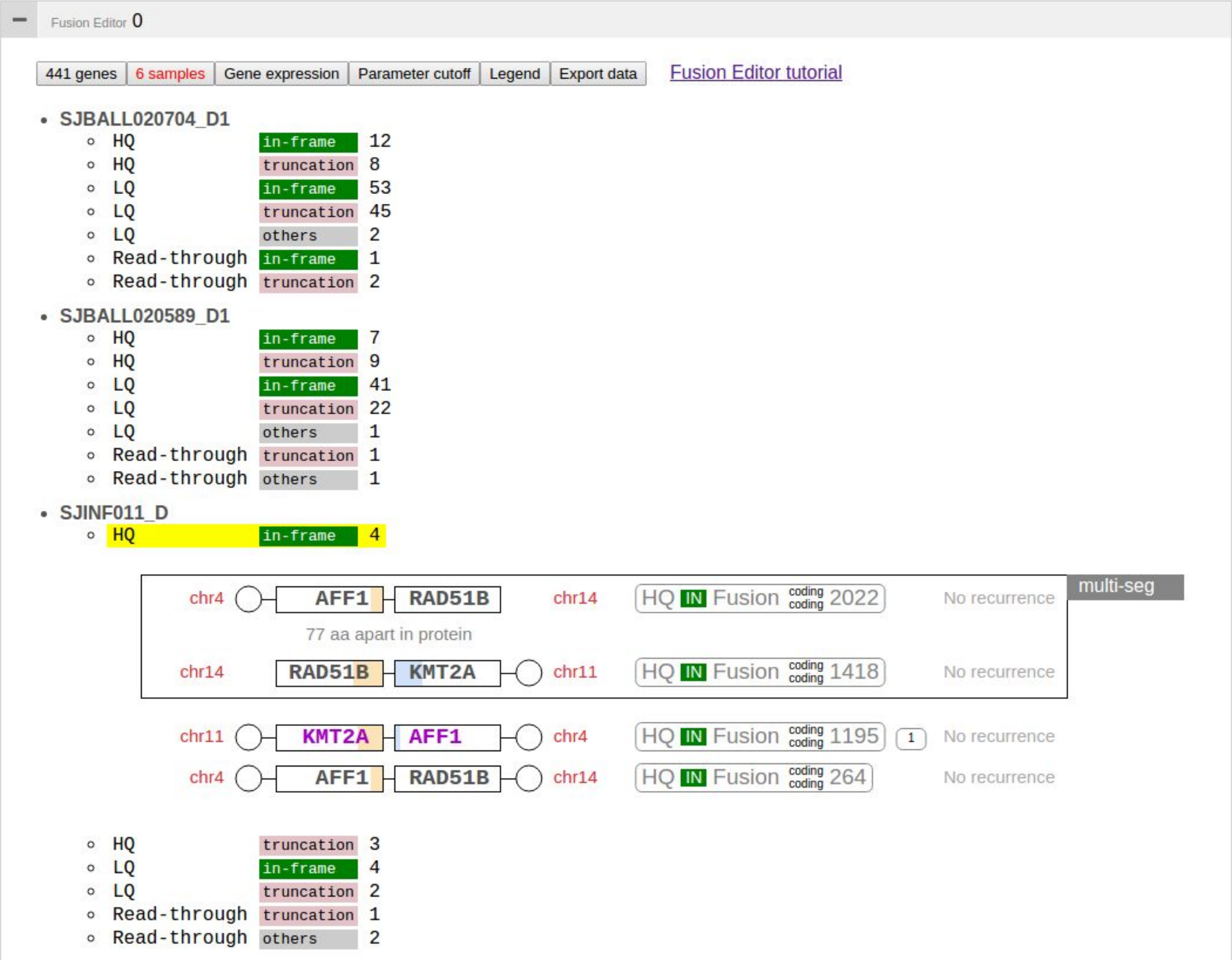
Fusion transcript data format by CICERO

CICERO is a program that detects fusion transcripts from RNA-seq results (CICERO: Yongjin Li, Michael Edmonson, and Jinghui Zhang, manuscript under preparation). In ProteinPaint, user can upload a CICERO output file into the Fusion Editor to explore the data and make reviews and edits.


Fusion Editor 0
441 genes
6 samples
Gene expression
Parameter cutoff
Legend
Export data
Fusion Editor tutorial

- SJBALL020704_D1
  - HQ in-frame 12
  - HQ truncation 8
  - LQ in-frame 53
  - LQ truncation 45
  - LQ others 2
  - Read-through in-frame 1
  - Read-through truncation 2
- SJBALL020589_D1
  - HQ in-frame 7
  - HQ truncation 9
  - LQ in-frame 41
  - LQ truncation 22
  - LQ others 1
  - Read-through truncation 1
  - Read-through others 1
- SJINF011_D
  - HQ in-frame 4

chr4 AFF1 RAD51B chr14 HQ IN Fusion coding coding 2022 No recurrence
multi-seg
77 aa apart in protein
chr14 RAD51B KMT2A chr11 HQ IN Fusion coding coding 1418 No recurrence
chr11 KMT2A AFF1 chr4 HQ IN Fusion coding coding 1195 1 No recurrence
chr4 AFF1 RAD51B chr14 HQ IN Fusion coding coding 264 No recurrence

  - HQ truncation 3
  - LQ in-frame 4
  - LQ truncation 2
  - Read-through truncation 1
  - Read-through others 2

General notes

- The upload file is a line-oriented tabular text file. The first line must be the header with required columns.
- Order of columns does not matter.
- Lines starting with # will be ignored.
- The same header line can appear subsequently in the file and will be ignored. This is allowed since CICERO outputs one text file for each sample, with one header line for each output file. To

upload to ProteinPaint, these files should be concatenated into one file, and the header lines are tolerated.

  - Tip: by concatenating and uploading data from multiple samples, recurrency of fusion events can be detected.

- To upload file and visualize the content, run the command “\cicero” at gene search box in ProteinPaint.
- User can edit:
  - Rating
  - In- or out-of-frame
  - Fusion type
  - Functional effect
  - Multiple breakpoints can be joined to form multi-segment fusions

File columns

- **sample**
  - name of the sample
- **geneA**
- **chrA**
- **posA**
  - 0-based
- **ortA**
- **featureA**
- **geneB**
- **chrB**
- **posB**
  - 0-based
- **ortB**
- **featureB**
- **sv_ort**
- **readsA**
  - Number of chimeric reads from geneA
- **readsB**
  - Number of chimeric reads from geneB
- **matchA**
- **matchB**
- **repeatA**
- **repeatB**
- **coverageA**
- **coverageB**
- **ratioA**
- **ratioB**
- **qposA**
- **qposB**

- **contig**
- **type**
    - not used
- **score**
- **rating**
    - Supported values: HQ, LQ, RT, bad
- **medal**
    - Allowed values: 0/1/2/3/4
    - 0: neither are known fusion partner
    - 1: only geneA is known fusion partner
    - 2: only geneB is known fusion partner
    - 3: both geneA and B are fusion partner, yet they do not form a known fusion product
    - 4: this is a known fusion product (highest medal)
- **functional effect**
    - Fusion_type, allowed values:
        - Fusion
        - CLoss
            - C-term loss
        - NLoss
            - N-term loss
        - ITD
            - Internal duplication (always in-frame)
        - upTSS
            - Upstream of translation start site
            - The fusion product frame will be set to “in-frame” upon “upTSS” case, disregarding original frame status
        - other
    - internally “type2”
- **frame**
    - internally “pair.frame”
- **sv_refseqA**
    - internally “pair.a.isoform”
- **sv_refseqA_codon**
    - internally “pair.a.codon”
- **sv_refseqA_exon**
    - internally “pair.a.exon”
- **sv_refseqA_anchor_type**
    - internally “pair.a.anchor”
- **sv_refseqA_coding_base_number**
- **sv_refseqA_last_coding_base_number**
- **sv_refseqA_AA_index**
    - 0-based index in amino acid frame of end of geneA anchoring
    - internally “pair.a.contigaa”
- **sv_refseqA_contig_index**
    - 0-based index in contig nucleotide sequence of last base of last codon in geneA anchoring

    - internally “pair.a.contigbp”
- **sv_refseqB**
    - internally “pair.b.isoform”
- **sv_refseqB_codon**
- **sv_refseqB_exon**
- **sv_refseqB_anchor_type**
    - internally “pair.b.anchor”
- **sv_refseqB_coding_base_number**
- **sv_refseqB_last_coding_base_number**
- **sv_refseqB_AA_index**
    - 0-based index in amino acid frame of start of geneB anchoring
- **sv_refseqB_contig_index**
    - 0-based index in contig nucleotide sequence of first base of first codon in geneB anchoring
- **sv_AA**
- **sv_desc**
- **sv_processing_exception**
- **sv_general_info**
- **sv_interstitial_AA**
    - amino acid sequence between end of geneA anchoring and beginning of geneB anchoring
- **sv_frame_index**
    - 0-based index in the nucleotide contig sequence the protein frame shown in the sv_AA field started from (0, 1, or 2)