb file is specified via `--pdb example/2Aq5.pdb`, intron positions and phases are mapped onto protein structure. Figure 8A demonstrates mapping of the exons of the human coronin HsCoro1A gene (`--pdb-ref-prot HsCoro1A`) onto the protein structure of mouse coronin MmCoro1A gene (the pdb file is part of the test data set). While for this figure all exons that are conserved in at least 80% of all proteins are considered (default), Figure 8B displays all exons present in the reference sequence (`--pdb-ref-prot-struct`). Accordingly, splice sites are shown in Figure 8C and Figure 8D. In this output, attention is drawn to intron phases. A three-color scheme and numbers denote phases.

Part of the underlying algorithm is the calculation of a global alignment between reference and pdb sequence. The implementation of the Needleman-Wunsch algorithm was adapted from Michael Ryan, Copyright (c) 2011 (part of the ruby gem `align`, downloaded from <https://rubygems.org/gems/align> at 01-15-2014)

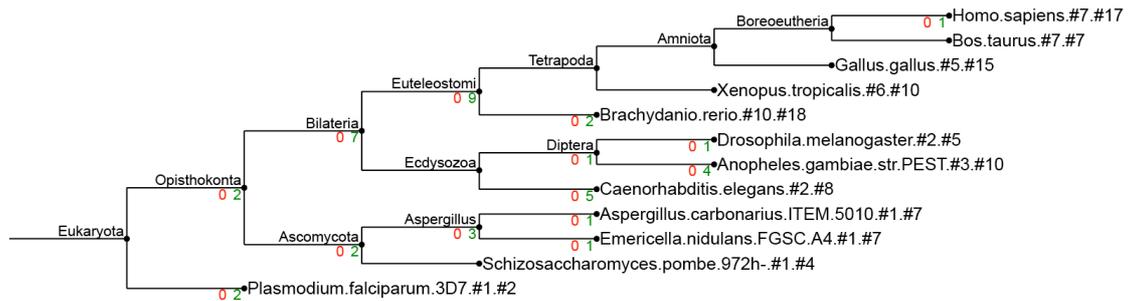


**Figure 8** - Mapping of conserved exons (9A, 9B) and intron position and phase (9C, 9D) onto protein structure.

### Intron gain and loss over taxonomy

Specifying parameter `--tree` in combination with taxonomy options `--taxonomy` and `--taxonomy-to-fasta` can generate a phylogenetic tree of the species analysed. Taxonomic relations of the species named in `--taxonomy-to-fasta` as defined by NCBI taxonomy are converted to a phylogenetic tree in newick file format and printed in an additional SVG file. The python script `"phb2svg.py"` does the conversion of newick tree to SVG. To use `--tree` option, the python interpreter needs to be in your executable path.

The generated phylogenetic tree focuses on gain and loss of intron positions. The generated tree file can be viewed using standard tree viewer software. Branch labels contain the branch name and the number of intron positions that were gained (first number) and lost (second number) at this node. Leafs are annotated with the number of intron positions occurring in that taxon (first number) and the number of genes analysed (second number). If introns were gained or lost in a leaf, these numbers are included in the leaf name (separated by an underscore). In addition, an SVG representation of the tree is generated (Figure 9). In the SVG file, branches are annotated with taxon name (written above the branch) and number of intron losses and (coloured in red and green).



**Figure 9** - Phylogenetic tree of those branches, at which introns were lost (coloured red) or gained (coloured in green). The numbers displayed behind the species comprise the total number of genes (first number) and intron positions present in these genes (second number)

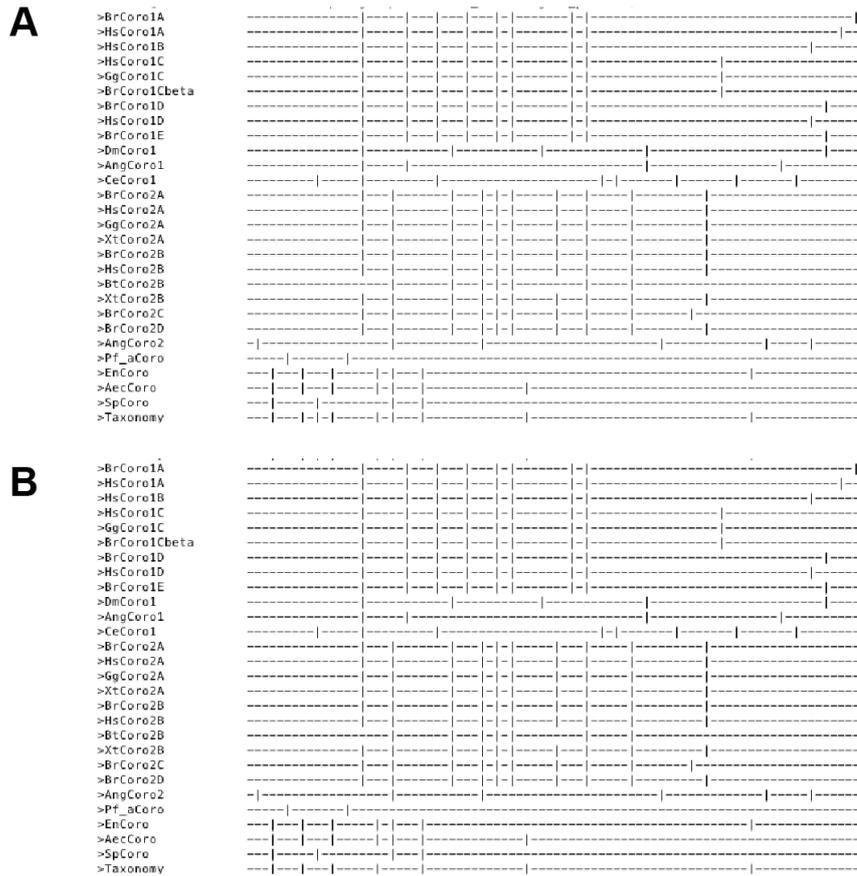
## Taxonomy and statistics

To project intron positions onto the species they occur in, the path to NCBI taxonomy database dump as well as the mapping between fasta headers and NCBI species names must be specified with parameters `--taxonomy` and `--taxonomy-to-fasta`. The file containing the mapping of fasta headers and species names must be in a specific format: Exactly one pair of fasta header and species name per line, which is separated by “:”. Blanks immediately before and after “,” are permitted, they are ignored by the program. More than one fasta header belonging to the same species might be given in one line, the list of fasta headers needs to be comma-separated (Figure 10).

```
RgCoro,RgCoro3:Rhodotorula graminis WP1
LaCoro1A,LaCoro1B,LaCoro1D,LaCoro1C,LaCoro2B,LaCoro2A:Loxodonta africana
MvCoro,MvCoro3:Microbotryum violaceum
PplCoro_A,PplCoro_B:Postia placenta
LabCoro_A,LabCoro_B:Laccaria bicolor
MrgCoro:Meyerozyma guilliermondii ATCC 6260
DehCoro:Debaryomyces hansenii CBS767
LoeCoro:Lodderomyces elongisporus NRRL YB-4239
Ct_aCoro:Candida tropicalis MYA-3404
ClLCoro:Clavispora lusitaniae ATCC 42720
TiaCoro2B,TiaCoro2A:Trichoplax adhaerens
```

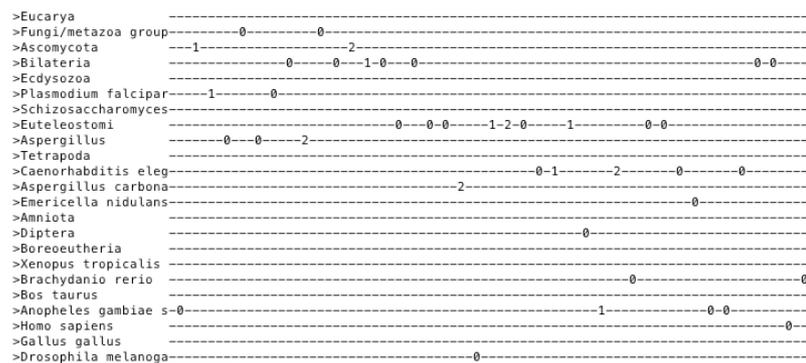
**Figure 10** - Excerpt of file providing mapping between gene names (fasta header) and NCBI species names.

In addition, one of the taxonomic output formats needs to be specified: Parameter `--taxonomy-common-to <x,y,z>` adds a merged exon-intron pattern to standard output containing all introns occurring in any of the species belonging to the specified taxa (Figure 10A). By default, introns need not be exclusively for the specified taxa. To change this behaviour and see only those introns occurring only in the specified taxa, add parameter `--exclusively-in-taxa` (Figure 10B).



**Figure 10** – A) Introns, which are common to fungi are included in the bottom-most exon-intron pattern. B) Only those introns that are exclusive for fungi are included in the bottom-most exon-intron pattern.

In order to assign each intron the last common ancestor of all species it occurs in, parameter `--introns-per-taxon` must be specified. The output is an additional file (extension `-taxonomy`; Figure 11).



**Figure 11** - Each intron is assigned to the last common ancestor of those species it occurs in. The introns are then combined to exon-intron patterns for each taxon.

Adding `--statistics` generates statistics about each intron position. These statistics include the total number of introns at each position as well as the last common ancestor of all species harbouring the respective intron. In addition, the

distribution of intron counts onto direct descendants of that last common ancestor is given. For a better overview, only intron positions occurring in human coronin genes were included in the figure (Figure 12).

```

>HsCoro1A      -|  --  -|  -|  --  -|  -  -|  -|  --  -|  -|  --  --  --  --  -|  -
>HsCoro1B      -|  --  -|  -|  --  -|  -  -|  -|  --  -|  -|  --  --  --  --  -|  -
>HsCoro1C      -|  --  -|  -|  --  -|  -  -|  -|  --  -|  -|  --  --  --  --  -|  -
>HsCoro1D      -|  --  -|  -|  --  -|  -  -|  -|  --  -|  -|  --  --  --  --  -|  -
>HsCoro2A      -|  -|  --  --  -|  --  |  -|  -|  -|  --  -|  -|  -|  --  --  --  -
>HsCoro2B      -|  -|  --  --  -|  --  |  -|  -|  -|  --  -|  -|  -|  --  --  --  -
>Intron number 1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17
Intron number # introns in all data last common ancestor of corresponding genes first unique ancestor of corresponding genes
1  21 Bilateria Deuterostomia (18), Protostomia (3)
2  14 Opisthokonta Fungi (3), Metazoa (11)
3  10 Bilateria Deuterostomia (9), Protostomia (1)
4  10 Bilateria Deuterostomia (9), Protostomia (1)
5  11 Bilateria Deuterostomia (10), Protostomia (1)
6  10 Euteleostomi Actinopterygii (4), Sarcopterygii (5)
7  11 Bilateria Deuterostomia (10), Protostomia (1)
8  20 Euteleostomi Actinopterygii (8), Sarcopterygii (11)
9  19 Euteleostomi Actinopterygii (8), Sarcopterygii (11)
10 9 Euteleostomi Actinopterygii (4), Sarcopterygii (5)
11 9 Euteleostomi Actinopterygii (4), Sarcopterygii (5)
12 19 Euteleostomi Actinopterygii (8), Sarcopterygii (11)
13 10 Euteleostomi Actinopterygii (4), Sarcopterygii (6)
14 8 Euteleostomi Actinopterygii (3), Sarcopterygii (5)
15 3 Euteleostomi Actinopterygii (1), Sarcopterygii (2)
16 3 Bilateria Deuterostomia (2), Protostomia (1)
17 1 Homo sapiens

```

**Figure 12** - Statistics per intron position, including the last common ancestor of all species encoding that intron

## Data and output selection

GenePainter includes all genes in its analysis, for which both an aligned protein sequence and the gene structure are present. Assuming that your multiple sequence alignment contains all sequences of your dataset, the easiest way to analyse only a subset of all data is to copy the corresponding gene structure files to a new folder and specify that folder. Another way to accomplish this task is to use the various data and output selection options GenePainter offers. This option allows you to specify the subset of data to use or analyse in various ways. First, you can provide a list of gene names (corresponding to gene structure file names and fasta headers in the multiple sequence alignment; option `--selection-based-on-list`). Second, you can specify one species name by using parameter `--selection-based-on-species`. You must combine this parameter with `--taxonomy-to-fasta` to associate genes with that species. Those genes associated with the given species are then used. Third, your selection can be based on a regular expression, which is then applied on gene names (parameter `--selection-based-on-regex`). Please enclose the regular expression by quotation marks and specify only the regular expression itself, not the surrounding ("`/`"). No modifiers (such as "`i`" for case insensitivity) can be specified. You can easily check your regular expression on online platforms such as <http://rubular.com/>.

The selection can be applied as data or as output selection by specifying `--output-selection` or `--data-selection`. These two options cannot be combined. For output selection, the complete dataset will be analysed, but only gene structures of selected data are included in output. If you combine the analysis with `--consensus` parameter to determine intron positions which are conserved in a given number of all genes, intron positions occurring in any gene, not only in those used for output generation, will be considered and will occur in consensus exon-intron pattern. Data selection on

the other hand means, that only those intron positions present in the selected data will be included in the output. This includes, that only those genes belonging to the selection will be used for output. In combination with `--consensus` option, this means that intron positions must occur in the selected data while their conservation is determined on all data.