Institut für Informatik
Praktische Informatik und Bioinformatik
Prof. Dr. Ralf Zimmer
Dr. Gergely Csaba
Markus Gruber

Wintersemester 2016/2017

GoBi: Excercise 1

Genome Annotation

Deadline: Thursday, 17.11.2016, 14:00

Save your solution to `/home/proj/biocluster/praktikum/genprakt/${team_account}/Solution1`. Provide also an executable jar file (containing also the sources) in this directory that allows for reproducing your results. The jar should print a usage info if invoked without parameters.

Task 1 (Genome annotations):

Implement a basic java framework to handle genome annotations and a parser for GTF files (see `http://www.ensembl.org/info/website/upload/gff.html`).

The framework must represent the genomic information in the classes *Gene, Transcript, CDS* and *Exon*. A *Gene* may contain multiple *Transcripts*, a *Transcript* is composed of a subset of the gene's *Exons* and may have one (or no) *CDS*.

All the classes must have an *id* member (parsed from the GTF file), additional information from the GTF file (e.g. *biotype*, *symbol*) and information about their genomic location, i.e. a tuple (chromosome, strand, start, end), or a list of such tuples. We call such a tuple or a list of those a *Genomic Region* or a *Genomic Region Vector*.

Hints:

- In some GTF files exons do not have an assigned *id*, in such cases generate internal ids.
- Gencode annotates the biotype as gene_type/transcript_type.

Always make sure to use the correct coordinate system (i.e. is a position zero or one based, are start and end of a location inclusive or exclusive).

The parser should return a class named *GenomeAnnotation* supporting the following operations:

- Gene/Transcript/CDS getGene(String id) getTranscript(id) getCDS(id)
- Iterator<Gene> iterator()

- Iterator<Gene> iterator(String chr, Integer start, Integer end)
  provides a list of genes overlapping the region chr: start-end

  if chr == null it should provide an iterator over all genes; otherwise,

  - if both start and end is null it should provide an iterator for all genes of the chromosome
  - if start is null it should provide an iterator for the genes of the chromosome starting before position end
  - if end is null it should provide an iterator for the genes of the chromosome endig after position start

- Gene getGeneBySymbol(String symbol)

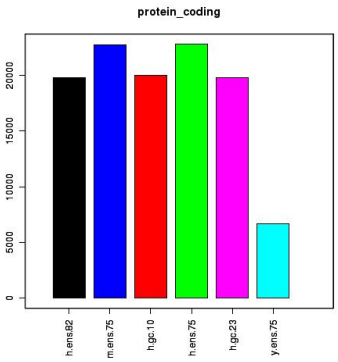
Hint: To get genes overlapping a given location you should use an appropriate data structure. We provide you with a red-black tree based implementation of interval trees
`/home/proj/biosoft/praktikum/genprakt/libs/augmentedTree.jar`, class: `IntervalTree`.

Task 2 (Compare number of annotated genes):

You find 6 GTF files, and a file named `annot.map` in
`/home/proj/biosoft/praktikum/genprakt/assignment/a1/data`.

Use your parser from Task 1 to compare the number of annotated genes per biotype in the different GTF files. Write a table to a file named `genetypes.txt` in your solution directory containing for every biotype and every gtf-name (first column in `annot.map`) the number of annotated genes per biotype in the corresponding gtf. The biotypes should be ordered by their total occurrence in all GTF files.
In addition, create barplots for each biotype and add them to a html file `genetypes.html` in this order.

Example of such a barplot:



Task 3 (Compare number of transcripts in annotated genes):

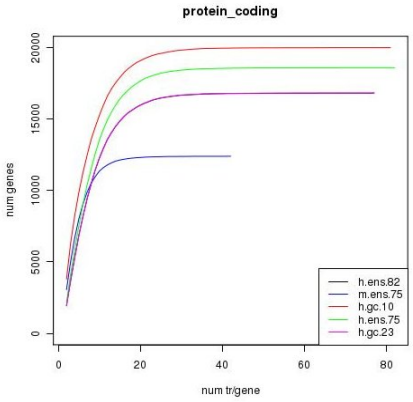
Extract the number of transcripts per gene using the same files as in Task 2. Create an html file showing the cumulative distributions of the number of transcripts for each biotype in the different GTF files. Write the biotypes sorted by the total number of genes having multiple transcripts.

The GTF file with the key **h.ens.86** refers to the current ENSEMBL annotation. Supply for every biotype a list of information (see below) and a link to the gene on the current ENSEMBL website for 10 genes with the most transcripts in this GTF.

Provide the information: id, symbol, biotype, chromosome, strand, start, end, number of transcripts, number of CDS-s. The url for a given gene on the ENSEMBL website is given by:

`http://www.ensembl.org/Homo_sapiens/Gene/Summary?db=core;g=<geneid>`

Example for such a cumulative distribution plot and html links:



- ADGRG1 ENSG00000205336(16+ protein_coding 57610652-57665580) num transcripts: 77 num proteins: 65
- NDRG2 ENSG00000165795(14- protein_coding 21016763-21070872) num transcripts: 77 num proteins: 60
- TCF4 ENSG00000196628(18- protein_coding 55222331-55664787) num transcripts: 75 num proteins: 62
- DMKN ENSG00000161249(19- protein_coding 35497220-35513658) num transcripts: 68 num proteins: 47
- SORBS2 ENSG00000154556(4- protein_coding 185585444-185956652) num transcripts: 64 num proteins: 40
- ST5 ENSG00000166444(11- protein_coding 8693351-8910951) num transcripts: 62 num proteins: 43
- DDR1 ENSG00000204580(6+ protein_coding 30876421-30900156) num transcripts: 58 num proteins: 46
- GNAS ENSG00000087460(20+ protein_coding 58839718-58911192) num transcripts: 57 num proteins: 21
- PTK2 ENSG00000169398(8- protein_coding 140657900-141002216) num transcripts: 57 num proteins: 41
- EEF1D ENSG00000104529(8- protein_coding 143579697-143599541) num transcripts: 56 num proteins: 50

Task 4 (Compare number of overlapping genes):

Extract the list of pairs of overlapping genes for every input GTF file. As genes may be located on the plus or minus strand, there are three different types of overlap: (i) overlapping genes disregarding their strands, (ii) overlapping genes on the same strand, (iii) overlapping genes on different strands. Write overlapping gene pairs of all three types into the file `${gtfname}.overlaps.tsv` into your solution directory (for each of the gtf files. These tab separated files must have the header: *geneid*1, *geneid*2, *strand*1, *strand*2, *biotype*1, *biotype*2, *num_overlapping_bases*.

Create an html file showing the cumulative distribution of overlapping genes for all three overlap types per biotype-pairs, showing the biotype-pairs ordered (descending) by their total number (over all GTF files) of overlapping genes.

Similar to Task 3 add a list to the html output for each biotype pair with the ten most overlapping genes in the current ENSEMBL version with links to the corresponding genes.

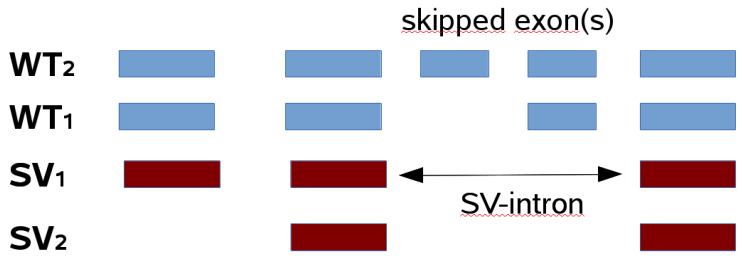
Task 5 (Analyze the transcribed lengths of genes):

Implement a method that calculates for a gene the "union transcript" i.e. the genomic region vector that covers all transcripts annotated by the gene (and nothing more).
Calculate for every gene the proportion "length of longest transcript"/"length of union transcript" in every gtf, plot their cumulative distributions and collect those plots in the html file `transcript_lengths.html`.

Task 6 (Analyze exon skippings):

One form of alternative splicing is exon skipping. An exon-skipping splicing event is a tuple (gene, intron-start, intron-end) and is defined by (at least) two transcripts: wildtype (WT) and spliced variant (SV) of the same gene, and an intron in SV with start and end corresponding to an exon end and exon start in WT, respectively, and the SV-intron spans at least one exon in WT.

For any exon-skip event there may be several WT-s and several SV-s, and there may be several sets of skipped exons (see figure below; this is **one** exon-skipping event).



Extract all exon-skipping events between CDS-es for each gtf file and write the results into tsv files `<gtfname>_exonskipping.tsv` with the following headers:

- id (gene id)
- symbol (gene symbol)
- chr (chromosome)

- strand (+ or -)
- nprots (number of annotated CDS in the gene)
- ntrans (number of annotated transcripts in the gene)
- SV (the SV intron as start:end)
- WT (the WT introns within the SV intron separated by | as start:end)
- SV_prots (ids of the SV CDS-s, separated by |)
- WT_prots (ids of the WT CDS-s, separated by |)
- min_skipped_exon the minimal number of skipped exons in any WT/SV pair
- max_skipped_exon the maximum number of skipped exons in any WT/SV pair
- min_skipped_bases the minimal number of skipped bases (joint length of skipped exons) in any WT/SV pair
- max_skipped_bases the maximum number of skipped bases (joint length of skipped exons) in any WT/SV pair

Write two cumulative plots into your output directory named `skipped_exons.jpg` and `skipped_bases.jpg` showing the distributions of the maximum number of skipped exons / skipped bases per event for the different GTF files, and an html file `exon_skipping.html` showing these plots and linking the top 10 genes for both criteria in the current ENSEMBL version.

Task 7 (Compare genome versions):

While the human genome was 90% completed already in the year 2000 and announced to be complete in 2003, there are still regularly changes both in the overall assembly and the gene annotation. The GTF-s with key **h.gc.10** and **h.gc.23** are based on different assemblies provided by GENCODE.

Compare the set of genes annotated in both versions (gene id up to the dot):

- For how many genes has the chromosome been changed?
- For all others, create a cumulative plot on the absolute chromosomal distance (minimum of start / end differences) of the genes (`chrdist.jpg`)
- Create a cumulative plot on the gene length differences (`glengthdiff.jpg`)
- Create a cumulative plot for the distribution of differences of the number of annotated transcripts / CDS-s (two curves) (`andiff.jpg`)

Collect these plots in the html file `genome_versions.html`.