CODE DOCUMENTATION and USER INSTRUCTIONS

Jennifer J. Stiens
j.j.stiens@gmail.com
https://github.com/jenjane118/kinesin_db

Instructions for implementation and updating database

1. Mysql database, 'kinesin' is located on 'kenobi' server (kenobi.cryst.bbk.ac.uk) accessed from the 'hope' server (hope.cryst.bbk.ac.uk). For access, use the file <config_kinesin.py>. The database was created using <kinesin.sql>.

2. To populate database download the following files:
   <mutation_parser.py>, <dbinsert_module.py>, <impact_table.py>, <domain_mapper.py>, <median_score.py>, config file (like <config_kinesin.py>), and data files (<mutations.2019-07-13.json>, <results.json>, <V89_38_MUTANT.csv>, <clinvar_result.txt>, <fathmm_results.txt>, <vep_complete_results.txt>)

   Run: <dbinsert_module.py>

3. To update data files (see notes on individual scripts for more info):

   TCGA:
   https://portal.gdc.cancer.gov/explorationfacetTab=genes&searchTableTab=mutations
   Submit 'KIF11' (in 'Genes' tab)
   Select 'JSON' button above chart of somatic mutations to download entire set of mutations.
   Use resulting file with 'mutation_parser.py' with updated filename.
   For TCGA tissue information using TCGA API:
      Select 'Save/Edit Mutation Set' above chart of somatic mutations, in 'manage sets', tick set and 'Export Selected' button.
      Copy text to file: 'tgdc_mutationset.tsv'.
      Use resulting file with 'gdc_api.py'.

   COSMIC:
   https://cancer.sanger.ac.uk/cosmic
   Go to Downloads in drop down menu and go to 'Cosmic Mutation Data'
   Enter 'KIF11' in 'Filter by Gene' box and download file.

   Clinvar:
   https://www.ncbi.nlm.nih.gov/clinvar and search for 'KIF11', filter results for 'variation-gene' relationship as 'single gene'. Download file in upper right corner, select 'tabular'.

   FATHMM cancer:
   http://fathmm.biocompute.org.uk choose 'Cancer' on webpage menu.
   Use 'fathmm_format.py' to get list of mutations in correct format for webserver ('P52732 N342V,E101V') and copy entries into box on webpage (in plain text).
   Set prediction threshold to 1.0 and submit. Download results into text file.

   VEP:
   Run 'vep_parse.vepScores' to get list of mutations in genomic format.
   Submit to https://www.ensembl.org/Multi/Tools/VEP?db=core by pasting list of mutations into box and select the following parameters:
      Use Ensembl/GENCODE transcripts.
      Turn off finding co-located variants in 'Variants and frequency data' and all additional annotations.

In ‘Predictions’ :
enable SIFT and PolyPhen ‘Prediction and Scores’
enable dNSFP and select desired scores:
MutationTaster_score, _converted_rankscore, and _prediction
MutationAssessor_score, _rankscore, and _prediction
FATHMM_score, _converted_rankscore, and _prediction
PROVEAN_score, _converted_rankscore, and _prediction
MetaSVM_score, _rankscore, _prediction
REVEL_score, and _rankscore
REVEL_rankscore
MutPred_score
MutPred_rankscore
CADD_raw
CADD_raw_rankscore
enable ‘Condel’
Run VEP and save downloaded file (‘vep_complete_results.txt’).

Run ‘dbinsert_module.py’ program with updated filenames to parse files and insert any new mutations.

Code Documentation / Module Descriptions

1. **mutation_parser.py**

This module contains all the functions for parsing the various text files from COSMIC and GDC(TCGA).

Functions

** ’gene’ or ‘my_gene’ is ‘KIF11’ for kinesin-5 human gene

parseGDC(gene, json_file)
Parses .json files of mutations from GDC (https://portal.gdc.cancer.gov). Returns list of entries for Mutation, source_info and impact tables.

cosmicParser(my_gene, csv_file)
Parses .csv file from COSMIC (https://cancer.sanger.ac.uk/cosmic). Returns a dictionary with protein change as key for entry into mutation, source_info, and impact tables.

tissueGDC(gene, json_file)
Parses json file of tissue results from GDC api request. Skips entries that are recorded in COSMIC database. Returns ‘tissue_list’ for entry into tissue table.

tissueCosmic(gene, csv_file)
Parses csv files from COSMIC for tissue information. Returns list of attributes for entry into tissue table.

2. **dbinsert_module.py**

This program calls on parsing module functions and inserts attributes to populate tables in kinesin database. Uses parsing functions from mutation_parser.py and impact_table.py. It was necessary to separate each table insert into a separate function to avoid foreign key constraint problems.

Functions

** ‘database’ is either ‘kenobi’ for department server, or ‘home’ for personal computer configuration.

insertMutation(mutations, database)
Inserts entries from GDC list into mutation table. Returns number of executed rows.

insertCosmicMutation(mutation_dict, database)

Inserts entries from COSMIC dictionary into mutation table. Returns number of executed rows.

insertSource(mutations, database)
Inserts entries from GDC into source_info table. Returns number of executed rows.

insertCosmicSource(mutation_dict, database)
Inserts entries from COSMIC mutation dictionary into source_info table. Returns number of executed rows.

insertCosmicTissue(cos_tissue_list, database)
Inserts entries from list of COSMIC tissue attributes into tissue table. Returns number of executed rows.

insertGdcTissue(tissue_list, database)
inserts entry list from GDC into tissue table. Returns number of executed rows.

3. **impact_table.py**

This program parses VEP (Variant Effect Predictor) files (https://www.ensembl.org/Homo_sapiens/Tools/VEP), FATHMM (http://fathmm.biocompute.org.uk) and Clinvar (**https://www.ncbi.nlm.nih.gov/clinvar**) for functional impact results. It includes functions to parse the output files and update mysql database impact table.

Functions

** ‘my_gene’/‘gene’= ‘KIF11’ and ‘database’ = ‘kenobi’

parseVep2(my_gene, vep_file)
Parses VEP flat file results (‘vep_complete_results.txt’) downloaded after using webservice (https://www.ensembl.org/Multi/Tools/VEP?db=core) with file from vep_parse.vepScores function to get all predictions and scores. Returns list of entries to impact table.

updateImpact2(impact_list, database)
Connects to kinesin database. Updates relevant entries in impact table with all metrics from VEP.
First must find mutation_id for each entry based on cds attribute from impact_list. Uses this to update impact table. Returns number of inserted rows.

fathmmResultsParser(csv_file)
Parses FATHMM results file downloaded from FATHMM website (http://fathmm.biocompute.org.uk)
for predictions of the oncogenic status of missense amino-acid substitutions in the KIF11 protein. Returns dictionary of FATHMM results with protein mutation as key.

fathmmInsert(results_dict, database)
Updates mutation entries in impact table with FATHMM cancer results using dictionary from fathmmResultsParser.py. Returns number of inserted rows.

parseClinvar(gene, csv_file)
This function parses Clinvar file (https://www.ncbi.nlm.nih.gov/clinvar) for clinical significance and converts protein change from 3-letter amino acid code to one-letter code. Returns list of entries.

clinvarUpdate(clinvar_list, database)
This function uses list of ClinVar attributes from parseClinvar script to update impact table. Returns number of inserted rows.

4. **median_score.py**

Module to retrieve functional impact scores, calculates median of all ranked scores and CONDEL and insert medians in impact table.

Functions

** ‘database’ = ‘kenobi’

getScores(database)
Queries kinesin database impact table for functional impact scores. Returns dictionary of scores with mutation_id as key.

calcMedian(score_dict)
Calculates median of all ranked functional impact scores. Returns median dictionary with mutation_id as key.

insertMedians(median_dict, database)
Inserts medians in impact table. Returns number of executed rows.

5. **gdc_api.py**

This module parses downloaded mutations file from GDC for sample ids (ssmid) for mutations and also contains function to make API requests with ssmids.

Functions

IdReader(csv_file)
Reads sample ids (ssmid) from flat file of downloaded mutations from GDC. Returns .json file of sample ids.

GdcRequest(file, results_file)
Uses file of sample ids to make requests from GDC webservice (**https://api.gdc.cancer.gov/ssms/**). Returns file of resulting tissue source info in .json format.

6. Misc Scripts:

domain_mapper.py Maps protein mutations to assigned Pfam domain: ‘motor’, ‘coiled-coil/disorder’ or ‘tail-bind’

config.kinesin.py Database configuration for kinesin database access on kenobi server

fathmm_format.py Script to query database and format mutation data into acceptable format for FATHMM webservice

vep_parse.py Module containing vepScores function to use to query database and format mutations in acceptable format for VEP webservice (other functions redundant)

Datafiles: Perhaps list and link to github
scripts R (edriver only)
mutation3D: list of mutations, cluster results files/pymol