Technical Report: Transformation

The given data model designed for this project consists of several classes including ChromosomeSequence, LocationInfo, Variant, Disease, Interpretation, HGVSExpression, and Database. Each class is tailored to hold specific pieces of genomic information:

* ChromosomeSequence: Stores the chromosome and its assembly.
* LocationInfo: Holds the genomic location data, such as position and reference allele.
* Variant: Contains information about variants, such as RS IDs, alternate alleles, and variant types.
* Disease: Captures disease information linked to specific variants.
* Interpretation: Includes clinical significance and review status for variants.
* HGVSExpression: Records the HGVS nomenclature describing the detailed variant structure.
* Database: Holds links to external databases where variant data is cataloged.

Two key relationships were established to demonstrate the many-to-many connections typical in genomic datasets and had to be therefore transformed to a csv as well:

* Located\_In: Links ChromosomeSequence to LocationInfo, indicating where on a chromosome a specific sequence is located. This relationship is crucial for understanding the geographical genomic context of data points.
* For a: Connects Disease and Interpretation, representing the clinical interpretations associated with diseases linked to genomic variants. This many-to-many relationship addresses the scenario where multiple diseases can share common clinical interpretations and vice versa.

Using the PyVCF library, genomic data was extracted from VCF files. The library efficiently parses the complex format, allowing access to structured fields like CHROM, POS, REF, and various INFO tags including CLNDN and CLNSIG.

One of the major challenges was the diverse format of the information in the INFO field. Tags like CLNSIG and CLNDN varied in format and content, necessitating a flexible parsing strategy. Standardization efforts included normalizing these formats to ensure consistent data entry into our model.

The actual mapping of data from the VCF format to our structured model involved creating DataFrame structures in Python with Pandas. Each class was populated with relevant data from the VCF fields. Relationships like "located\_in" were particularly challenging due to the need to link position data with chromosome identifiers reliably.

During the transformation process, issues such as missing data, duplicate entries, and inconsistent formats posed significant challenges.

By standardizing the data format, the transformed data is now ready to be integrated with other genomic datasets. This standardization facilitates interoperability, essential for advanced genomic analyses and research.