geneviz-java: Design Planning Documentation

Table of Contents

1. Software Requirements

1.1 User Needs:

ID	Description	Downstream
UN-001	As an end user I need a GUI-based tool to visualize gene models based on gene structure data.	DI-001 DI-002
UN-002	As an end user I need a tool that I can pass FASTA and GTF files.	DI-003 DI-004 DI-005 DI-006
UN-003	As an end user I need the gene features to be displayed in a list or table.	DI-007
UN-004	As an end user I need the sequence data from the FASTA file displayed as text.	DI-008
UN-005	As an end user I need the interface to be simple in design and easy to navigate.	DI-009
UN-006	As an end user I need the tool to show me basic statistics on data from both files [FASTA, GFF3].	DI-010
UN-007	As an end user I need the tool to load multiple input files. e.g. multiple FASTA files.	DI-003 DI-010 DI-011 DI-016
UN-008	As an end user I need ability to <emphasize> exons in a displayed sequence.	DI-012
UN-009	As an end user I need ability to display a visual representation of the gene model.	DI-013 DI-014 DI-015

Note: Entries reflect user requirements only; they do not guarantee deliverables. Bold elements indicate outstanding implementation.

1.2 Design Inputs

ID	Description	Downstream
DI-001	The tool shall provide the user a front end GUI.	SRS-001
DI-002	The tool shall allow the user to visualize a gene model.	SRS-012
DI-003	The tool shall allow the user means to provide input data — eg a file.	SRS-006 SRS-009
DI-004	The tool shall have a parsing module to parse a FASTA file. Notes: Consider BioJava.	SRS-005
DI-005	The tool shall have a parsing module to parse a GTF file. Notes: GFF3 support should be prioritized, the formats are similar enough so consider implementing both GTF and GFF3. Main difference seems to be the last column. Consider BioJava.	SRS-003 SRS-004
DI-006	The tool shall have a system to store data. Notes: This could be dedicated class objects or an SQL database. If SQL database, this is likely to be an ephemeral database that exists while the use has the application open. Avoid mySQL/postgres as these require hosting on a server. SQLite database is an option to consider.	SRS-004 SRS-005 SRS-006
DI-007	The tool shall allow the user means visualize the gene features (from the GFF3) in a tabular visualization.	SRS-002
DI-008	The tool shall allow the user means visualize the FASTA file as display text.	SRS-007
DI-009	The tool shall allow consider multiple tabs to separate out visual features. Notes: Consider separating out gene-feature table from sequence annotations. Consider where the exon sequence and stats displays fit into the GUI design	SRS-002 SRS-007
DI-010	The tool shall allow the user means visualize basic statistics from the FASTA file.	SRS-008
DI-011	The tool shall allow the user means visualize basic statistics from the GFF3 file.	SRS-010
DI-012	The tool shall allow the FASTA display text to be intractable — either by bulk highlighting exons on import (MVP) or by allowing the user the ability to select a gene and emphasize its exon. Note:	SRS-007 SRS-011

	Could be highlighted, bold, underlined — anything that makes sense (or is easy!?)	
DI-013	The tool shall allow the user a method to select a specific gene.	SRS-011
DI-014	The tool shall allow the user an interface with a visual representation of a gene.	SRS-012
DI-015	Context menu for export (PNG/SVG) of gene model.	
DI-016	The tool shall be able to handle zipped or unzipped files.	SRS-003 SRS-004

Note: Entries reflect design inputs only; they do not guarantee deliverables. Bold elements indicate outstanding implementation.

1.3 Software Requirement Specifications

ID	Description
SRS-001	The tool shall be written in Java since this is an easy to work with framework for GUI applications.
SRS-005	The tool shall be able to parse FASTA formats. Notes: — date accessed: 15-Nov-25: https://www.ncbi.nlm.nih.gov/genbank/fastaformat/ — date accessed: 15-Nov-25: https://www.ncbi.nlm.nih.gov/genbank/mods_fastadefline/ In FASTA format the line before the nucleotide sequence, called the FASTA definition line, must begin with a carat (">"), followed by a unique SeqID (sequence identifier). The SeqID must be unique for each nucleotide sequence and should not contain any spaces. Please limit the SeqID to 25 characters or less. The SeqID can only include letters, digits, hyphens (-), underscores (_), periods (.), colons (:), asterisks (*), and number signs (#). Information about the source organism from which the sequence was obtained follows the SeqID and must be in the format [modifier=text]. Do not put spaces around the "=". At minimum, the scientific name of the organism should be included. Addition modifiers exist. e.g: """ ›SeqABCD [organism=Mus musculus] [strain=C57BL/6] """ The FASTA definition line must not contain any hard returns. All information must be on a single line of text. Note: The format is not fixed, so don't over engineer this part of the code base. Examples from https://www.bioinformatics.nl/tools/crab_fasta.html have different characters in the header, some FASTA files seem to have "|" character.

SRS-003	The tool shall be able to parse GTF formats. Notes: — date accessed: 15-Nov-25 — date accessed: 15-Nov-25 The GTF (General Transfer Format) is identical to GFF version 2. GFF2 can only represent 2 level feature hierarchies, while GFF3 can support arbitrary levels. GFF2 also does not require that column 3, the feature type, be part of the sequence ontology. The GFF format is a flat tab-delimited file, each line of which corresponds to an annotation, or feature. Each line has nine columns and looks like this: """ Chr1 curated CDS 365647 365963 . + 1 Transcript "R119.7" """ reference sequence: This is the ID of the sequence that is used to establish the coordinate system of the annotation. In the example above, the reference sequence is “Chr1”. source: The source of the annotation. This field describes how the annotation was derived. In the example above, the source is “curated” to indicate that the feature is the result of human curation. The names and versions of software programs are often used for the source field, as in “tRNAScan-SE/1.2”. method: The annotation method, also known as type. This field describes the type of the annotation, such as “CDS”. Together the method and source describe the annotation type. start: The start of the annotation relative to the reference sequence. stop: The stop of the annotation relative to the reference sequence. Start is always less than or equal to stop. score: For annotations that are associated with a numeric score (for example, a sequence similarity), this field describes the score. The score units are completely unspecified, but for sequence similarities, it is typically percent identity. Annotations that do not have a score can use “.” strand: For those annotations which are strand-specific, this field is the strand on which the annotation resides. It is “+” for the forward strand, “-“ for the reverse strand, or “.” for annotations that are not stranded. phase: For annotations that are linked to proteins, this field describes the phase of the annotation on the codons. It is a number from 0 to 2, or “.” for features that have no phase. group: GFF provides a simple way of generating annotation hierarchies (“is composed of” relationships) by providing a group field. The group field contains the class and ID of an annotation which is the logical parent of the current one. In the example given above, the group is the Transcript named “R119.7”. The group field is also used to store information about the target of sequence similarity hits, and miscellaneous notes.

SRS-004	The tool shall be able to parse GFF3 formats. Notes: — date accessed: 15-Nov-25: https://gmod.org/wiki/GFF2 — date accessed: 15-Nov-25: https://www.ensembl.org/info/website/upload/gff.html The GFF (General Feature Format) format consists of one line per feature, each containing 9 columns of data, plus optional track definition lines. Fields must be tab-separated. All but the final field in each feature line must contain a value; "empty" columns should be denoted with a '.' seqname: name of the chromosome or scaffold; chromosome names can be given with or without the 'chr' prefix. Important note: the seqname must be one used within Ensembl, i.e. a standard chromosome name or an Ensembl identifier such as a scaffold ID, without any additional content such as species or assembly. See the example GFF output below. source: name of the program that generated this feature, or the data source (database or project name). feature: feature type name, e.g. Gene, Variation, Similarity start: Start position* of the feature, with sequence numbering starting at 1. end: End position* of the feature, with sequence numbering starting at 1. score: A floating point value. strand: defined as + (forward) or - (reverse). frame: One of '0', '1' or '2'. '0' indicates that the first base of the feature is the first base of a codon, '1' that the second base is the first base of a codon, and so on.. attribute: A semicolon-separated list of tag-value pairs, providing additional information about each feature.
SRS-006	User Input module to get file from user. Following Friday's practical I could implement a menu bar with option to load a file — seems generally conventional. I could also have a button somewhere on the GUI to say load file. The requirement is to have the user be able to add their file. The SRS should also restrict the user to opening only permitted file types — eg the user can select a GFF file but cannot select a PDF file. Notes: Check out jMenuBar. Check out jMenu. Check out jMenuItem. Check out FileDialog | JFileChooser
SRS-009	User Input from multiple FASTA / GFF3 files. Note: How shall the tool manage this. Does every FASTA file require a GFF3 file? What happens if the user only provides one.

SRS-002	Visualize tables in GUI. Notes: Check out jTable Component.
SRS-007	Visualize sequences in GUI. Notes: Check out jTextArea Component.
SRS-011	The tool shall have a way to determine what is an exon (from GFF3 file) and what is the start and stop point of this feature (GFF3 file). Notes: Genes can have more than one exon, so you likely need a method that can return the start:stop points for all the exons in a gene. Again, database operations could help. " SELECT exon_id, start, stop, FROM gff_table WHERE gene_id IN ("x1", "x2") " Then select these locations from the sequence and highlight the corresponding values. In addition, I think this approach makes it easy to have a drop down box and select individual genes. Consider an option to have all genes selected, or a selection of genes. Simple way uses jTextPane and HTML tags — explore other options. See HighlightPainter
SRS-008	The tool shall calculate: Length of sequence from FASTA file. Average sequence length (multi-file). GC content from FASTA file. [BioJava: DNASequence.getGCCount()]: —date accessed 15-Nov-25: https://biojava.org/docs/api7.1.4/org/biojava/nbio/core/sequence/DNASequence.html Note: Database operation could make these summaries quick and easy.

SRS-010	The tool shall calculate: Average number of exons per gene from GFF3 file. Longest and shortest gene models from GFF3 file. Average gene length from GFF3 file. Note: Database operation could make these summaries quick and easy
SRS-012	The tool shall render a gene image and superimpose exons (from GFF3). — date accessed: 15-Nov-25: https://biojava.org/docs/api7.1.4/org/biojava/nbio/genome/parsers/gff/FeatureList.html Notes: Consider drawing shapes on a JPanel. Reuse methods from SRS-011 to get start and stop points of exons from a gene. Implement a Utils method to draw a generic rectangle of size y-axis size fixed to gene shape and x-axis determined by the proportionate start/stop from the gene to the length of the gene-shape rectangle. Consider external packages — they look a bit heavy! Add tooltip OnHover to get information on the exon. How does it handle overlapping exons? [omitOverlapping method on the FeatureList] How does it handle multiple FASTA files?

Note: Entries reflect design inputs only; they do not guarantee deliverables. Bold elements indicate outstanding implementation.