The file **RealDataPipeline\_v2.R** contains the data pipelines in the functions:

* **realDataPipeline\_JST** 
  + This runs our new method and uses the joint testing
* **realDataPipeline\_SKAT**
  + This runs the SKAT method for joint testing
* **realDataPipeline\_GeneScoreTest**
  + This runs the partially penalized score test for testing for effect of gene-based score only

**Changes And/Or Unique Inputs for each function:**

* **Shared between all functions:**
  + score (Can take value of “All”, “AMS”, “Bin MM”, “IBS”, or “Incomp”
    - I added this option in so that you don’t have to run the code for all 4 of the scores if you don’t want to. The default option is “All”, which outputs the scores and p-values for all four models, like before. If you specify a score, you will get an output of 1 line, with the multi-marker score value and the p-value for a model fit using R geno and the specified score.
    - Output for option “All”:

|  |  |  |
| --- | --- | --- |
|  | Multi-marker Score Value | P-value |
| R Geno/IBS Score |  |  |
| R Geno/Incomp Score |  |  |
| R Geno/AMS Score |  |  |
| R Geno/Bin MM Score |  |  |

* + - Output for option “AMS”: (example)

|  |  |  |
| --- | --- | --- |
|  | Multi-marker Score Value | P-value |
| R Geno/AMS Score |  |  |

\*\*\*Output is slightly different for the **realDataPipeline\_GeneScoreTest**, I’ll explain that below

* **realDataPipeline\_JST** 
  + s (number of PCs to be used)
    - I’ve changed this to number of PCs instead of percent of variance explained
    - In simulations, I used between 1-5
      * Most of the time, 1 or 2 PCs had highest power
      * Sometimes when the LD is lower, needed more PCs to get highest power, but power loss for smaller number of PCs wasn’t too bad
* **realDataPipeline\_SKAT**
  + kernel (which kernel to use in the SKAT method)
    - For simulations, I used either linear or IBS kernel
    - IBS tends to have slightly higher power than linear when gene score is associated and much higher power than linear when R genotype SNPs are associated
    - I found that the IBS kernel took much longer to run (over 8 hours for 5K simulations vs about 1 hour for 5K simulations with linear kernel), so maybe try IBS out on a few genes first to see how long it takes
* **realDataPipeline\_GeneScoreTest**
  + fitIntercept (TRUE or FALSE; do you want to include an intercept in the fitted models?)
    - I’ve included an option to either fit an intercept with the model or fit a model without an intercept. Both JST and SKAT have intercepts, so I have been fitting the simulations with the intercept.
  + Unpen (vector of integer values)
    - This option lets you input a vector of values that you do not want to be penalized by the function. The original function code penalizes all coefficients that are not specified in the null hypothesis (including those for covariates and the intercept). I added in this part of the code to force the model to fit unpenalized coefficients for my included covariate values. It is similar to the penalty.factor option in glmnet, where you can set a penalty to 0 in order to force the fit of certain covariates
    - I have set up the default to force fit of intercept (if it is included) and all covariate values. This means that the only coefficients being penalized are the Beta values for the R genotype SNPs. This is what I have been using in the simulations.
  + Output format for Gene Score Test:
    - Along with score value and p-value, I have included the number of non-zero betas in the restricted (under null) and unrestricted models. Sometimes it is interesting to see the difference between how many of the R genotype SNPs were penalized to 0 in each of the models.
    - If the Score value = -1, this means there was some issue with the matrix algebra when calculating the Score Test Stat. In order to avoid the code crashing I had it arbitrarily set the value to -1. Let me know if you run into this a lot, we could potentially switch to fitting the partially penalized LRT instead of the score test. I have included the Score test only for now, but the method also has a LRT and Wald test that can be used.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Multi-marker Score Value | P-value | Number of Non-Zero Betas in the Restricted Model | Number of Non-Zero Betas in the Unrestricted Model |
| R Geno/IBS Score |  |  |  |  |
| R Geno/Incomp Score |  |  |  |  |
| R Geno/AMS Score |  |  |  |  |
| R Geno/Bin MM Score |  |  |  |  |

* Shared Inputs for the functions: (Anything listed here hasn’t changed)
  + Main datafile. **\*\*This has not changed since the first code \*\*** (The same type of file that could be used with the single SNP method)
    - This can be in any file format as long as  the data is formatted like a matrix.
    - Number of rows should equal 2\*number of D/R pairs (ie all Donors and Recips from the pairs)
    - Number of columns should correspond to number of SNPs. Columns should have values between 0-2 for additive model.
    - Rownames should be the Ids of the Donor or Recipient the genotype information corresponds to in order to do D/R matching.
  + ID File. **\*\*This has not changed since the first code \*\***
    - A txt file with two columns.
    - First column is Recipient Id, second is matched Donor ID.
    - This file needs to be in .txt format or it won’t be read in correctly
  + Covariate and Phenotype File. **\*\*This has not changed since the first code \*\***
    - A text file containing additional covariate information, including the phenotype/outcomes.
    - First column should be Recipient ID
    - Second column should be outcomes (Y) – can be either binary or continuous
    - Additional columns can contain other covariates
      * All covariates should be in numeric form (ex: gender should be 0/1 instead of male/female)
  + File path to data **\*\*This has not changed since the first code \*\***
    - The path to where the data is located
  + standardizeScores (TRUE or FALSE) **\*\*This has not changed since the first code \*\***
    - A true or false value of whether the gene-based scores will be standardized. Essentially, this will divide the unweighted sum of single SNP scores by *m* = total number of SNPs (for binary scores – incompatibility and binary mismatch score) or by 2*m* (for IBS and AMS scores – 0,1,2 values) to make all the single SNP scores range from 0-1
    - Based on simulations, the standardization and weighting is not very useful, it is possible weighting could be more helpful for rarer SNPs but I haven’t looked into that
  + weightedScores (TRUE or FALSE) **\*\*This has not changed since the first code \*\***
    - A true or false value of whether the SNPs that are summed into gene-based scores will be weighted.
    - Can be used to upweight SNPs with lower MAF if wanted
    - I haven’t been able to test this with simulations yet either, so not sure if it should be used yet
  + scoreWeights **\*\*This has not changed since the first code \*\***
    - A vector of scores if weightedScores = TRUE
    - Should have one weight for each SNP in the gene region being tested
  + phenosBinary (TRUE or FALSE) **\*\*This has not changed since the first code \*\***
    - TRUE if phenotype of interest is binary
    - FALSE if phenotype of interest is continuous
  + Out File **\*\*This has not changed since the first code \*\***
    - Name of the output file created by the function
    - File contains final score values and p-values for the four combinations of R genotype/score combinations