GeneProteinViz (GPViz)

Version 1.2.7

User manual

Rene Snajder

17.04.2013

Table of Contents

1. Introduction

GPViz is a versatile Java-based software for dynamic gene-centered visualization of genomic regions and/or variants. User defined data can be loaded in common formats as resulting from analysis workflows used in sequencing applications and studied in the context of the gene, the corresponding transcript isoforms, proteins and their domains or other protein features. Both the genomic regions and variants can be also defined interactively. Various gene filter options are provided to enable an intersection of variants, genomic regions, and affected protein features. Display options and link outs allow the user to adapt the visualization to individual needs and applications. Finally, it is possible to save publication ready high resolution images in various formats for each individual selected gene or for a batch of genes. GPViz is freely available at http://icbi.at/gpviz (released under GNU general public license), is based on Java 7, and can be used as standalone or webstart application.

1.1. Purpose of this document

This user manual aims to explain how to use GPViz, what data to use, and how to interpret the output. If you're looking for a simple step by step overview of the main functions, check out the **Quick tour** document. You can download it from: http://icbi.at/software/gpviz/gpviz.shtml

1.2. System requirements

GPViz runs on any system that supports the Java 7 runtime. If you do not have Java 7 installed, please get it from http://www.java.com.

Mac users please visit http://www.java.com/en/download/faq/java_mac.xml to learn about how to install Java 7 on OSX.

2. Installation

2.1. Downloading GPViz

To download GPViz, please visit http://icbi.at/software/gpviz and click the “Download” tab. There, click “download” next to the newest version of the client package.

This should download a **ZIP** file, containing all the files you need to run GPViz. After downloading, extract the contents of the ZIP file wherever you want.

You can also use the **Web start** version, which will download a JNLP file that you then have to run with Java Webstart (Java version 7 or higher). This does not contain the demo data set, though, so you'll have to download the ZIP file if you want that.

2.2. Demo Data

We provide a demo data set that will help you to run the “quick tour” section of this manual. This demo data is included in the **ZIP** file package of GPViz in the folde “sample_data”. At the time, this demo data contains data from human genome hg19 chromosome 22 only (to keep it small). It's not suited for regular analyses, since it lacks all other chromosomes.

It contains the following files:

- **Chr22_Ensembl_GRCh37.gtf:** The Ensembl GTF file for Chr22
- **Chr22_Ensembl_IPR.txt:** Protein domains with Ensembl IDs
- **Chr22_Ensembl.txt:** The Ensembl annotation file
- **Chr22_Refseq_hg19.gtf:** The Refseq GTF file for Chr22
- **Chr22_Refseq_CDD.txt:** Protein domains with Refseq IDs
- **Chr22_Refseq_Uniprot.txt:** The Refseq annotation file
- **Chr22_Exons.bed:** A few sample regions on Chr22
- **Chr22_somatic_mutations.maf:** A few sample variants on Chr22

2.3. Running GPViz

On Windows

Unpack the ZIP file you downloaded and double-click “GeneProteinViz.exe”. That should be it. If Java Runtime 7 (or newer) can't be found, there will be a message telling you to install it (see System requirements).

On Mac/Linux

All you need to do is run the JAR file. If Java Runtime 7 is installed (and your system is configured to run JAR files properly, all you might need to do is double-click “GeneProteinViz.jar”.

If that doesn't work, make sure Java 7 is installed. Then, open a shell/terminal and navigate to the folder you extracted GPViz to and run:

JAVA_HOME/bin/java -jar GPViz.jar
(replace “JAVA_HOME” with your Java installation directory)

Note: The first time you start GPViz it will inform you that you haven't defined any annotation file. If you want to work purely with Ensembl data, or have any other data source with consistent IDs, you won't need that. See the “Annotation Files” section for more information.

2.4. Adjusting memory settings

If you plan to load large input files, you might need to adjust the memory settings to allow for more RAM.

On Windows

Open the file GeneProteinViz.l4j.ini in a text editor and change the line **“-Xmx1024m”** to a higher value. For

example **“-Xmx1536m”** for 1.5GB of memory.

On Mac/Linux

Simply add the memory parameter (for example **“-Xmx1536m”**) to the java command when launching the program. For example:

JAVA_HOME/bin/java -jar GPViz.jar -Xmx1536m

Note: If you run a **32bit version** of Java, the amount of memory you can use will probably be limited to around **1,5GB**. If you need more, you'll have to install a 64bit version of Java on a 64bit operating system.

If you're using the web start application you are limited 1024 MB of memory.

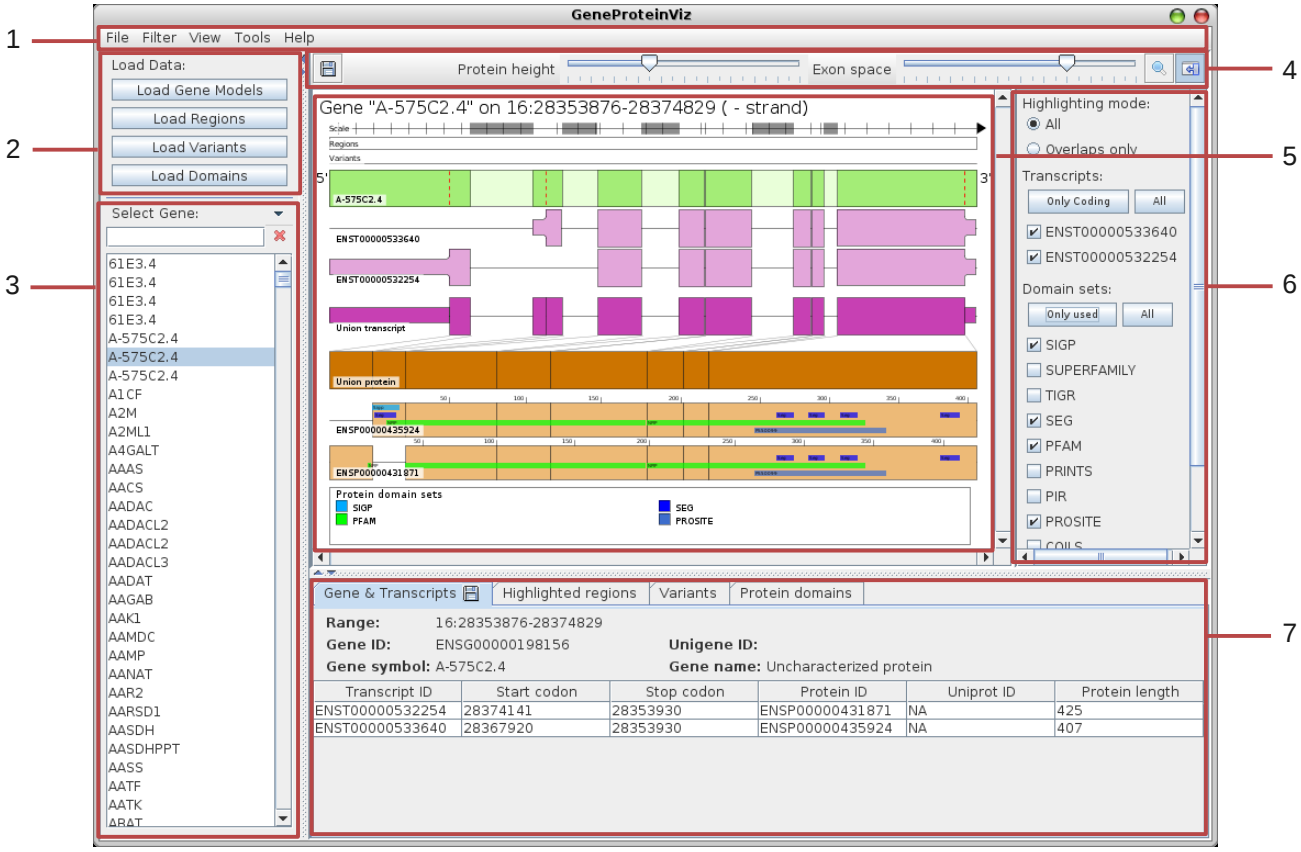
3. Glossary

There are certain terms that may be specific to GPViz or need some clarification.

Term	Description
(Highlighted) region	When talking about “Regions” in GPViz, we usually refer to any type of genomic region that has been selected for highlighting, whether those regions have been loaded from a BED file or selected by hand. Highlighted regions are displayed in the “Regions” bar above the gene representation. A typical use would be to load differentially expressed regions from a file, and then display them in GPViz and see whether they overlap with a protein domain.
Domain	In here a “domain” refers to a protein domain that has been loaded from a file. In GPViz, protein domains are displayed on the proteins, color coded based on their “domain sets”.
Domain set	When proteins are loaded from a file they will be categorized in different sets. The different sets are then color coded and can enabled/disabled individually. How those sets are defined depends on the input data. For CDD (conserved domain database) exports, for example, the column “Hit type” is used to distinguish different sets. For IPR (InterProScan) files, it's the “Type” column. If you build your own protein domain files, check the data format section for more information.

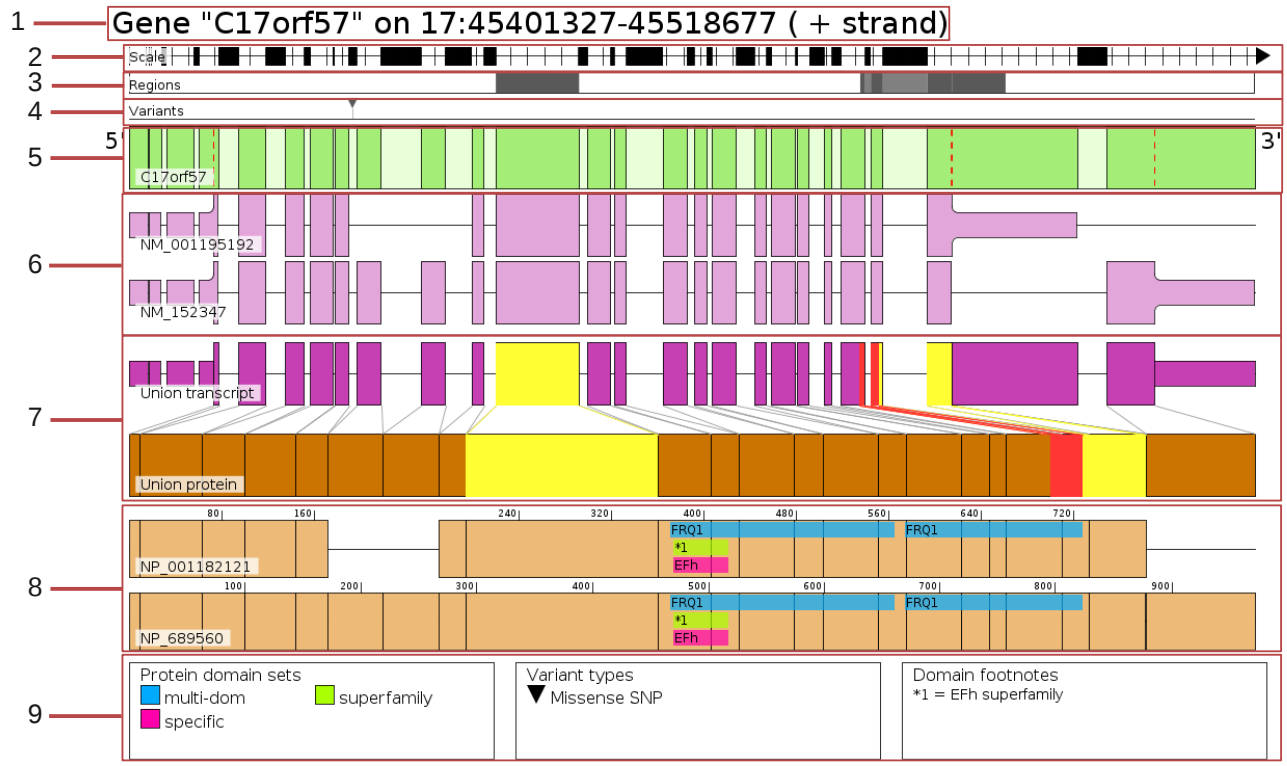
4. The user interface

This is what the user interface looks like with some data loaded:



Item	Description
1) Menu bar	The program's main menu bar.
2) Load data area	From here all data can be loaded.
3) Gene list and search function	After loading a GTF file, this list will be propagated with all genes that have been encountered in the file. To view a specific gene you have to select it in this list, and the figure, as well as the tables and display properties, will be updated accordingly. You can use the search box above it to search for specific genes. If you click the little down arrow next to it, you can also sort the genes and change the display value from gene symbol to gene id or name.
4) Tool bar	In the tool bar you can set some basic display options for the figure. You can configure the height of the protein blocks or the amount of space exons take over. You can also zoom in and hide/show the side bar.
5) Canvas	Here is where the figure is painted
6) Sidebar	Contains the transcripts, protein domain sets and samples (regions&variants) loaded, and lets you enable/disable them separately.
7) Detail area	Shows details and tables about the gene that has currently been selected, the regions and variants that apply to this gene, and the protein domains associated with it. Using the small disk icon in the tab header you can save every table to a tab delimited file.

4.1. The figure



Item	Description
1) Title	Shows the gene's display **name** (usually the gene symbol), which **strand** and which **chromosome** it is on, and at which nucleotide **position** the first exon begins and the last exon ends.
2) Scale	This scale shows which areas of the genome have been compressed/expanded. GPViz aims to deliver a schematic view on exons more than intros, so by default introns are compressed (this behavior can be controlled by the “Exon space” slider above the image). The distance between two lines on the scale represents **100 nucleotides**.
3) Region highlighters	Here GPViz displays which **regions** have been highlighted, for example differentially expressed regions.
4) Variants	Here variants are displayed. Each type of **variant** is shown with different symbol. The symbols are explained in the “Variant types” legend at the bottom of the figure.
5) Gene	This is a schematic view at the exons and introns of this gene. The dark green areas represent exons and the light green areas are introns. The red dashed line shows where an exon is “cut” in transcription or coding, due to different transcripts and start/stop codons.
6) Transcripts	Here we see which transcripts are known from this gene.
7) Union transcript and union protein	The **union transcript** is an imaginary transcript that consists of all the exons and exon segments from all the transcripts. This is used to better visualize which segments end up creating which parts of the protein. The **union protein** works the same way, as it is an imaginary protein that contains all the protein segments from all the various proteins we know about. Think of it as “what if the union transcript was coded into a protein”. It is spliced into different segments to show which exons end up causing which parts of the protein. The union transcript and union protein also contain the highlighters for the selected **regions** and **variants**. You can see them in yellow or red. **Yellow** means a region/variant

	that applies to at least one protein, where a **red** highlighter shows where in addition the region/variant also overlaps with a protein domain. This can help to visualize for example which differentially expressed region or variant is more likely to cause a functional change in a protein.
8) Proteins	Here you see all the **proteins** that can be created by the transcripts above. Since there are sometimes non-coding transcripts too, there might be fewer proteins than transcripts. The proteins are **aligned** to the union protein, so you can easily trace the highlighters on the union protein. Each protein also shows the **protein domains**. Each domain set (or category) is represented by a different color and painted at a different height, in an attempt to avoid overlaps. When the domain is too short to display the full name in the bar, a footnote (*1, *2, *3, ...) is placed which is then explained in the **footnotes** legend.
9) Legends	There are three types of legends. **Protein domain sets:** This legend shows which color represents which protein domain set. **Variant types:** Explains the symbols for the different variant types present in this figure. **Footnotes:** When a protein domain name is too long to be printed in the figure, a footnote is placed instead. These footnotes are then referenced here. **Note** that every legend is only displayed **when required**. If there are no protein domains, there will be no protein domain sets legend or footnotes. If there are no variants in the selected gene, there will be no variant types legend.

5. Data formats

In GPViz we tried to give the user as much freedom as possible, allowing him to provide each and every bit of data by themselves. At the same time, we tried to use and support established file formats where possible.

5.1. Exons

So far only the **GTF** file format is supported for loading exons.

5.1.1. Loading Exons from a GTF File:

See: http://mblab.wustl.edu/GTF2.html.The following fields are required for GPViz to work:

- **Seqname:** Read as “chromosome” in GPViz.
- **Feature:** Only the lines with feature “start_codon”, “stop_codon” and “exon” are read. All other lines are ignored.
- **Start & End:** Start and stop position of the exon/codon.
- **Attributes:**
  - **gene_id**: Should be a **unique** ID of the gene this exon/codon is on
  - **gene_name:** This is interpreted as the **gene symbol** in GPViz. Does **not** have to be unque.
  - **transcript_id:** Also needs to be **unique**. It's the ID of each transcript.
  - protein_id (optional): Protein IDs are optional, but you need them if you want to map protein domains later. Check out the **annotation files** section for how to add protein IDs if they are not in your GTF file.

5.2. Protein domains

Since there isn't really any clearly defined format for protein domains, we tried to support a wide variety of formats to make sure it works with any type of data.

5.2.1. CDD export:

See http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd.shtml. The CDD (conserved domain database) is a great source to get protein domains with RefSeq IDs.

Format:
There has to be a header of exactly 8 lines, and it will look something like this:

```
#Batch CD-search tool   NIH/NLM/NCBI
#cdsid  QM3-qcdsearch-18A9185BF062555A-14CE4649E2AFAE71
#datatype       hits Concise data
#status 0
#Start time     2012-07-20T07:46:22     Run time        0:00:08:05
#status success

Query  Hit type  PSSM-ID  From  To  E-Value  Bitscore  Accession  Short name  Incomplete
Superfamily
```

The header lines written in bold are the columns that are actually mandatory. Regardless, the other columns have to exist, but can be empty. The **Hit type** column is then used to separate domains into domain sets.

After that the columns are **tab delimited** and look something like this:

```
Q#1 - NP_000005  multi-dom    203733  131  268  0.0041439  37.9102  pfam07703 A2M_N_2     -  -
Q#2 - NP_000006  superfamily  186276  20   280  3.031e-99  289.57   cl00949 superfamily   -  -
Q#3 - NP_000007  specific     173846  41   418  0          779.074  cd01157 MCAD          -
cl09933
Q#3 – NP_000007  superfamily  209100  41   418  0          779.074  cl09933 superfamily   -
```

Note that the first column “Q#1 – NP_000005” is only one field. It needs to be in this format, as this is the way CDD outputs it's query results.

5.2.2. InterProScan format:

See: http://www.ebi.ac.uk/Tools/webservices/services/pfa/iprscan_soap. Ebi provides a multitude of tools to extract protein domains from their database. This is very useful if you are working with an Ensembl GTF file and thus need protein domains that reference Ensembl protein IDs.

We've used their **Perl** interface in particular to create an export that looks like this:

```
# ensembl homo sapiens core 70 37
# protein domain, motifs, and feature mapping by InterProScan
(http://www.ebi.ac.uk/Tools/pfa/iprscan/
# time 21 Jan 2013

Protein ID      Type  ID      IPR ID  Short Name      Long Name    From   To  E-value  Score_1
Score_2
ENSP00000447024 TM    Tmhmm   -       Tmhmm           Tmhmm        78     95  0        0        0
ENSP00000381589 TM    Tmhmm   -       Tmhmm           Tmhmm        13     27  0        0        0
```

Again there is a fixed header of exactly 5 lines that needs to be present and columns written in bold are required.

5.2.3. PFAM format:

PFAM (http://pfam.sanger.ac.uk/) is another database that lets you create exports. They use **uniprot IDs**, though, so you have to make sure your data is annotated with uniprot IDs in order to use this file format. Please check the **annotation file** section for more information.

The format contains 3 mandatory header lines and looks like this:

```
#Pfam-A regions from Pfam version 26.0 for ncbi taxid 9606 'Homo sapiens'
#Total number of proteins in proteome: 59052
#<seq id> <alignment start> <alignment end> <envelope start> <envelope end> <hmm acc> <hmm name>
<type> <hmm start> <hmm end> <hmm length> <bit score> <E-value> <clan>
C3U398   304   358   304   358   PF00046 Homeobox   Domain   1   55   57   62.60   2e-14    CL0123
C3U398   229   283   229   285   PF00046 Homeobox   Domain   1   55   57   61.30   5.2e-14  CL0123
```

The column called “seq id” is actually the uniprot ID and the column called “type” is what's used to create domain sets. Columns written in bold are required by GPViz.

5.2.4. Simple tab delimited format:

Since the other two formats aren't really standard formats and contain a lot of columns that aren't really required, we also support a custom file format for protein domains. Since it's a custom format you can use whatever IDs you like, and it doesn't matter whether your data uses RefSeq or Ensembl IDs. The file has 1 single header line and the rest is tab delimited. It looks like this:

```
# Protein ID     Type     From     To       ID       Short Name     Long Name     Comment
ENSP00000447024 TM       78       95       Tmhmm    Tmhmm          Tmhmm         Some comment
ENSP00000381589 TM       13       27       Tmhmm    Tmhmm          Tmhmm         -
ENSP00000381589 TM       693      715      Tmhmm    Tmhmm          Tmhmm         -
```

The columns Long Name and Comment are optional, but they will be read and displayed in the protein domains table in the details area.

5.3. Regions

The **BED format** is a very common format for defining something as simple as genomic regions. You can read up on it here: http://genome.ucsc.edu/FAQ/FAQformat.html#format1

Any BED file that is compatible with the standard requirements of the BED format will work with GPViz. The columns that are actually used by GPViz are: **chrom, chrom_start, chrom_end, strand**. All other columns will simply be ignored.

5.4. Variants

The **MAF** format is currently best supported for variant files, as it is very clearly defined and has a finite number of variant types and variant classifications, which makes it easy to create a consistent visualization. Nevertheless GPViz also supports **VCF**, **SNP**, and **MUT** files as input.

5.4.1. MAF Format

The MAF format is very well defined, and you can read all about it here:
https://wiki.nci.nih.gov/display/TCGA/Mutation+Annotation+Format+(MAF)+Specification

The format is pretty extensive, but GPViz uses only the columns **Start_Position**, **End_Position**, **Strand**, **Variant_Classification**, **Variant_Type**, and **dbSNP_RS**. Example (with columns after the last mandatory column removed to make it shorter):

```
Hugo_symbol   Entrez_Gene_ID   Center   NCBI_Build   Chrom   Start_Position   End_Position   Strand
Variant_Classification   Variant_Type   Reference_Allele   Tumor_Seq_Allele1   Tumor_Seq_Allele2
DBSNP_RS   ...
```

```
PRKCZ   5590   hgsc.bcm.edu   36   1977815   1977815   +   Missense_Mutation   SNP   C   C   A
novel   ...
```

5.4.2. VCF Format

GPViz also supports Variants in the VCF format version 4.0. You can found more about it here:

http://www.1000genomes.org/node/101

To determine the type of Variant, GPViz reads the SVTYPE value in the VCF entry. If the SVTYPE value is not present, GPViz tries to distinguish SNPs, Insertions, and Deletions based on the Ref and Alt bases in the VCF entry. If you want to influence this behavior, please add the SVTYPE value to the INFO column.

```
##fileformat=VCFv4.0
##fileDate=20100501
##reference=1000GenomesPilot-NCBI36
##assembly=ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/release/sv/breakpoint_assemblies.fasta
##INFO=<ID=BKPTID,Number=-1,Type=String,Description="ID of the assembled alternate allele in the
assembly file">
##INFO=<ID=CIEND,Number=2,Type=Integer,Description="Confidence interval around END for imprecise
variants">
##INFO=<ID=CIPOS,Number=2,Type=Integer,Description="Confidence interval around POS for imprecise
variants">
##INFO=<ID=END,Number=1,Type=Integer,Description="End position of the variant described in this
record">
##INFO=<ID=HOMLEN,Number=-1,Type=Integer,Description="Length of base pair identical micro-homology
at event breakpoints">
##INFO=<ID=HOMSEQ,Number=-1,Type=String,Description="Sequence of base pair identical micro-homology
at event breakpoints">
##INFO=<ID=IMPRECISE,Number=0,Type=Flag,Description="Imprecise structural variation">
##INFO=<ID=MEINFO,Number=4,Type=String,Description="Mobile element info of the form
NAME,START,END,POLARITY">
##INFO=<ID=SVLEN,Number=-1,Type=Integer,Description="Difference in length between REF and ALT
alleles">
##INFO=<ID=SVTYPE,Number=1,Type=String,Description="Type of structural variant">
##ALT=<ID=DEL,Description="Deletion">
##ALT=<ID=DEL:ME:ALU,Description="Deletion of ALU element">
##ALT=<ID=DEL:ME:L1,Description="Deletion of L1 element">
##ALT=<ID=DUP,Description="Duplication">
##ALT=<ID=DUP:TANDEM,Description="Tandem Duplication">
##ALT=<ID=INS,Description="Insertion of novel sequence">
##ALT=<ID=INS:ME:ALU,Description="Insertion of ALU element">
##ALT=<ID=INS:ME:L1,Description="Insertion of L1 element">
##ALT=<ID=INV,Description="Inversion">
##ALT=<ID=CNV,Description="Copy number variable region">
##FORMAT=<ID=GT,Number=1,Type=Integer,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Float,Description="Genotype quality">
##FORMAT=<ID=CN,Number=1,Type=Integer,Description="Copy number genotype for imprecise events">
##FORMAT=<ID=CNQ,Number=1,Type=Float,Description="Copy number genotype quality for imprecise
events">
#CHROM  POS    ID  REF          ALT    QUAL    FILTER   INFO    FORMAT   NA00001
1   2827693    .   CCGGC        C     PASS    SVTYPE=DEL    GT:GQ   1/1:13.9
2   321682     .   T   <DEL>    6     PASS    IMPRECISE;SVTYPE=DEL;     GT:GQ   0/1:12
2   14477084   .   C   <DEL:ME:ALU>   12   PASS    IMPRECISE;SVTYPE=DEL   GT:GQ   0/1:12
3   9425916    .   C   <INS:ME:L1>   23   PASS    IMPRECISE;SVTYPE=INS;   GT:GQ   1/1:15
3   12665100   .   A   <DUP>    14   PASS    IMPRECISE;SVTYPE=DUP;   GT:GQ:CN:CNQ  ./.:0:3:16.2
4   18665128   .   T   <DUP:TANDEM>   11   PASS    IMPRECISE;SVTYPE=DUP;   GT:GQ:CN:CNQ   ./.:0:5:8.3
```

5.4.3. SNP Format

The SNP file format is defined here:

http://www.broadinstitute.org/cancer/software/genepattern/gp_guides/file-formats/sections/snp

Since there is no real definition of variant types in this format, all Variants loaded with this file format will simply be displayed as “SNP”.

5.4.4. MUT Format

The MUT format is a very basic tab delimited format, briefly described here:

http://www.broadinstitute.org/igv/MUT

Since the “mutation type” in this format can be any arbitrary text value, it is simply displayed in GPViz directly as it is.

5.5. Annotation files

Annotation files are files that we configure once in the **Options** dialog, and that will then be loaded automatically every time GPViz starts. They serve the purpose of mapping additional information onto genes and proteins, that were missing in the original GTF file.

For example, the GTF files downloaded from the RefSeq database contain RefSeq transcript IDs (NM_...), but no RefSeq protein IDs (NP_....). The CDD export protein domain files, on the other hand, use RefSeq protein IDs. So if we want to use those two together, we first need to find some kind of mapping between RefSeq transcript ID and RefSeq protein ID.

Another application is when we have Ensembl IDs in the GTF file but want to link to a PFAM protein domain file which uses Uniprot IDs.

We therefore defined 2 annotation files. One called **Refseq annotation file** and one **Ensembl annotation file**.

5.5.1. RefSeq annotation file

The RefSeq annotation file uses the **Transcript ID** to map additional information onto the transcript. It's a tab delimited file without header. So if you are working with RefSeq GTF files, you are most likely going to need this annotation file. It contains the following fields:

- RefSeq Transcript ID
- RefSeq Protein ID
- Uniprot ID
- Unigene ID
- Long gene name
- Gene symbol

Example:

```
NM_130786       NP_570602       P04217   1       alpha-1-B glycoprotein    A1BG
NM_000014       NP_000005       P01023   2       alpha-2-macroglobulin     A2M
NM_000662       NP_000653       P18440   9       N-acetyltransferase 1     NAT1
NM_001160170    NP_001153642    P18440   9       N-acetyltransferase 1     NAT1
NM_001160171    NP_001153643    P18440   9       N-acetyltransferase 1     NAT1
NM_001160172    NP_001153644    P18440   9       N-acetyltransferase 1     NAT1
```

5.5.2. Ensembl annotation file

The Ensembl annotation file is mainly used for mapping UniProt IDs on Ensembl protein IDs, thus enabling you to read PFAM protein domains. If you don't need UniProt IDs you probably don't need to provide this file.

Like the RefSeq annotation file, this is a tab delimited file without any header. It contains the following columns:

- Ensembl Transcript ID
- Ensembl Protein ID
- UniProt ID
- Ensembl Gene ID
- Long gene name
- Gene symbol

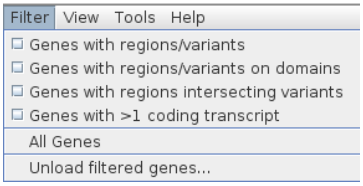
Example:

```
ENST00000381700   ENSP00000371119   Q13068   ENSG00000255738   G antigen 4                  GAGE4
ENST00000553856   ENSP00000452581   NA       ENSG00000258664   interferon alpha             IFRG15

ENST00000600108   ENSP00000471569   A8MPY1   ENSG00000269434   gamma-aminobutyric acid      GABRR3

ENST00000305831   ENSP00000306381   Q9UPU9   ENSG00000262355   Protein Smaug homolog 1      SAMD4A

ENST00000594964   ENSP00000471397   Q9H227   ENSG00000269541   Cytosolic beta-glucosidase   GBA3
```

How to create it: Visit Ensembl BioMart (http://www.ensembl.org/biomart/martview/) and select the proper fields from there.

6. Filters

To filter genes in GPViz, click the **Filter** menu in the main menu bar.

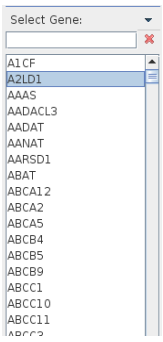


When a filter is a applied, the entire gene list will be searched based on the selected criteria and the list will be reduced to those genes that qualify. You can then go back to uncheck those filters at any time or click “All Genes” to restore the full list.

Here is a short explanation of the functions:

Menu Item	Description
Genes with regions/variants	This filter will pass only those genes that have **regions or variants present.** It's very useful when you load a BED file or MAF file and want to look only at the genes that are actually affected by it.
Genes with regions/variants on domains	Similarly to the previous one, when this filter is activated, only genes that have regions or variants present will be displayed. In addition, it will narrow it down to those genes where the regions or variants are also **overlapping with protein domains**, pointing out where functional changes on the protein might occur.
Genes with regions intersecting variants	If you loaded both regions from a bed file as well as variants, you can use this filter to find those genes where variants are found **within** regions. You can use this if you are interested in finding variants within a specific genomic region.
Genes with >1 coding transcript	Lot's of genes in your GTF file might have only 1 **coding transcript** and thus create only 1 protein, or contain no coding transcript at all and thus create no protein. With this filter you can remove those genes from the list.
All Genes	Removes all filters and restores the original gene list
Unload filtered genes...	This menu **unloads** all genes that are **not accepted** by your currently selected filters. This means that all genes that have been filtered out will be completely removed and even the “All Genes” menu won't get them back. The only way to undo this action is to load the GTF file again. This function is very useful when you are **running low on memory**. For example, you could load your GTF file and not have enough memory to load the protein domains. So instead you load a BED file with regions, use the “Genes with regions/variants” filter, which dramatically reduces your gene list, run “Unload filtered genes...” to clear up memory and then load the protein domains.

7. Gene list



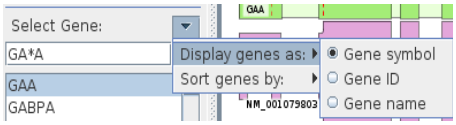
The gene list on the left side displays all the genes that have been loaded from the GTF file and (if applicable) accepted by the selected filters.

At the top you see the **search field**. Here you can type the name of the gene you are looking for, or use asterisks (*) to search for partial matches. For example, searching for “GA*A” would return all genes that start with “GA” and then later have “A” somewhere in the name, like GABPA, GATA3 and GAGE2A.

To **clear** that search simply click the little red **X** next to the search field.

If you click a gene in this list, the figure, tables and display properties will automatically be updated to show this gene. You can also make multiple selections by using the **Shift** and **Ctrl** keys, when you want to **save multiple figures**.

Next to the text “Select Gene:” you can also see a small **down arrow**. This opens up the gene list menu.



This menu allows you to change how the genes are displayed and sorted. Note that when you change the gene sorting, filters might have to be re-applied. This can take a couple of seconds.

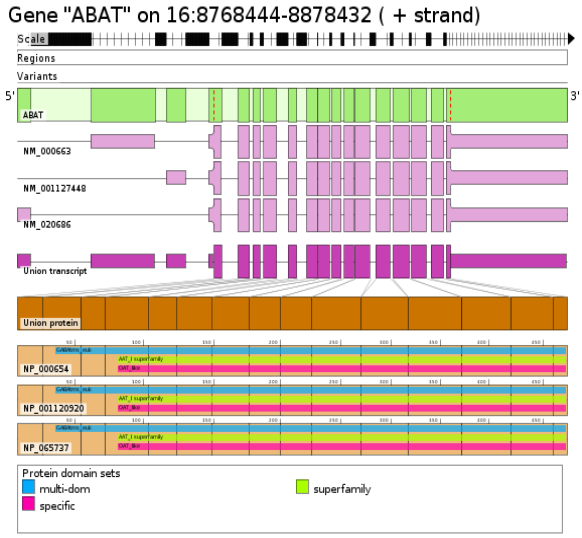
8. Display options

There are several options that give you some control over what your figure looks like. Most of them just hide/show elements, but others need a little more explanation.

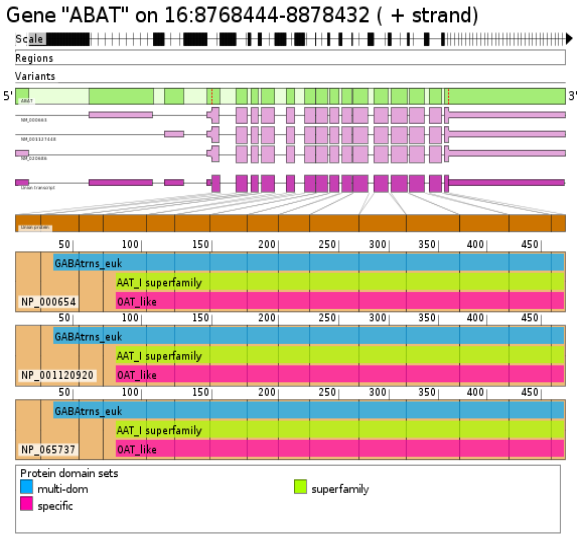
8.1. Protein height & Exon space

In the toolbar above the figure you can see these two sliders. The **protein height** slider adjusts how high the protein blocks are displayed in the figure. This can be very useful when there are so many protein domain sets, or so many proteins, that the protein domain labels become unreadable.

Let's take this picture for example. On the left side, we can hardly read the protein domains. On the right side, though, we made increased the protein height, and the domains are much more readable.



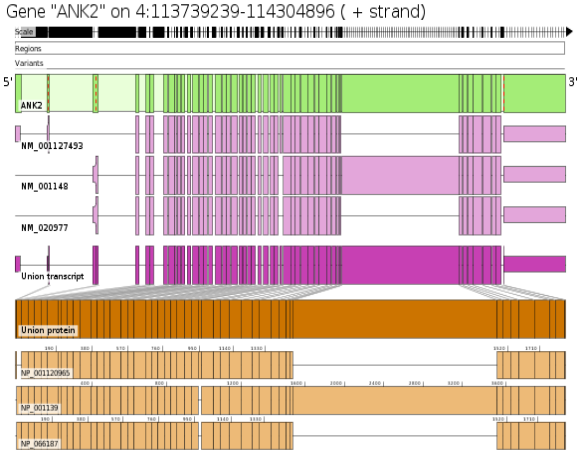
Low protein height

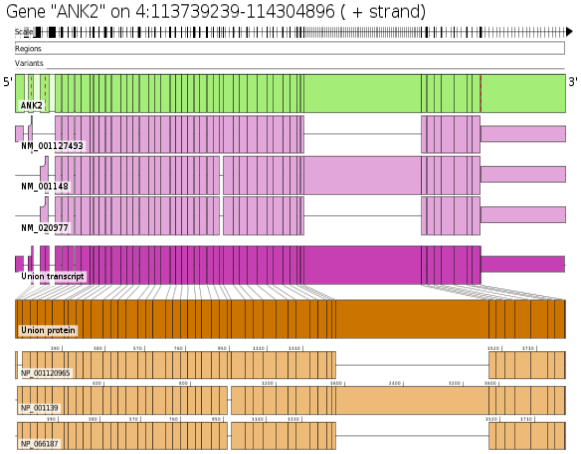


Higher protein height

The **exon space** slider, changes how much of the horizontal space is used for exons relative to introns. Since GPViz focuses on a more schematic view on the exons (rather than keeping everything to scale), exons are always “inflated” to use a certain percent of the screen. This is reflected by the **scale** directly below the figure title.

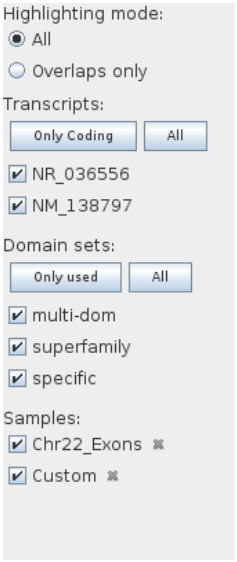
The default exon space setting should work fine for most genes, but sometimes there are genes with lots of small exons, and then it can get really crowded. In that case increasing the exon space might make it easier to distinguish them. Here is an image comparing default exon space (left) and higher exon space (right).





8.2. Sidebar

The sidebar currently contains some of the display settings for the figure.



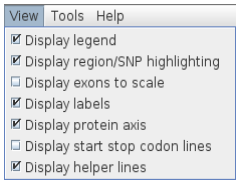
The **Highlighting mode** let's you choose whether to highlight all regions/variants that have been loaded, or only those that overlap with protein domains.

The **Transcripts** list shows all the transcripts present in this gene. You can uncheck transcripts here, which will remove them from the figure. By clicking **Only Coding** all non-coding transcripts will automatically be unchecked. This is useful when you have a large number of transcripts and are only interested in those that actually produce a protein.

The **Domain sets** section lists all domain sets that have been loaded. You can uncheck them to not display them in the figure. This can help you to create more space to make the other domain sets more readable. The **Only used** button will automatically uncheck those protein domain sets that have no domains in the current gene.

The **Samples** list represents all regions and variants files that have been loaded. A sample called **Custom** will be created whenever the user manually adds regions or variants to a gene. Unchecking samples will remove them from being highlighted in the figure.

8.3. View menu



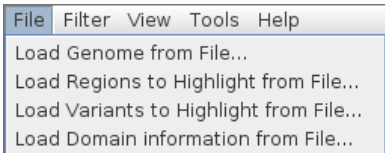
You can either select the **View** menu in the menu bar, or right-click the figure in order to open the View menu. There you will find a few more options to further customize how the figure is drawn.

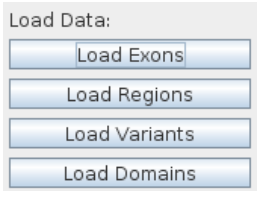
Option	Description
Display legend	When unchecked, the three legends at the bottom of the figure will not be drawn. If checked, the legends will be drawn, but only if they are required.
Display region/SNP highlighting	This enables/disables all region and variant highlighting and even controls whether the “Region” and “Variant” bars should be displayed at the top of the figure.
Display exons to scale	Besides inflating exons to make them more visible in between introns, GPViz also rescales exons so that very large exons shrink a bit in order to make very small exons more visible. While it is programmed so that large exons are still always larger than small exons (small exons can't “outgrow” large exons), this mechanism skews the proportions between exons. When you select “Display exons to scale” this mechanism will be disabled. This means exons will always be at scale to each other (an exon appearing twice as big as another exon is in reality also twice as big).
Display labels	Whether to display labels in the figure, such as the gene name, transcript IDs and protein IDs.
Display protein axis	Whether to show the ticks with the amino acid count above proteins.
Display start stop codon lines	If enabled, the start and stop codon of each transcript is displayed as a line.
Display helper lines	Whether to draw the helper lines between the union transcript and union protein.

9. Loading data

You can load data either through the File menu or using the Load Data section in the main window. There are 4 types of data you can load:

- Exons (GTF files)
- Regions (BED files)
- Variants (MAF, SNP, MUT or VCF files)





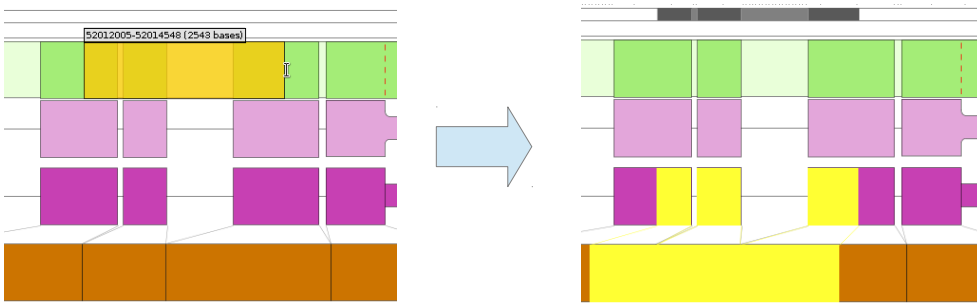
- Protein Domains (CDD export, PFAM files, Ensembl domains, Custom tab delimited format)

For more details on the file formats check the File formats section.

9.1. Manually adding regions and variants

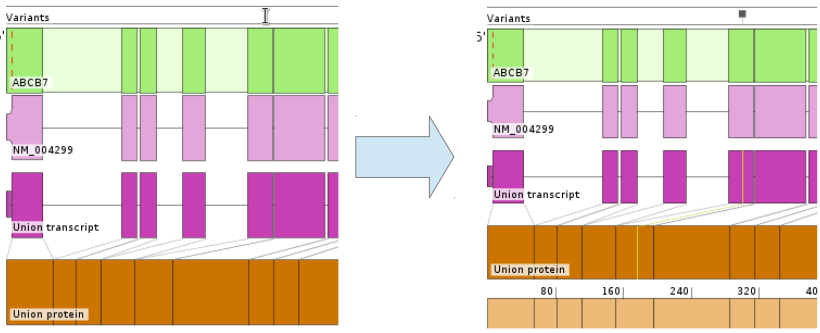
You can also manually add regions and variants using the selection tool in the figure.

To add a **region**, for example, drag your mouse across the gene, transcript, or region bar to mark it, and then double-click the marking to accept it as a highlighting.

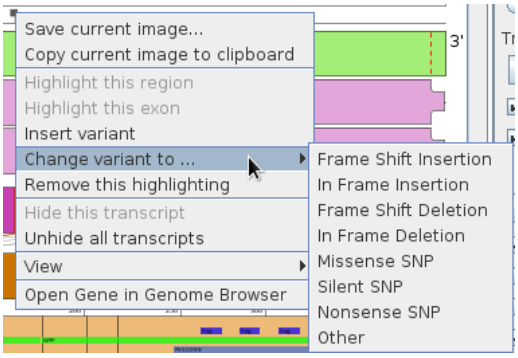


You can also highlight an entire exon simply by double-clicking that exon.

To add a **variant** you can simple double click anywhere in the Variants block, or hold the **Strg** key while double clicking a position in a gene or transcript block.



You can then right click the variant and select “Change variant to ...” to change the variant to one of the predefined types:



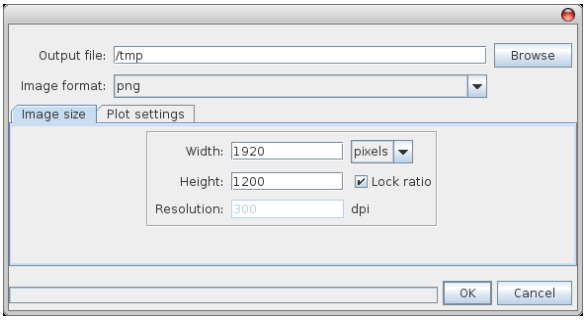
10. Saving images

There are a number of ways how you can save images. You can save them **individually**, you can **batch save** several images into several different files, or you can save multiple images into one **multipage document** (which has to be either PDF or TIFF format).

10.1. Saving individual images

To save an individual image you can either click the **save icon** in the toolbar, you click **File** -> **Save current image**, or you **right-click** the figure and then click **Save current image**.

In any way, it will open this dialog:

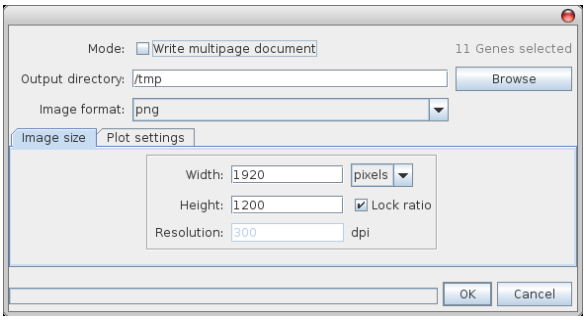


Here you can set the output file, file format, and resolution.

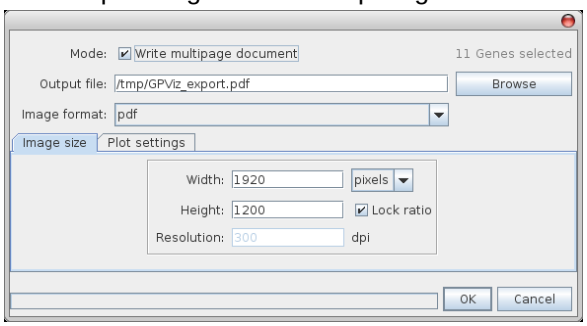
10.2. Saving multiple images

If you want to save figures for **multiple genes** at once, you can either click **File** -> **Save images for filtered genes** to save all genes currently in your gene list (note, if you do this with a lot of genes it will probably take a very long time to save) or you can **select multiple genes** in the gene list, using the **Ctrl** and **Shift** keys, and then click **File** -> **Save images for selected genes**.

You will then get the same dialog, but slightly different:



Notice how now the dialog is asking for an **output directory** rather than an output file. This is because you are now planning to save multiple figures at once.



Also you now have the option to switch to **Write multipage document**. If you select this, the image format list will be reduced to **PDF** and **TIFF**, since those are the only two supported formats that allow for multiple

pages in one file. Also the output directory will change back to output file.

10.3. Setting image resolution

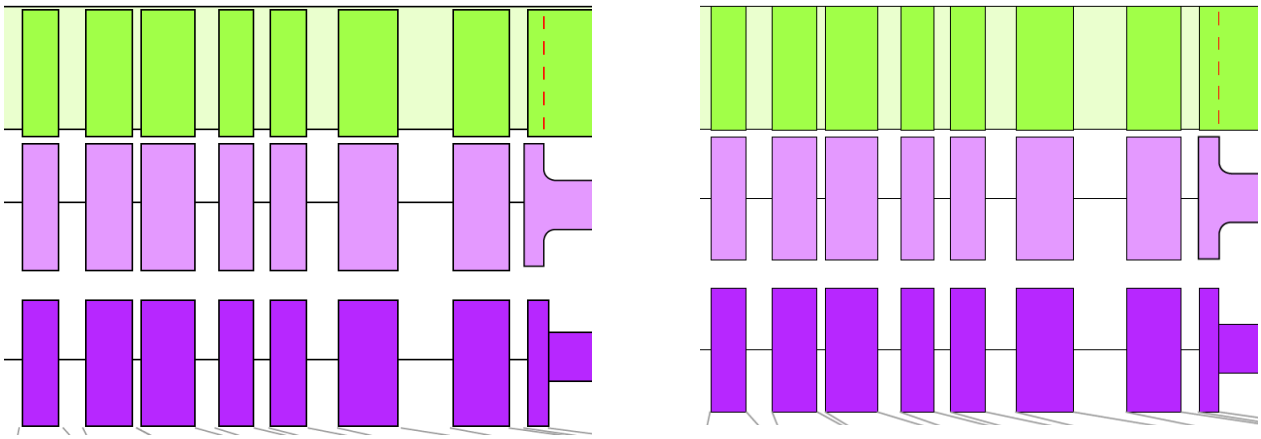
Regardless of whether you save images individually or in multiple files, you'll always have the same options available to set the image resolution.

You can either set the exact pixel size, or you can change to “mm” or “inch” to set the print size. When you change to “mm” or “inch”, you'll see that you can now set the resolution in dpi.

Note that you'll only achieve the desired resolution (for example 300dpi for a publication) if you set an appropriate print size. If the print size is too small, even setting a high dpi value will result in bad resolution, so be sure to estimate the print size generously.

10.3.1. A word about PDFs

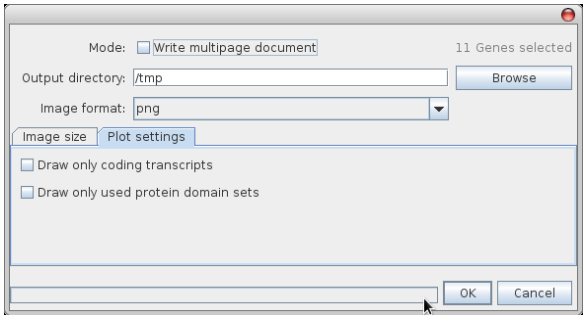
PDFs created by GPViz are drawn entirely as **vector graphics**. So they are a very good way of saving images without any loss of quality. Setting the **resolution** for a vector graphic is theoretically meaningless, as they aren't rendered, **but** in GPViz the set resolution affects the vector graphics' **accuracy**. So, even if you save the figures as PDFs you should still set an appropriate resolution.



Here you can see the **difference** between a **high accuracy** and **low accuracy** PDF.

10.4. Plot settings when saving images

You may have noticed that you can switch to the “Plot settings” tab in the save dialog.

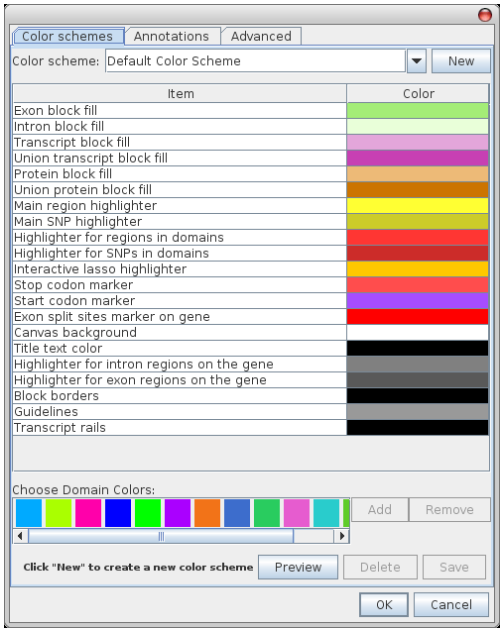


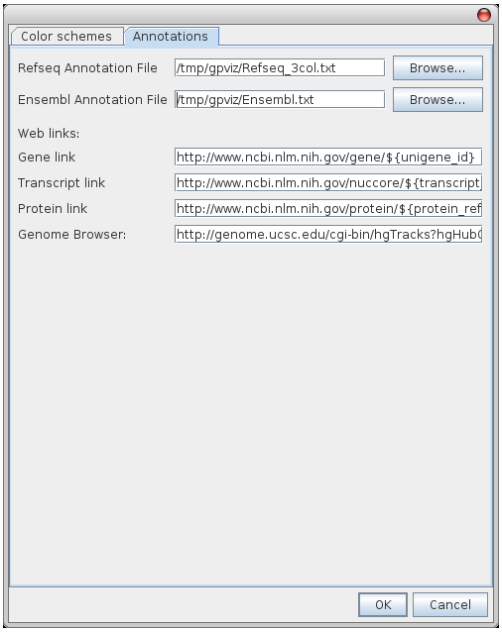
This currently allows you to choose to draw only coding transcripts and/or draw only those protein domain sets that are actually used in a gene. Both options can help you to save space in very crowded figures, and they do exactly the same as the “Only coding” and “Only used” buttons in the sidebar, except that here it is applied to every figure that is drawn.

11. Options menu

You can access the options dialog by clicking **Tools** and then **Options** in the menu bar.

There are currently two tabs in the option dialog: **Color schemes** and **Annotations**.





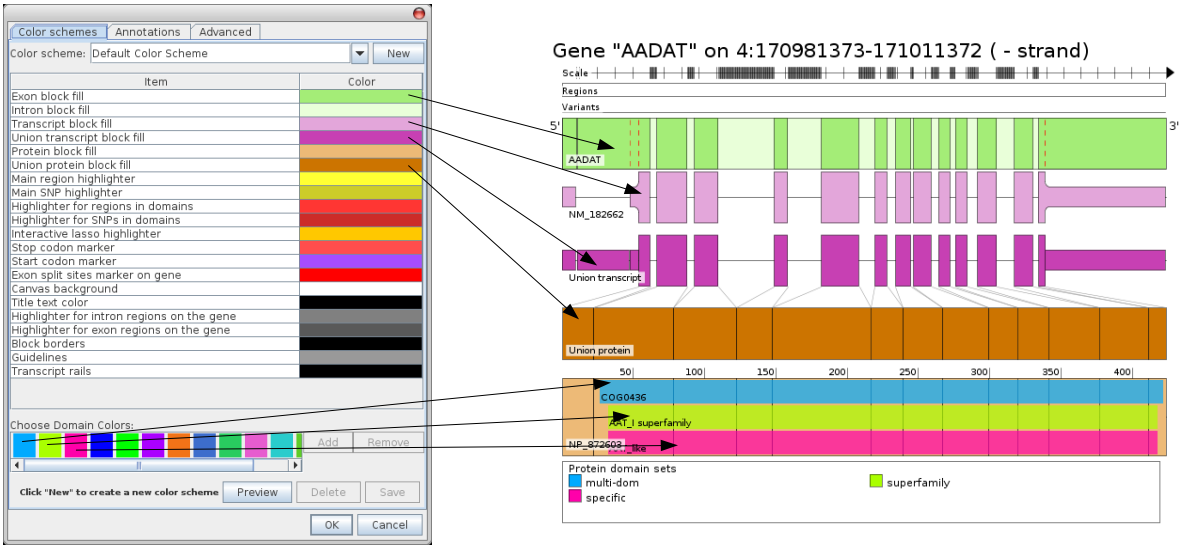
11.1. Color schemes

Since the flashy default colors in GPViz will surely not be suitable for every purpose, GPViz allows you to completely customize the colors in figures. To create a **new color scheme** click the **new** button at the top right corner. This will create a color scheme that is a copy of the one you previously selected (the default scheme if this is your first time).

Once you created your own color scheme, you can **click the colored boxes** in the table to **set a new color** for this specific item. For example, to set the color of the union protein, click the brown box next to “Union protein block fill”. A **color chooser** will appear where you can select the color you want to use.

Below the color table you see a field called **Choose domain colors**. These are the colors used for the different **protein domain sets**. You can add and remove colors using the buttons on the right side, and you can **drag** them around to change the order. When loading protein domains, these colors will be used **from left to right** for the number of domain sets in your input file. So make sure the colors you prefer to use are on the far left. The default color scheme contains 13 different colors, but you don't need to define that many. If you load more domain sets than you have colors defined, the colors will simply repeat.

Here's an example showing a few of the colors in the figure:

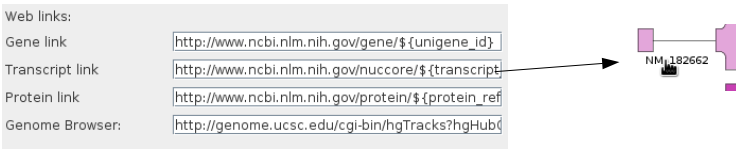


You can also click the **Preview** button to slide open a preview image that will immediately show the changes you made to the color scheme. With the **Delete** button you can also delete existing color schemes.

11.2. Annotations

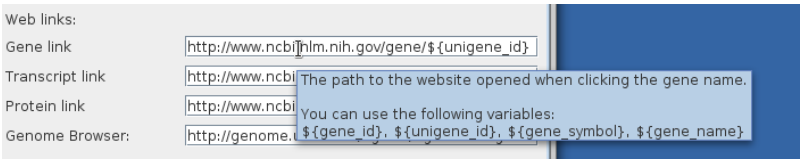
In the upper section you can set the annotation files. Please check out the **Annotation files** section in the **Data formats** chapter for more information.

Below that, you see a number of **web links** for you to configure:



These links define what happens when you click specific areas in the figure. For example when you click the Transcript ID a **web browser** will open and go to the website specified under **Transcript link**. These links are constructed using **variables decorated with ${}**.

If you move your mouse over the text field, you can see which variables you can use.



For example, to change the **Gene link** from a direct link to the NCBI database using the UniGene ID, we can change it to a search query using the gene symbol. So from

http://www.ncbi.nlm.nih.gov/gene/${unigene_id}

we'd change it to:

http://www.ncbi.nlm.nih.gov/gene/?term=${gene_symbol}

12. FAQ

Please check out our FAQ on:

http://www.icbi.at/software/gpviz