Background Tempus is building the world's largest repository of clinical and molecular data and leveraging those data to improve cancer care. In order to provide the best care possible, we often need to ingest, clean, and structure external data generated outside of Tempus. A major responsibility of a computational biologist at Tempus is to incorporate such external data into our internal knowledge database (KDB) sanely. For this project, you will be cleaning and structuring the data in Clinical Interpretations of Variants in Cancer (CIViC).

Data Sources:

CIViC : Evidence and variant data

Assignment

Instructions: Complete the assignment below to the best of your ability. In general there is no "one correct answer", so make assumptions as you see fit. Please note any assumptions that are made. Complete this assignment programmatically (from download to output) using either R or Python. You may use any R/Python packages you see fit. If you are unable to complete a step, document your approach and roadblocks. Please understand that roadblocks are common and do not disqualify you as a candidate as long as you can formulate questions about how to proceed.

Goal:

Produce a database of missense variants with clinical evidence for therapeutic response from CIViC. This database will contain only variants where the mutation in question exists independently, i.e. it does not occur in tandem with a gene fusion. Additionally, you must clean the data to get an integer value for the exact amino acid position at which the missense variant occurs.

You are to produce a single SQLite output file with four tables: `variant`, `evidence`, `evidence_drug`, and `variant_alias`.

1. Download the "evidence" and "variants" data from CIViC (links above)
2. Exclude all evidence that does not relate to therapies (drugs) and all variants that are not missense variants.
3. Further exclude "variant" entries that are combined missense variant + another type (often fusions) or that are non-specific, i.e. don't relate to a single AA position.
4. Exclude all evidence that does not relate to missense variants, and remove all variants that are not referenced by the remaining evidence.
5. Separate drugs in the Evidence table `drugs` column into a new mapping table with one drug per row. E.g.

evidence_id	drug
35	erlotinib
35	gefitinib

6. Do the same thing with `variant_aliases` in the Variants table, mapping to `variant_id`.
7. Extract the amino acid position from each variant and make it a new field. For example, variant “V600E” will have a new field `aa_position` == 600
8. Output an SQLite DB with four tables and (at least) the following fields:

- variant
    - variant_id
    - gene
    - entrez_id
    - variant
    - aa_position
- evidence
    - evidence_id
    - disease
    - evidence_type
    - evidence_direction
    - clinical_significance
    - evidence_level
    - citation_id
    - variant_id
- evidence_drug
    - evidence_id
    - drug
- variant_alias
    - variant_id
    - alias

Bonus: Add primary and foreign key constraints to the tables as you see fit, and feel free to do that outside of your R or Python code.

The output of this assignment will be:

1) An R or Python script completing the required steps (or Rmarkdown / Jupyter lab notebook)
2) An SQLite db file
3) A document of at most a single page giving a concise overview of your process, as well as any roadblocks or assumptions you made. Please include a rough estimate of the time that you took to complete this assignment. Taking your time and completing the steps correctly is significantly more important than raw speed. Or if using Rmarkdown/Jupyter feel free to explain your process in-line.