Research Notebook

Ashish Jayamohan

12/05/2020

Resources

1. Main Research notes for reference
2. Check *PLINK Docs*
3. *Box* for all sources
4. *Original Email Source*
5. *Genomics Sequencing Info*
6. *https://app.box.com/folder/118113081562*
7. *PLINK v1.9 docs*
8. *https://www.illumina.com/products/by-type/microarray-kits/infinium-omni25-exome-8.html*
9. *https://www.illumina.com/science/technology/microarray.html*
10. *https://www.bayhonors.org/14-sample-data-articles/115-preparing*

Input Standards

1. At `bfile-path`, put in the path as if input to PLINK
2. For example, if the bfile was in the directory `test-dir` and had the common name `test` (with extension names `.bed`, `.bim`, `.fam`), you would input `test-dir/test`
3. `hg19 db` and other static files can be imported/downloaded externally

---

12/12/2020

Plink Documentation

- —file specifies the input file of some function
- —function in the middle

- —out specifies where the output should be exported to
- <> required argument for specific function
- [''] optional arguments that can refine your search
- —bfile will reference the .bim, .bed, and .fam files with the prefix
- All other file systems are converted into bfile before processing
- Main `gemfile` will be in perl

12/14/2020

Initial File Manipulation

- Exclude the lines that have a 0 in the first column
- Rs numbers are better than kgp, so convert all kgp to rs
- Exclude the ones that have 0 in the second to last column
- Exclude the ones that have more than one allele per column
- AWK for command line could help in this case
- Lines should be excluded if they have zeroes in the first column (chromosome), the fourth column (base pair position along the chromosome), and the fifth column (the rarer allele - a zero here means only the column 6 allele was observed in all participants).

Annovar Interface

- Annovar -- make input files like the ones in the example
- Use file to switch from kgp to rs numbers (utilize map)
- Download `avsnp150 hg19 {\[kaiwang@biocluster ~/\]$ annotate\_variation.pl –buildver hg19 –downdb –webfrom annovar avsnp150 humandb/}`
- Dropped means that It matched with a specific subset in the hg19 db

---

PLINK --recode Preperation

- Use both rs columns for processing
- Compile all dropped files of 1,3,4 and then pare to unique lines
- Make avinput format and then keep allele ordering as in dropped files [STEP 2]
- Give avinput file to Beth for step 3 with downloading of hg19 db

---

12/20/2020

Annotation Interpretation

- **Sample Line:** SIFT_score SIFT_pred Polyphen2_HDIV_score Polyphen2_HDIV_pred Polyphen2_HVAR_score Polyphen2_HVAR_pred LRT_score LRT_pred MutationTaster_score MutationTaster_pred MutationAssessor_score MutationAssessor_pred FATHMM_score FATHMM_pred PROVEAN_score PROVEAN_pred VEST3_score CADD_raw CADD_phred DANN_score fathmm-MKL_coding_score fathmm-MKL_coding_pred MetaSVM_score MetaSVM_pred MetaLR_score MetaLR_pred integrated_fitCons_score integrated_confidence_value GERP++_RS phyloP7way_vertebrate phyloP20way_mammalian phastCons7way_vertebrate phastCons20way_mammalian SiPhy_29way_logOdds

---

Annotation Analytics

- Find number of each prediction value
- Find number with each type of score
- Find number without each type of score
  - This number is integral as it proves whether a certain variant is more rare or is relatively common
  - May point to whether a variant is exonic or not
- Graphing histograms of score and rank-score distribution

Annovar Full Recode

- **Sample Full Line:** Chr Start End Ref Alt SIFT_score SIFT_converted_rankscore SIFT_pred LRT_score LRT_converted_rankscore LRT_pred MutationTaster_score MutationTaster_converted_rankscore MutationTaster_pred MutationAssessor_score MutationAssessor_score_rankscore MutationAssessor_pred FATHMM_score FATHMM_converted_rankscore FATHMM_pred PROVEAN_score PROVEAN_converted_rankscore PROVEAN_pred MetaSVM_score MetaSVM_rankscore MetaSVM_pred MetaLR_score MetaLR_rankscore MetaLR_pred M-CAP_score M-CAP_rankscore M-CAP_pred MutPred_score MutPred_rankscore fathmm-MKL_coding_score fathmm-MKL_coding_rankscore fathmm-MKL_coding_pred Eigen_coding_or_noncoding Eigen-raw Eigen-PC-raw GenoCanyon_score GenoCanyon_score_rankscore integrated_fitCons_score integrated_fitCons_score_rankscore integrated_confidence_value GERP++_RS GERP++_RS_rankscore phyloP100way_vertebrate phyloP100way_vertebrate_rankscore phyloP20way_mammalian phyloP20way_mammalian_rankscore

phastCons100way_vertebrate phastCons100way_vertebrate_rankscore phastCons20way_mammalian phastCons20way_mammalian_rankscore SiPhy_29way_logOdds SiPhy_29way_logOdds_rankscore Interpro_domain GTEx_V6p_gene GTEx_V6p_tissue

- For the data sets, if we are given only a select number of scores (or rank scores), so we take the average of the ones we have only or take the average of all values, assuming the ones not given to be zero?

01/03/2021

PLINK --recode

- Extract, recode done in any order - do both at once
- Go to main file .bim and replace kgp values with respective rs ids
- Certain arbitrary values can be dropped
- If total number of lines exceeds limit (10% of all lines), dump core and proceed with full termination
- If core is dumped, print error analysis?

Frequency Analysis

- Ensembl Gene biomart - Gathers which genes are related to which phenotypical data
- Use —freq once by itself
- Use —freq with —counts
- Graph average rank score versus all freq
- Plot histogram with allele1 frequency from output files (column5)
- Make histogram of only variants which are in cholesterol metal
- How many variants appear only once?
    - These variants are likely to be exonic and have a large effect

---

01/13/2021

Host Analytics

- Two hosts in total
- One with freqs below some percent

- One with those above the percent cutoff
- RecodeA table subset with only the rs ids for those in cholesterol
- Other file with two columns: gene id, list of rs ids in gene
- Make subtable to csv with just the nine gene identifiers
- Exon should have overlap

Post-Experiment Analysis

- Find all the ones that were rank score <0.5
- Exonic and the variants included in cholesterol
- Find how many lines were disposed and print out number

01/24/2021

File Structure

- In the `main` folder, there will be a sub-folder for documentation, titled `docs`
  - Docs will be in markdown and can have backlinks to the main `README`
- `main.py` will be hosted in the main folder whilst the other scripts will be held in a `scripts` folder
  - Inside the `scripts` folder will be the following `.py` files
  - `tests.py` for main tests, unit tests, and overall checks. This will have no core checkup statements and will just point error
  - `annovar.py` for main Annovar interface
  - `plink.py` for main PLINK interface

Initial Necessities

- Develop tests for checking Annovar and PLINK location
- After further review, it may not be the brightest idea to have a `scripts` folder
- Need some way to point to the PLINK/Annovar website if the selected directory does not contain the necessary files

---

02/01/2021

Opening Webpages via Python

- Import `webbrowser` package via `import webbrowser` at the file header
- To open specific file, use `webbrowser.open(web_url)`
- Selected url will open in the automated preferred brower

Running Shell Commands From Python

1. Include `import os` at file import heading
2. Store command in final buffer string (`buffer_command`)
3. Execute final command via `os.system(buffer_command)`
4. Cleanup and final termination

02/06/2021

PLINK/Annovar Test

- Using `os.popen`, we can gather the string data from a specific terminal
- We can use `os.popen("ls -la")` to show all files in the given directory
- If `annovar` and `plink` folders are present, then the program is ready to run and the core does not need to be dumped
- **PLINK test has been implemented along with timing**
- **ANNOVAR test has been implemented along with timing**

Real File Structure

- Likely, we will have the `main.py` file which will implement various other `.py` files
- Files included will have to obtain `tests.py` in order to ensure the program can run safely without issues\

Tests.py

- Check for `static` folder
  - How do we redirect?
  - Do we send people to the github?

    - But, if we send people to the github, the repo is private, so it may pose problems for people to download the selected files
- Check for db
    - Should be pretty basic
    - Just check for `hg19.db` file and run specific download using annovar if not found

---

ANNOVAR Packaging Information

- *ANNOVAR Paper*
- Note the usage of segmentation and core dumping requirements
- Sample data that is strapped to the main package can be removed via `rm`
- Do we really want to delete it, or can we just retain the normal factory settings?

02/07/2021

Example Data

- Example data is currently proprietary and cannot be released
- We could use PLINK's base `toy` files, but those don't have the entire `bfile` mapping
- Include snippet of prop. data with no identifying information?

I/O Procedure

- Take in file paths and histogram preferences
- Do we want to include histograms?
- Histograms are hard to decipher, hard to produce, but may provide some visual reinforcement of the efficacy of the product?

02/10/2021

Initial File Formatting

- Following procedures outlines initially, we were able to dispose of the unnecessary lines
- From the CAP dataset, there were a total of 325698 lines disposed, amounting to less than our cutoff

- Sample Output:
  `PLINK Directory Has Been Detected. Routing To Core. Process Completed. Time Elapsed: -0.00054890199999999 seconds. ANNOVAR Directory Has Been Detected. Routing To Core. Process Completed. Time Elapsed: 4.190599999999434e-05 seconds. Static Directory Has Been Detected. Routing To Core. Process Completed. Time Elapsed: 0.0004020830000000114 seconds. Data Path --example_data/test Initial Disposal Completed. 325698 Lines Disposed. python3 main.py 2.60s user 0.23s system 19% cpu 14.178 total`
- The example data has been stored in `example_data` where each file is labeled `test`

Program Timing Structure

- Timing works in short processes (checking for certain directories)
- Consider removing the timing system from higher-level and more difficult tasks as this can also help boost runtime efficiency

---

Graphing Possibilities

- Graphing would be complex and would require human input, thus adding complexity to the final product
- Perhaps add a separate module with graphing capabilities later where the user can select which patterns's fitting seems to be the best
- Would this detract from the true efficacy of the program?

Checkpoint

- Tests for basic program requirements has been completed
- Annovar hg19 db download has been completed
  - There are some nuances that still need to be worked out, but that can be refined later on
- Still need to work on the timing struct because that still hasn't been solidified
- Github is not completely working, so we'll need to find a time when we can push many things
  - Many commits to organize the process

02/12/2021

Thoughts on Ensembl VEP

- Ensembl VEP is web-based, so there would be no need to keep any PLINK or ANNOVAR files on locally
- This would make the product easier to use
- VEP has a file size limit, which most datasets would go over this limit
- We would have to prune each file individually before sending into VEP in order to bypass this limit

---

ANNOVAR DB Specs

- `\[kaiwang@biocluster ~/project/annotate_variation\]$ annotate_variation.pl -build hg19 -downdb genomicSuperDups humandb/```
- `NOTICE: Web-based checking to see whether ANNOVAR new version is available ... Done`
- `NOTICE: Downloading annotation database http://hgdownload.cse.ucsc.edu/goldenPath/hg19/database/genomicSuperDups.txt.gz ... OK`
- `NOTICE: Uncompressing downloaded files`
- `NOTICE: Finished downloading annotation files for hg19 build version, with files saved at the 'humandb' directory`
- `\[kaiwang@biocluster ~/project/annotate\_variation\]$ annotate\_variation.pl -regionanno -build hg19 -out ex1 -dbtype genomicSuperDups example/ex1.avinput humandb/`
- `NOTICE: Reading annotation database humandb/hg19\_genomicSuperDups.txt ... Done with 51599 regions`
- `NOTICE: Finished region-based annotation on 12 genetic variants in ex1.hg19.avinput`
- `NOTICE: Output files were written to ex1.hg19\_genomicSuperDups`
- Focus is on specific chromosome, without deviation or adaptability on program-side

---

Timing Issues

- DB has been downloaded, but it took about 1-2 hours totally
- Timing struct reported a total of about 2 seconds, which is wildly off from the true time
- Potential solve: keep timing struct in there and troubleshoot later

---

02/13/2021

Debugging Write To File

- The `echo` is not working
- We are trying to echo the pruned `.bim` array to another `init` file, but the `echo` is not carrying out completely
- The `touch` preceding the `echo` is working perfectly fine
- Finally, I found that the solution will be to `open` the file from the main program, thus copying the `echo` command locally onto the main program
- The `echo` sector, now with `open` with `write` option, now works flawlessly!!

Allele Permutations

- Using `allele_map`, we can find in $O(1)$ time the location of a particular key in the set
- With this knowledge, we can create a simple map which contains all the alleles and their respective ops
- We can also use the main `init_dispose` function to get rid of the indels in the main fileset
- The above action will take $0(n)$ if done separately
- By doing the above section in addition with the main `init_dispose` function, we are able to achieve the same results in $O(n)$ rather than $O(2n)$

02/14/2021

Color Coding

- Colors have been added to the `fileout` to make interpreting the output just a little easier
- `OKGREEN` is used for successful runs and tests
- `OKCYAN` is used to line data which is necessary to be noted
  - Currently, this is being used for the "verification" part of the program
  - This can be replaced later on if there is some automatic checking for the veracity of the program
- `FAIL` is used for demonstrating a failed run or test
- `WARNING` is used for showing pointers in error statements, ensuring that the user knows about the necessary

---

02/16/2021

Allele Permutations

- We are going to save 4 different arrays, each named `final_list_n`, where n is the specific number, with respect to the total number of lists
- We have to pay attention to 4 different configurations. For example, consider the configuration mentioned in the table below:

Initial Position	A	G	Implementation Status
Configuration 1	T	C	Implemented and Tested
Configuration 2	C	T	Currently In Progress
Configuration 3	G	A	Currently In Progress
Configuration 4	A	G	Currently In Progress

02/17/2021

Allele Permutations

- We are going to save 4 different arrays, each named `final_list_n`, where n is the specific number, with respect to the total number of lists
- We have to pay attention to 4 different configurations. For example, consider the configuration mentioned in the table below:

Initial Position	A	G	Implementation Status	Final Check
Configuration 1	T	C	Implemented and Tested	✅
Configuration 2	C	T	Implemented and Tested	✅
Configuration 3	G	A	Implemented and Tested	✅
Configuration 4	A	G	Implemented and Tested	✅

Permutation Testing

- Configuration 4

    - 1 kgp15717912 0 534247 A G
    - 1 rs9701779 0 565374 G A
    - 1 kgp7727307 0 569624 A G
    - 1 kgp15297216 0 723918 A G
    - 1 rs3094315 0 752566 A G
    - 1 rs3131972 0 752721 G A
    - 1 kgp6703048 0 754063 A C
    - 1 kgp15557302 0 757691 G A
    - 1 kgp12112772 0 759036 A G
- Configuration 1
    - 1 kgp15717912 0 534247 T C
    - 1 rs9701779 0 565374 C T
    - 1 kgp7727307 0 569624 T C
    - 1 kgp15297216 0 723918 T C
    - 1 rs3094315 0 752566 T C
    - 1 rs3131972 0 752721 C T
    - 1 kgp6703048 0 754063 T G
    - 1 kgp15557302 0 757691 C T
    - 1 kgp12112772 0 759036 T C
- Configuration 3
    - 1 kgp15717912 0 534247 G A
    - 1 rs9701779 0 565374 A G
    - 1 kgp7727307 0 569624 G A
    - 1 kgp15297216 0 723918 G A
    - 1 rs3094315 0 752566 G A
    - 1 rs3131972 0 752721 A G
    - 1 kgp6703048 0 754063 C A
    - 1 kgp15557302 0 757691 A G
    - 1 kgp12112772 0 759036 G A
- Configuration 2
    - 1 kgp15717912 0 534247 C T
    - 1 rs9701779 0 565374 T C
    - 1 kgp7727307 0 569624 C T
    - 1 kgp15297216 0 723918 C T
    - 1 rs3094315 0 752566 C T
    - 1 rs3131972 0 752721 T C
    - 1 kgp6703048 0 754063 G T
    - 1 kgp15557302 0 757691 T C
    - 1 kgp12112772 0 759036 C T

02/19/2021

Annovar Annotations

- Annotation command: `annotate\_variation.pl ex1.avinput humandb/ \-filter\-build hg19 \-dbtype avsnp150` (has not been cleaned)
- We need to prepare the `avinput` file from the `bim` file
- A sample `bim` line is shown below
  - `1 kgp15717912 0 534247 A G`
- A sample `avinput` line is shown below
  - `1 948921 948921 T C comments: rs15842`

02/22/2021

Annovar Preparations

- I am still working on making sure that the annotations are functional
- Unfortunately, the `Convert2Annovar` perl gemfile is not working currently
- The `Convert2Annovar` problem has been fixed!
- The db that we were previously using is the avsnp version, one which actually is pruned
- I also updated the downloads section of the fileset to reflect this change

---

02/23/2021

Annovar Annotate

- I have to implement the annovar annotations using the `avsnp150` database instead of the `snp138` database
- Issues with space and storage
  - Each one of the databases take up a huge amount of space (13-14 GB)
  - In total, we are downloading 3 databases with one default database coming pre-installed with ANNOVAR
  - Additionally, when we create maps for each one of the databases, we end up using more space because the maps use more space to facilitate $O(1)$ lookup
  - How do we take care of this problem?

- Delete unnecessary databases

DBNSFP Annotations

- Download the DBNSFP database
  - Create new option for download with color scheme and the `DBNSFP` identifier
  - Analyze space threading
- Annotate using `os` package (`os.system`)
- Ensure that output contains the necessary scores and rankscores

---

Annotation Analysis

- We need to figure out how much is "a lot" relative to the other values in the dataset and also compared with other variants
- The rankscore provides us with how the variant compares against the values in the dataset
- The score provides us with the real value that the variant corresponds to
- Presentation tips
  - Incorporate images (show 5-line sample as it passes through program)
  - Make ideal user and issues with current system very clear
  - Exclude technical terminology

02/27/2021

PLINK recode

- In order to `recode`, we need to format the scores and rankscores into plink `bim` format
- After producing a `bim` file, we can then produce the respective `bed` and `fam` files using PLINK
- We need to prune to those with scores and rrank scores that are less than 0.5
- We also have to implement the selective average methodology
  - We take in the `Sift`, `mutpred`, and all of the other scores and scale it so that none of them are particularly biased

Completion of Project

- After debugging and implementing the `ensGene` database download, we were able to finish the process

- We made the project more "simple" by removing unnecesary functions
- We also consolidated the entire `tests` file into one central function, with implements used to invoke the function
- We also changed up the text coloring to ensure no mistakes are made with coloring text
- Downloads are also consolidated into one function case, with each invokation mentioning the db name