Barbera Thomas 845538

GTF validator
Documentazione ed elenco delle violazioni considerate

Il codice sorgente del progetto è disponibile al seguente repository GitHub:
https://github.com/citrangoli/GTF-validator

Per poter usare il validatore è necessario eseguire il notebook Jupiter "`GTF validator.ipynb`" presente sul repository, che contiene – in aggiunta al codice – delle sezioni di commento in formato Markdown atte a descrivere le funzionalità e spiegare alcune scelte implementative.

Il notebook è stato scritto e testato usando l'ambiente Python 3.7.6.

Sintassi GTF, validazione del file e assunzioni
La principale fonte per la descrizione di una corretta sintassi GTF è stato il sito https://mblab.wustl.edu/GTF22.html.
Questo validatore controlla la correttezza sintattica di un file GTF considerando le seguenti assunzioni:

- Un file può contenere righe lasciate vuote, in questo caso esse vengono scartate senza costituire un errore.
- Un file può contenere commenti. Essi iniziano con il carattere cancelletto '#' e proseguono fino alla fine della riga. Tutto ciò che si trova dopo un # non viene parsato. Solitamente un commento può occupare una riga intera o essere posto dopo il campo `attributes` di un record.
- Ogni record valido deve contenere esattamente nove campi separati da tabulazioni:
  - **seqname**: può essere composto da qualsiasi carattere (alfanumerico e di punteggiatura) eccetto caratteri di spaziatura.
  - **source**: analogamente al campo precedente, gli unici caratteri non ammessi sono quelli di spaziatura, mentre tutti gli altri sono ritenuti validi.
  - **feature**: ogni file GTF deve necessariamente contenere almeno un record con le feature `CDS`, `start_codon` e `stop_codon`. Se una o più di queste feature risulta mancante, vengono segnalati degli errori nel report finale. Le altre feature ammesse e riconosciute (`5UTR`, `3UTR`, `inter`, `inter_CNS`, `intron_CNS` e `exon`) sono opzionali. Ogni altro tipo di feature viene ignorato, ma è comunque ammesso. In questo caso nel report finale viene indicato un warning al posto di una grave violazione.
  - **start** e **end**: questi due campi possono esclusivamente contenere coordinate rappresentate da valori interi 1-based (0 non è quindi ammesso). Inoltre, il valore start deve essere obbligatoriamente minore o uguale al valore end.
  - **score**: questo campo può assumere diverse forme: un numero intero (eventualmente negativo), un numero decimale (eventualmente negativo) oppure il campo può essere vuoto (indicato dal carattere punto ".").
  - **strand**: questo campo può assumere esclusivamente due valori: "+" e "-".
  - **frame**: questo campo deve essere valutato in funzione della feature. Se la feature è `CDS`, `start_codon` o `stop_codon`, allora il frame deve essere per forza 0, 1 o 2. Per tutte le altre feature il valore del campo frame deve essere lasciato vuoto (".").
  - **attributes**: il campo attributes permette di specificare gli attributi di ogni record con queste condizioni: non c'è un numero massimo di attributi, ma ogni riga

deve necessariamente avere gli attributi `gene_id` e `transcript_id`. Essi devono essere i primi due attributi presenti anche se il loro valore potrebbe essere vuoto. Tutti gli attributi che seguono i primi due devono comunque essere correttamente formattati, ma verranno poi ignorati ai fini del parsing. L'ordine in cui possono apparire `gene_id` e `transcript_id` tra di loro non è rilevante, quindi è necessario verificare entrambe le opzioni. C'è da considerare inoltre la seguente condizione: se la feature del record è "`inter`" o "`inter_CNS`", allora il campo `transcript_id` deve essere lasciato vuoto. In caso contrario l'attributo deve avere un valore associato.

Violazioni del formato GTF
Le violazioni che seguono sono state numerate perché ad ognuna di esse è associato un file nella cartella "Violazioni" nominato con il numero corrispondente. Effettuare la validazione di uno di questi file permette di generare la violazione desiderata.
Ad esempio, se si prova a validare `08.gtf`, si ottiene la violazione "campo end non composto da cifre".

Sono state considerate le seguenti violazioni/errori:

- [00] Il file passato come input non esiste (banale)
- [01] Il file passato come input è vuoto (banale)
- [02] Nel file non è presente nessun record con feature obbligatorire (`CDS`, `start_codon` o `stop_codon`)
- [03] Una riga non ha esattamente nove campi separati da tabulazioni
- Per ogni riga:
  - [04] Il campo seqname contiene caratteri non ammessi (spazi)
  - [05] Il campo source contiene caratteri non ammessi (spazi)
  - [06] Il campo feature non appartiene alla lista di nove feature riconosciute (warning)
  - [07] Il campo start non è composto da cifre
  - [08] Il campo end non è composto da cifre
  - [09] Il campo start è inferiore a uno
  - [10] Il campo end è inferiore a uno
  - [11] Il campo start è maggiore del campo end
  - [12] Il campo score contiene qualcosa di diverso da un numero intero, da un numero decimale e dal punto "."
  - [13] Il campo strand non contiene "+" o "-"
  - [14] Il campo frame per un record con feature `CDS`, `start_codon` o `stop_codon` è diverso dai valori 0, 1, 2
  - [15] Il campo frame per un record con feature differente dalle tre precedenti è diverso dal punto "."
  - [16] Il campo attributes non contiene gli attributi minimi necessari o contiene informazioni mal formattate (per esempio se gli attributi non vengono separati da esattamente uno spazio)