Ensembl Gene Annotation (*e!98*)

Cat (*Felis catus*)

Assembly: Felis_catus_9.0, GCA_000181335.4

Table of Contents

This document describes the annotation process of an assembly. The first stage is Assembly Loading where databases are prepared and the assembly loaded into the database.

Section 1: Genome Preparation

The genome phase of the Ensembl gene annotation pipeline (Fig. 1) involves loading an assembly into the Ensembl core database schema and then running a series of analyses on the loaded assembly to identify an initial set of genomic features. The most important aspect of this phase is identifying repeat features (primarily through RepeatMasker) as soft masking of the genome is used extensively later in the annotation process.

Repeat Finding

After the genomic sequence has been loaded into a database, it is screened for sequence patterns including repeats using RepeatMasker [1] (version 4.0.5 with parameters, *-nolow -engine "crossmatch"*), Dust [2] and TRF [3]. For the cat annotation, the Repbase mammals library was used with RepeatMasker. In addition to the Repbase library, a custom repeat library was used with RepeatMasker. This custom library was created using RepeatModeler [1].

Low complexity features, ab initio predictions and BLAST analyses

Transcription start sites are predicted using Eponine–scan [4]. CpG islands longer than 400 bases and tRNAs are also predicted. The results of Eponine-scan, CpG, and tRNAscan [5] are for display purposes only; they are not used in the gene annotation process. Genscan [6] is run across repeat-masked sequence to identify ab initio gene predictions. The results of the Genscan analyses are also used as input for UniProt [7], UniGene [8] and Vertebrate RNA alignments by NCBI-BLAST [9]. Passing only Genscan results to BLAST is an effective way of reducing the search space and therefore the computational resources required.

Genscan predictions are for display purposes only and are not used in the model generation phase.

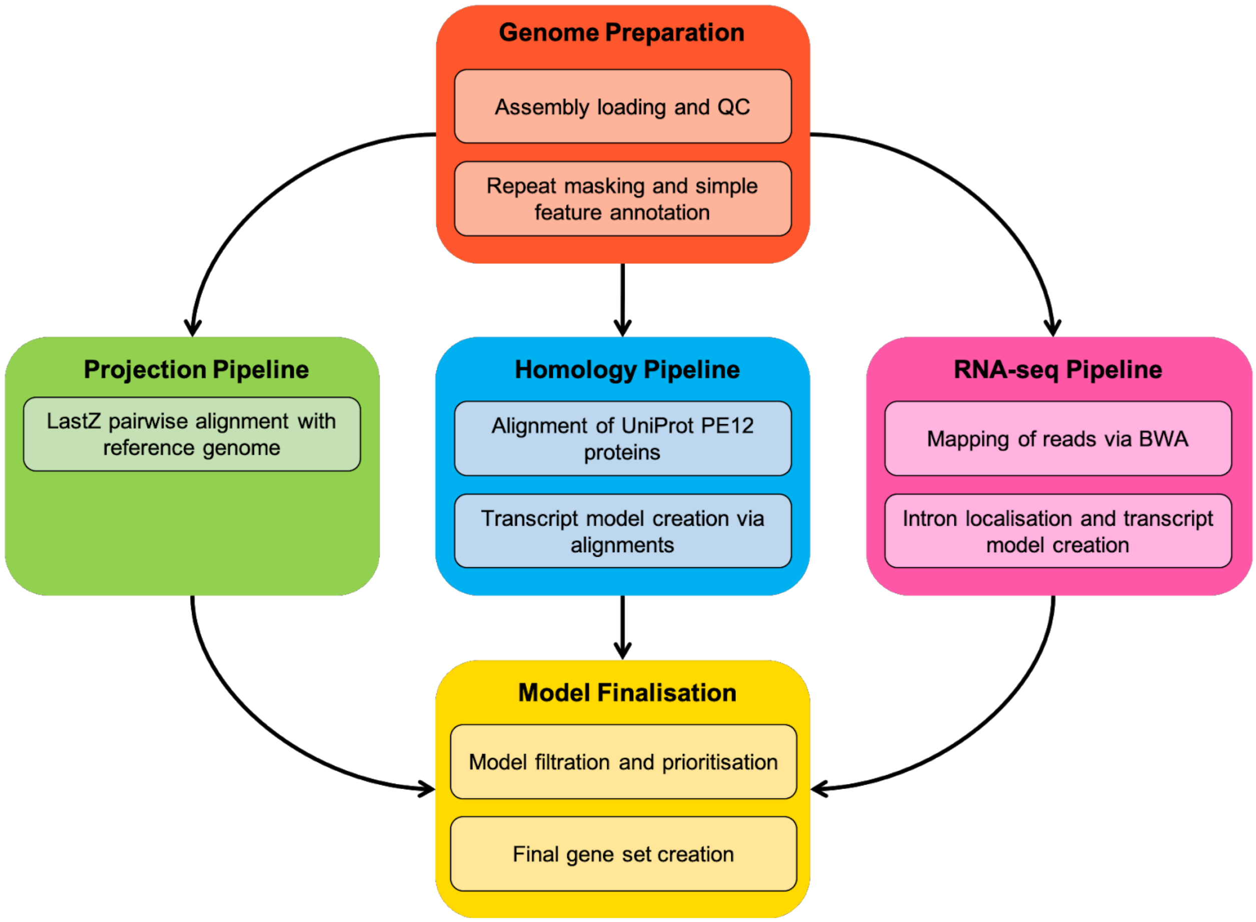

Fig. 1: Flowchart of the protein-coding annotation pipeline. Small ncRNAs, Ig genes, TR genes, and pseudogenes are computed using separate pipelines.

Section 2: Protein-Coding Model Generation

Various sources of transcript and protein data are investigated and used to generate gene models using a variety of techniques. The data and techniques employed to generate models are outlined here. The numbers of gene models generated are described in gene summary.

Species specific cDNA and protein alignments

cDNAs are downloaded from ENA (www.ebi.ac.uk/ena) and RefSeq [10], and aligned to the genome using Exonerate [11]. Only known mRNAs are used (NMs). The cDNAs can be used to add UTR to the protein coding transcript models if they have a matching set of introns. Proteins are downloaded from UniProt and filtered based on protein existence (PE) at protein level and transcript level. The proteins are aligned to the genome using PMATCH to reduce the search space, then with genewise, which is a splice-aware aligner, to generate spliced models.

Projection mapping pipeline

For all species we generate a whole genome alignment against a suitable reference assembly using LastZ [12]. Syntenic regions identified using this alignment are then used to map protein-coding annotation from the most recent closely-related species or the Ensembl/GENCODE [13] gene set. For the cat annotation, we used the human assembly GRCh38.p12 as a reference to map protein-coding annotation. For each protein-coding gene in human, we projected the coding exons within the canonical transcript to the cat. In case of exonic overlap on the projected sequence, the longest exon took precedence. If the mapping did not succeed, we selected the next successful projection of the transcript having the longest translation.

Protein-to-genome pipeline

Protein sequences were downloaded from UniProt and aligned to the genome in a splice aware manner using GenBlast [12]. The set of proteins aligned to the genome is a subset of UniProt proteins used to provide a broad, targeted coverage of the cat proteome. The set consists of the following:

- Self SwissProt/TrEMBL PE 1 & 2
- Human SwissProt/TrEMBL PE 1 & 2
- Mouse SwissProt/TrEMBL PE 1 & 2
- Other mammals SwissProt/TrEMBL PE 1 & 2
- Vertebrates SwissProt/TrEMBL PE 1 & 2

Note: PE level = protein existence level

For the cat annotation, logical thresholds for coverage (50%), percent identity (30%) and e-value (E-1) were used for GenBlast with the exon repair option turned on. The top 10 transcript models built by GenBlast for each protein passing the cut-offs are kept.

RNA-seq pipeline

RNA-seq data was downloaded from ENA (https://www.ebi.ac.uk/ena/) and used in the annotation. A merged file containing reads from all tissues/samples was created. The merged data is less likely to suffer from model fragmentation due to read depth. The available reads were aligned to the genome using BWA [13], with a tolerance of 50% mismatch to allow for intron identification via split read alignment. Initial models generated from the BWA alignments were further refined using exonerate. Protein-coding models were identified fromBLAST alignments of the longest ORF against the UniProt vertebrate PE 1 & 2 data set. create a gene track for each tissue/sample viewable in the Ensembl browser and accessible through the Ensembl API.

Immunoglobulin and T-cell Receptor genes

Translations of different human IG gene segments were downloaded from the IMGT database [14] and aligned to the genome using GenBlast. For the cat annotation, logical thresholds for coverage (80%), percent identity (70%) and e-value (E-1) were used for GenBlast with the exon repair option turned on. The top 10 transcript models built by GenBlast for each protein passing the cut-offs are kept.

Selenocysteine proteins

We aligned known selenocysteine proteins against the genome using Exonerate, checking that the generated model had a selenocysteine in the same positions as the known protein. We only kept models with at least 90% coverage and 95% identity.

Section 3: Filtering the Protein-Coding Models

The filtering phase decides the subset of protein-coding transcript models, generated from the model-building pipelines, that comprise the final protein-coding gene set. Models were filtered based on information such as: what pipeline was used to generate them, how closely related to the target species and how good the alignment coverage and percent identity are to the original data.

Prioritising models at each locus

The LayerAnnotation module was used to define a hierarchy of input data sets, from most preferred to least preferred. The output of this pipeline includes all transcript models from the highest ranked input set. Models from lower ranked input sets were included only if their exons do not overlap a model from an input set higher in the hierarchy. Note that models cannot exist in more than one layer. For UniProt proteins, models were separated into clades, to help selection during the layering process. Each UniProt protein was in one clade only, for example mammal proteins were present in the mammal clade and not in the vertebrate clade to avoid aligning proteins multiple times.

When selecting the model or models kept at each position, we prioritise based on the highest layer with available evidence. In general, the highest layers contain the set of evidence containing the most trustworthy evidence in terms of both alignment/mapping quality, and also in terms of relevance to the species being annotated. So, for example, when a primate is being annotated, well aligned evidence from either the species itself or other closely related vertebrates would be chosen over evidence from more distant species. Regardless of what species is being annotated, well-aligned human proteins are usually included in the top layer as human is the current most complete vertebrate annotation. For further details on the exact layering used please refer to section 6.

Addition of UTR to coding models

The set of coding models were extended into the untranslated regions (UTRs) using RNA-seq data and alignments of species-specific RefSeq cDNA sequences. The criteria for adding UTR from cDNA or RNA-seq alignments to protein models lacking UTR (such as the projection models or the protein-to-genome alignment models) is that the intron coordinates from the model missing UTR exactly match a subset of the coordinates from the UTR donor model.

Generating multi-transcript genes

The above steps generated a large set of potential transcript models, many of which overlap one another. Redundant transcript models were collapsed and the remaining unique set of transcript models were clustered into multi-transcript genes where each transcript in a gene has at least one coding exon that overlaps a coding exon from another transcript within the same gene.

Pseudogenes

Pseudogenes were annotated by looking for genes with evidence of frame-shifting or lying in repeat heavy regions. Single exon retrotransposed pseudogenes were identified by searching for a multi-exon equivalent elsewhere in the genome. Identified pseudogenes and processed pseudogenes are included in the core db, please check Final Gene set Summary (Fig 2).

Immunoglobulin and T-cell Receptor genes

Translations of different human IG gene segments were downloaded from the IMGT database [14] and aligned to the genome using GenBlast. For the cat annotation, logical thresholds for coverage (80%), percent identity (80%) and e-value (E-1) were used for GenBlast with the exon repair option turned on. The top 10 transcript models built by GenBlast for each protein passing the cut-offs are kept.

Section 4: Creating the Final Gene Set

Small ncRNAs

Small non-coding (sncRNA) genes were added using annotations taken from RFAM [15] and miRBase [16]. For miRNAs, NCBI-BLAST was run for these sequences to identify homologs in the genome sequence and models were evaluated for expected stem-loop structures using RNAfold [17]. Additional machine learning based filters were applied to exclude predictions with sub-optimal alignments to the genome and non-conforming secondary structures. For other sncRNAs, models were built using the Infernal software suite [18].

Cross-referencing

Before public release the transcripts and translations were given external references (cross-references to external databases). Translations were searched for signatures of interest and labelled where appropriate.

Stable Identifiers

Stable identifiers were assigned to each gene, transcript, exon and translation. When annotating a species for the first time, these identifiers are auto-generated. In all subsequent annotations for a species, the stable identifiers are propagated based on comparison of the new gene set to the previous gene set.

Section 5: Final Gene Set Summary

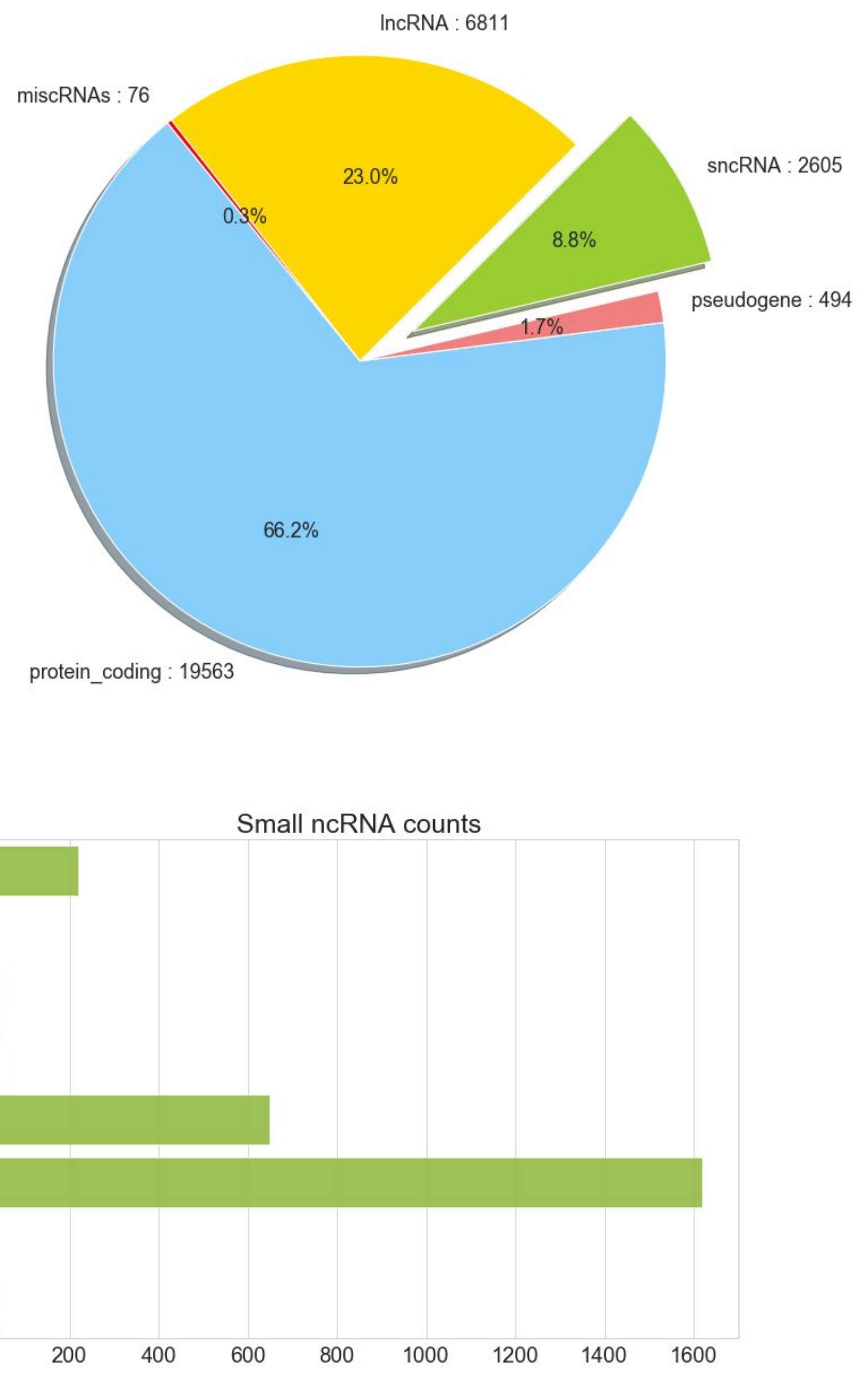

Figure 2: Counts of the major gene classes in cat

Section 6: Appendix - Further information

The Ensembl gene set is generated automatically, meaning that gene models are annotated using the Ensembl gene annotation pipeline. The main focus of this pipeline is to generate a conservative set of protein-coding gene models, although non-coding genes and pseudogenes may also be annotated. Every gene model produced by the Ensembl gene annotation pipeline is supported by biological sequence evidence (see the “Supporting evidence” link on the left-hand menu of a Gene page or Transcript page); *ab initio* models are not included in our gene set. *Ab initio* predictions and the full set of cDNA and EST alignments to the genome are available on our website.

The quality of a gene set is dependent on the quality of the genome assembly. Genome assembly can be assessed in a number of ways, including:

1. Coverage estimates
    - A higher coverage usually indicates a more complete assembly.
    - Using Sanger sequencing only, a coverage of at least 2x is preferred.
2. N50 of contigs and scaffolds
    - A longer N50 usually indicates a more complete genome assembly.
    - Bearing in mind that an average human gene may be 10-15 kb in length, contigs shorter than this length will be unlikely to hold full-length gene models.
3. Number of contigs and scaffolds
    - A lower number top level sequences usually indicates a more complete genome assembly.
4. Alignment of cDNAs and ESTs to the genome
    - A higher number of alignments, using stringent thresholds, usually indicates a more complete genome assembly.

Assembly Information

Table 1: Assembly information

Species	Common name	Assembly	Genbank accession	Date released
Felis catus	cat	Felis_catus_9.0	GCA_000181335.4	2017-11

Statistics of Interest

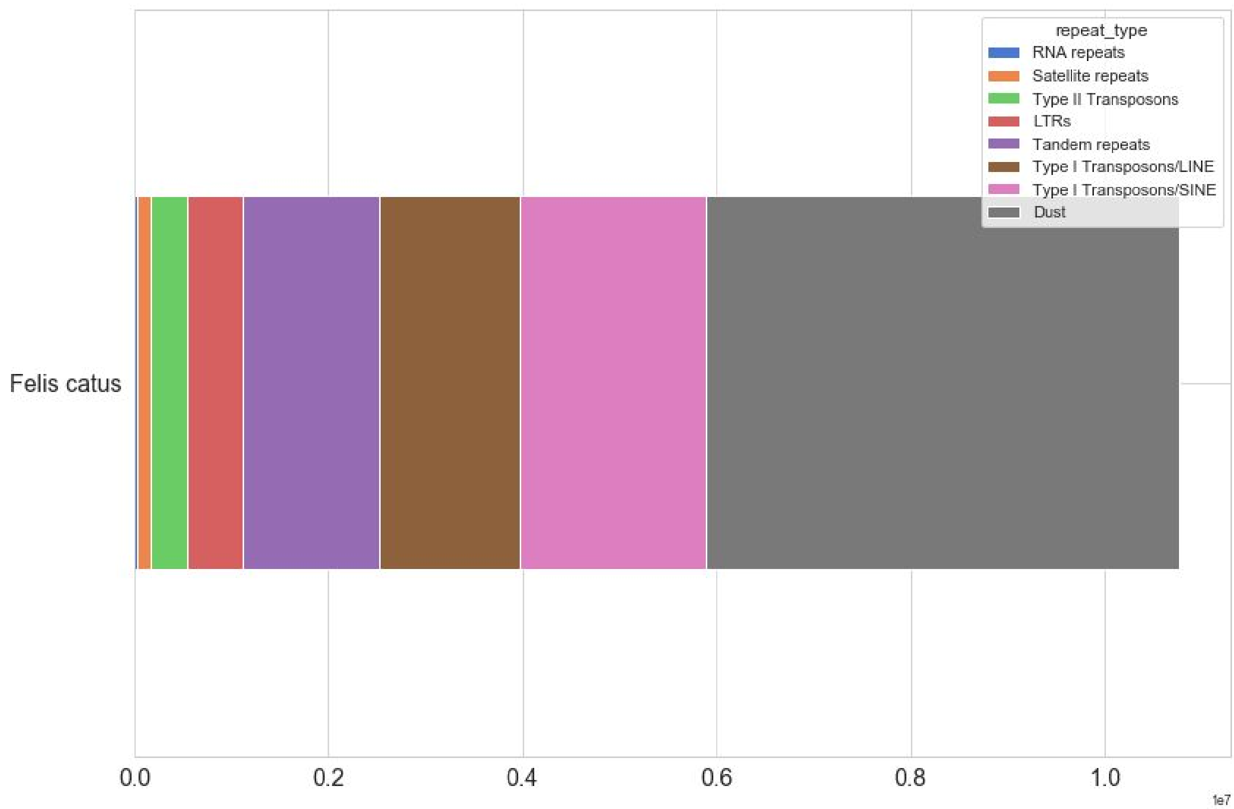

Figure 3: Number of repeat features identified in cat using "repbase_mammals" repeat library.

Table 2: Transcript sizes

biotype	min	median	max
IG_C_gene	6038	6038	6038
IG_V_gene	311	329	1299
Mt_rRNA	959	1266	1573
Mt_tRNA	58	68	74
TR_C_gene	422	422	422
TR_J_gene	59	77	77
TR_V_gene	284	323	8342
Y_RNA	78	101	114
lncRNA	163	8216	718779
miRNA	54	80	143
misc_RNA	102	293	314
processed_pseudogene	146	254	1337
protein_coding	74	27879	2129751
pseudogene	71	750.5	281367
rRNA	105	118	3644
ribozyme	230	305	334
scaRNA	80	135	410
snRNA	51	106	250
snoRNA	48	107	328
vaultRNA	89	93.5	98

Table 3: Transcript sizes (spliced)

biotype	min	median	max
IG_C_gene	924	924	924
IG_V_gene	273	320	332
Mt_rRNA	959	1266	1573
Mt_tRNA	58	68	74
TR_C_gene	422	422	422
TR_J_gene	59	77	77
TR_V_gene	284	323	582
Y_RNA	78	101	114
lncRNA	62	1058	10274
miRNA	54	80	143
misc_RNA	102	293	314
processed_pseudogene	146	254	1337
protein_coding	40	1914	81136
pseudogene	66	579	25730
rRNA	105	118	3644
ribozyme	230	305	334
scaRNA	80	135	410
snRNA	51	106	250
snoRNA	48	107	328
vaultRNA	89	93.5	98

Layers in detail

Layer 1

'IG_C_gene','IG_J_gene','IG_V_gene','IG_D_gene','TR_C_gene','TR_J_gene','TR_V_gene','TR_D_gene','seleno_self'

Layer 2

'cdna2genome','edited','gw_gtag','gw_nogtag','gw_exo','rnaseq_merged_1','rnaseq_merged_2','rnaseq_merged_3','rnaseq_tissue_1','rnaseq_tissue_2','rnaseq_tissue_3','self_pe12_sp_1','self_pe12_tr_1','self_pe12_sp_2','self_pe12_tr_2','projection_1','projection_2','projection_3'

Layer 3

'rnaseq_merged_4','rnaseq_tissue_4','human_pe12_sp_1','human_pe12_tr_1','mouse_pe12_sp_1','mouse_pe12_tr_1','human_pe12_tr_2','human_pe12_sp_2','mouse_pe12_sp_2','mouse_pe12_tr_2','genblast_rnaseq_top','projection_4'

Layer 4

'rnaseq_merged_5','rnaseq_tissue_5','mammals_pe12_sp_1','mammals_pe12_tr_1','mammals_pe12_sp_2','mammals_pe12_tr_2','self_pe3_sp_1','self_pe3_tr_1','genblast_rnaseq_high'

Layer 5

'human_pe12_sp_3','human_pe12_tr_3','mouse_pe12_sp_3','mouse_pe12_tr_3','human_pe12_sp_4','human_pe12_tr_4','mouse_pe12_sp_4','mouse_pe12_tr_4','genblast_rnaseq_medium'

Layer 6

'mammals_pe12_sp_3','mammals_pe12_tr_3','mammals_pe12_sp_4','mammals_pe12_tr_4'

Layer 7

'rnaseq_merged_6','rnaseq_tissue_6','human_pe12_sp_int_1','human_pe12_tr_int_1','human_pe12_sp_int_2','human_pe12_tr_int_2','human_pe12_sp_int_3','human_pe12_tr_int_3','human_pe12_sp_int_4','human_pe12_tr_int_4','mouse_pe12_sp_int_1','mouse_pe12_tr_int_1','mouse_pe12_sp_int_2','mouse_pe12_tr_int_2','mouse_pe12_sp_int_3','mouse_pe12_tr_int_3','mouse_pe12_sp_int_4','mouse_pe12_tr_int_4','mammals_pe12_sp_int_1','mammals_pe12_tr_int_1','mammals_pe12_sp_int_2','mammals_pe12_tr_int_2','mammals_pe12_sp_int_3','mammals_pe12_tr_int_3','mammals_pe12_sp_int_4','mammals_pe12_tr_int_4'

Layer 8

'rnaseq_merged_7','rnaseq_tissue_7'

Layer 9

'rnaseq_merged','rnaseq_tissue'

More information

More information on the Ensembl automatic gene annotation process can be found at:

- Publication : Aken B et al.: The Ensembl gene annotation system. Database 2016.
- Web: Link to Ensembl gene annotation documentation

References

1. Smit, A., R. Hubley, and P. Green, http://www.repeatmasker.org/. RepeatMasker Open, 1996. **3**: p. 1996-2004.
2. Kuzio, J., R. Tatusov, and D. Lipman, *Dust.* Unpublished but briefly described in: Morgulis A, Gertz EM, Schäffer AA, Agarwala R. A Fast and Symmetric DUST Implementation to Mask Low-Complexity DNA Sequences. Journal of Computational Biology, 2006. **13**(5): p. 1028-1040.
3. Benson, G., *Tandem repeats finder: a program to analyze DNA sequences.* Nucleic acids research, 1999. **27**(2): p. 573.
4. Down, T.A. and T.J. Hubbard, *Computational detection and location of transcription start sites in mammalian genomic DNA.* Genome research, 2002. **12**(3): p. 458-461.
5. Lowe, T.M. and S.R. Eddy, *tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence.* Nucleic acids research, 1997. **25**(5): p. 955-964.
6. Burge, C. and S. Karlin, *Prediction of complete gene structures in human genomic DNA.* Journal of molecular biology, 1997. **268**(1): p. 78-94.
7. Consortium, U., *UniProt: the universal protein knowledgebase.* Nucleic acids research, 2017. **45**(D1): p. D158-D169.
8. Pontius, J.U., L. Wagner, and G.D. Schuler, *21. UniGene: A unified view of the transcriptome.* The NCBI Handbook. Bethesda, MD: National Library of Medicine (US), NCBI, 2003.
9. Altschul, S.F., et al., *Basic local alignment search tool.* Journal of molecular biology, 1990. **215**(3): p. 403-410.
10. O'Leary, N.A., et al., *Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation.* Nucleic acids research, 2015. **44**(D1): p. D733-D745.
11. Slater, G.S.C. and E. Birney, *Automated generation of heuristics for biological sequence comparison.* BMC bioinformatics, 2005. **6**(1): p. 31.

12. She, R., et al., *genBlastG: using BLAST searches to build homologous gene models.* Bioinformatics, 2011. **27**(15): p. 2141-2143.
13. Li, H. and R. Durbin, *Fast and accurate short read alignment with Burrows–Wheeler transform.* Bioinformatics, 2009. **25**(14): p. 1754-1760.
14. Lefranc, M.-P., et al., *IMGT®, the international ImMunoGeneTics information system® 25 years on.* Nucleic acids research, 2014. **43**(D1): p. D413-D422.
15. Griffiths-Jones, S., et al., *Rfam: an RNA family database.* Nucleic acids research, 2003. **31**(1): p. 439-441.
16. Griffiths-Jones, S., et al., *miRBase: microRNA sequences, targets and gene nomenclature.* Nucleic acids research, 2006. **34**(suppl_1): p. D140-D144.
17 . R. Lorenz, S.H. Bernhart, C. Hoener zu Siederdissen, H. Tafer, C. Flamm, P.F. Stadler and I.L. Hofacker (2011), "ViennaRNA Package 2.0", Algorithms for Molecular Biology: 6:26
18. Nawrocki, E.P. and S.R. Eddy, *Infernal 1.1: 100-fold faster RNA homology searches.* Bioinformatics, 2013. **29**(22): p. 2933-2935.
19. Finn, R.D., et al., *The Pfam protein families database: towards a more sustainable future.* Nucleic acids research, 2016. **44**(D1): p. D279-D285.