0.1 Aim One (1a): A data-adaptive association test for longitudinal data analysis within GEE framework

0.1.1 Statistical Modeling

Suppose for each subject $i = 1, \ldots, n$, we have k total longitudinal measurements $y_i = (y_{i1}, y_{i2}, \ldots, y_{ik})'$ with y_{im} as a element, p SNPs of interest as a row vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ with x_{ij} coded as 0,1 or 2 for the count of the minor allele for SNP $j = 1, \ldots, p$, and $z_i = (z_{i1}, z_{i2}, \ldots, z_{iq})$ is a row vector for q variates. We assume common effect sizes of the SNPs and covariates on the longitudinal phenotype/trait measurements. Thus, we construct the design matrix for the SNPs and covariates as:

$$X_i = \begin{pmatrix} x_i \\ x_i \\ \vdots \\ x_i \end{pmatrix}, Z_i = \begin{pmatrix} 1 & z_i \\ 1 & z_i \\ \vdots & \vdots \\ 1 & z_i \end{pmatrix}$$

where x_i and z_i are row vectors of length p and q respectively. X_i is a $k \times p$ matrix, and Z_i is a $k \times (q+1)$ matrix. Denote the regression coefficient $\beta = (\beta_1, \beta_2, \ldots, \beta_p)'$ and $\varphi = (\varphi_1, \varphi_2, \ldots, \varphi_{q+1})'$ for X_i and Z_i respectively. The marginal mean of each measurement, $E(y_{im}|x_i, z_i) = \mu_{im}$ where $m = 1, 2, \ldots, k$ for k total measurements, or the vectorized format of all measurements, $E(y_i|x_i, z_i) = \mu_i$, relates to the SNPs and covariates through a generalized linear model (GLM):

$$g(\mu_i) = \eta_i = Z_i\varphi + X_i\beta = H_i\theta$$

with $H_i = (Z_i, X_i), \theta = (\varphi', \beta')'$ and $g(.)$ as a suitable link function. For continuous outcome, an identity link is usually used.

The consistent and asymptotically Normal estimates of β and φ can be obtained by solving the GEE [?]:

$$U(\varphi,\beta)=\sum_{i=}^{n}U_i(\varphi,\beta)=\sum_{i=1}^{n}(\frac{\partial\mu_i}{\partial\theta'})'V_i^{-1}(Y_i-\mu_i)=0,$$

with

$$\frac{\partial\mu_i}{\partial\theta'}=\frac{\partial g^{-1}(H_i\theta)}{\partial\theta'},V_i=\phi A_i^{\frac{1}{2}}R_wA_i^{\frac{1}{2}},$$

and

$$A_i=\begin{bmatrix} v(\mu_{i1}) & 0 & \cdots & 0 \\ 0 & v(\mu_{i2}) & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & v(\mu_{ik}) \end{bmatrix}$$

ϕ in V_i is the dispersion parameter in GEE and is usually treated as nuisance parameter. $v(\mu_{im})=\phi\text{Var}(y_{im}|x_i,z_i)$. $R_w(\alpha)$ is a working correlation matrix depending on some unknown parameter α. For convenience, a working independence model with $R_w=I$ is often used. With a canonical link function and a working independence model, we have a closed form of the U vector with two parts corresponding to SNPs and covariates, and its covariance estimator:

$$U=(U'_{.1},U'_{.2})'=\sum_i(Z_i,X_i)'(Y_i-\mu_i)$$

$$\widetilde{\Sigma}=\widehat{\text{Cov}}(U)=\sum_i(Z_i,X_i)'\,\widehat{\text{var}(Y_i)}\,(Z_i,X_i)=\sum_i(Z_i,X_i)'(Y_i-\hat{\mu}_i)(Y_i-\hat{\mu}_i)'(Z_i,X_i)=\begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix} \tag{1}$$

where $\hat{\mu}_i$ is an estimator of μ_i, $\widetilde{\Sigma}$ is an estimate of the covariance of score (U) vector. $\widetilde{\Sigma}$ is partitioned with the dimensions according to the score vector component $U_{.1}$ and $U_{.2}$ for φ

and β respectively.

Quantitative traits

We use the identity link, i.e. $g(\mu_{im}) = \mu_{im}$ and $v(\mu_{im}) = \phi \times 1 = \phi$. Then we have:

$$\begin{aligned} U &= \sum_i (Z_i, X_i)' \, R_w^{-1}(Y_i - \mu_i) \\ \widetilde{\Sigma} &= \sum_i (Z_i, X_i)' \, R_w^{-1}(Y_i - \hat{\mu_i})(Y_i - \hat{\mu_i})' R_w^{-1} \, (Z_i, X_i) \end{aligned} \tag{2}$$

if the assumption of a common covariance matrices across Y_i for i is valid, e.g. for quantitative continuous traits study [**?**], we can adopt a more efficient covariance estimator:

$$\widetilde{\Sigma} = \sum_{i=1}^{n} (Z_i, X_i)' \, \widehat{\mathrm{var}(Y_i)} \, (Z_i, X_i) = \sum_{i=1}^{n} (Z_i, X_i)' \left(\sum_{i=1}^{n} \frac{(Y_i - \hat{\mu_i})(Y_i - \hat{\mu_i})'}{n} \right) (Z_i, X_i) = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix}$$

which is used by default for its better finite-sample performance [**?**].

In my dissertation, I will **focus on** the case with quantitative traits, since they are most typical traits used as response variable in longitudinal data analysis. Nevertheless, I introduce the binary traits strategy as below. In general, the only difference lies in which canonical link we will use, with all other equations/formulas keep the same.

Binary traits

For binary traits (trait value coded as 0 and 1), we use the logit link function so that $g(\mu_{im}) = log\frac{\mu_{im}}{1-\mu_{im}}$ and $v(\mu_{im}) = \mu_{im}(1 - \mu_{im})$. Additionally the (m, l)th element of $\frac{\partial \mu_i}{\partial \theta'}$ is $H_{i,ml}\mu_{im}(1 - \mu_{im})$ with $H_{i,ml}$ as the (m, l)th element of H_i, which is the short notation for (Z_i, X_i).

Then we have:

$$\begin{aligned} U &= \sum_{i=1} (\frac{\partial \mu_i}{\partial \theta'})' V_i^{-1}(Y_i - \mu_i) \\ &= \sum_{i=1} (\frac{\partial \mu_i}{\partial \theta'})' \phi A_i^{-\frac{1}{2}} R_w^{-1} A_i^{-\frac{1}{2}} (Y_i - \mu_i) \end{aligned}$$

and

$$\begin{aligned} \widetilde{\Sigma} &= \sum_i \left(\frac{\partial \mu_i}{\partial \theta'}\right)' \phi A_i^{-\frac{1}{2}} R_w^{-1} A_i^{-\frac{1}{2}} (Y_i - \hat{\mu_i})(Y_i - \hat{\mu_i})' \phi A_i^{-\frac{1}{2}} R_w^{-1} A_i^{-\frac{1}{2}} \left(\frac{\partial \mu_i}{\partial \theta'}\right) \\ &= \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix} \end{aligned}$$

Several Current Association Tests

Our goal is to detect whether there is any association between the longitudinal trait and the SNPs via testing on hypothesis $H_o : \beta = (\beta_1, \beta_2, \ldots, \beta_p)' = 0$. We have under the null hypothesis with $g(Y_i) = Z_i \varphi$ to obtain φ and predict $\hat{\mu} = g^{-1}(Z\hat{\varphi})$. We hereby have score vector under the null hypothesis, with a working independence model, is:

$$U(\hat{\varphi}, 0) = (U'_{.1}, U'_{.2})' = \sum_{i=1}^{n} (U'_{i1}, U'_{i2})'$$

where

$$U_{.1} = \sum_i Z_i'(Y_i - \hat{\mu_i}), U_{.2} = \sum_i X_i'(Y_i - \hat{\mu_i})$$

As U asymptotically follows a multivariate normal distribution under H_0, then the score vector for β also has an asymptotic normal distribution:

$$U_{.2} \sim N(0, \Sigma_{.2}),\ \Sigma_{.2} = \widehat{Cov}(U_{.2}) = V_{22} - V_{21} V_{11}^{-1} V_{12}$$

, where V_{xx} are defined in Equation 1.

- **The Wald Test:** The Wald Test known as $T = \hat{\beta}'\text{cov}\ (\hat{\beta})\ \hat{\beta}$ is most commonly used, where $\hat{\beta}$ is the estimate of β after fitting the full GEE model with $g(\mu_i) = Z_i\varphi + X_i\beta$. Under H_0, we have $T \sim \chi^2_p$. The Wald test is more time consuming by fitting full model, may fail to converge with many SNPs put on RHS of the regression-like equation to test, and more importantly, the type I error tends to inflate in such case [**?**, **?**].

- **The Score Test:** $T = U_{.2}'\Sigma_{.2}^{-1}U_{.2}^{-1}$, where $U_{.2}$ and $\Sigma_{.2}$ are discussed above; the statistic is asymptotically equivalent to the Wald test with the same null distribution $T \sim \chi^2_p$. Since we only need to fit the null model with covariates, it is computationally easier and less likely to have numerical convergence problems. More importantly, the score test controls the type I error well [**?**, **?**].

- **The UminP Test:** $T = \underset{j}{max}\frac{U^2_{.2,j}}{\Sigma_{.2,jj}}$ for $j \in 1, 2, \ldots, p$, of jth SNP effect. The $\Sigma_{.2,jj}$ is the jth entry on the diagonal of $\Sigma_{.2}$. With max T, we can get minimal p-value accordingly. An asymptotic multivariate normal distribution numerical integration based method provided a fast way to calculate its p-value [**?**, **?**]; alternatively, a simulation based method relying on the asymptotic normal distribution of the score vector can be used to calculate its p-value [**?**, **?**]. Specifically, we first simulate the score vector $U_{(b)} = (U_{(b).1}, U_{(b).2}, \ldots, U_{(b).p})'$ from its null distribution $U_{(b)} \sim N(0, \Sigma_{.2})$ for $b = 1, 2, \ldots, B$, then calculate a total number of B null statistics: $T^{(b)} = \max_{j=1,\ldots,p}\frac{U^2_{(b).j}}{\Sigma_{.2,jj}}$, and the p-value is calculated as $\sum_{b=1}^{B}\frac{I(T^{(b)} \geq T)+1}{B+1}$.

  With a working independence correlation matrix $R_w = I$, every element $\frac{U^2_{.2,j}}{\Sigma_{.2,jj}}$ is equivalent to running the model on each single SNP (e.g. jth) one by one and get the Score test statistics. Hence, in this condition, the GEE-UminP test is equivalent to the usual UminP test that combines multiple single-SNP based longitudinal association test statistics.

A new class of tests and a data-adaptive test in longitudinal data settings

Before I introduce the proposed new test method, let me explain the logic in current GEE and Score test based methods.

$$T_{Sum} = 1'U = \sum_{j=1}^{p} U_j, \qquad T_{SSU} = U'U = \sum_{j=1}^{p} U_j^2,$$

These two tests are called Sum test and SSU test [?]. The former is closely related to other burden tests such like those in [?, ?, ?] If there is a common association either in direction or strength for causal SNVs with no or few non-associated SNVs, then Sum test and the likes will be most powerful; otherwise, the SSU test and its closely relatives, such as kernel machine regression (KMR or SKAT) [?, ?, ?, ?, ?] and C-alpha test [?], will be most powerful.

Sum test and SSU test are all based on score vector. A more general form of score-based statistic can be generalized as:

$$T_w = W'U = \sum_{j=1}^{p} W_j U_j$$

where $W = (W_1, \ldots, W_p)'$ is a vector of weights for the p SNVs [?]. Different researchers proposed different weighting schemes to pool the information of all SNVs in a region of interest, such as those used in [?, ?, ?, ?, ?, ?, ?, ?]. However, all of these weighting schema used fixed weights, e.g. proportional to the MAF of SNV, proportional to standard deviation of SNV, proportional to regression coefficient, proportional to single SNV p-value, etc, and there is no uniformly best weighting scheme as shown in [?, ?, ?].

As a complement to SNVs weighted average, SNVs selection is preferred in the case that there are many non-associated SNVs among the group of SNVs to be tested. Such methods include aSum+ and aSSU which are based on Neyman-type tests [?]. However, variable selection will also omit those variables with mild to moderate information. In our context, due to extremely low MAF of RVs, even underlying fact is that the individual RV is strongly associated with trait, there is only limited information stored in this single RV. Dumping seemingly non-informative RVs may actually omit the signals within the group of SNVs. Therefore, we expect the model averaging based test will outperform the model selection based test in above settings.

The SPU test

Our goal is to specify a whole class of weights which can cover a wide range of association patterns: for any given data with unknown association pattern, we hope at least one member of the whole class of weights can render a powerful test. We reason that, since association information is largely maintained in the score vector itself as comparable to regression coefficient, score vector is not only the basis in GEE and Score test based methods aforementioned, but also may be an informative and simple weight! Specifically, we propose a class of weights

$$W_j = U_{.2,j}^{\gamma-1}$$

for a series of integer value $\gamma = 1, 2, \ldots, \infty$, leading to the sum of powered score (U) tests called SPU tests:

$$T_{SPU(\gamma)} = \sum_{j=1}^{p} U_{.2,j}^{\gamma-1} U_{.2,j}$$

When $\gamma = 1$, the SPU(1) test uses $\mathbf{1}$ as weight and sums up the information contained in all the SNVs in the region of interest, equivalent to Sum test or burden test; when $\gamma = 2$, the SPU(2) test uses U as weight to itself and is equivalent to SSU test and other variance-component test such as SKAT; when γ keeps increasing, the SPU(γ) test puts higher weights on the jth SNV with larger $|U_{.2,j}|$, while gradually decreasing the weights of other SNVs with smaller $|U_{.2,j}|$. As the large value of $|U_{.2,j}|$ indicates strong association information stored in SNV j and small value of $|U_{.2,j}|$ indicates weak or none association information stored in SNV j, a higher γ tends to put more and more weights on those informative SNVs. When

$\gamma \to \infty$ as an extreme situation, where ∞ is assumed to be an even number, we have

$$T_{SPU(\gamma)} \propto ||U||_\gamma = \left(\sum_{j=1}^{p} |U_{.2,j}|^\gamma\right)^{\frac{1}{\gamma}} \;\to\; ||U||_\infty = \max_{j=1}^{p} |U_{.2,j}| \equiv T_{SPU(\infty)}.$$

which takes only the largest element (in absolute value) of score vector. Apparently, SPU(∞) is equivalent to UminP test except the variance of each score component is replaced by 1 as in the denominator part.

By above explanation, we can see SPU(γ) test can connect to a few current test with a simplified framework though. Treating $U^{\gamma-1}$ as the weight with $\gamma \geq 1$ have at least two advantages:
First, score vector is as informative as a vector of the estimated regression coefficient while being computationally much simpler and more stable in the case of low frequency SNVs. Specifically, since U_j contains association information about SNV j and U_j under null hypothesis follows $N(0, V)$, a larger component of $|U_j|$ corresponds to strong evidence of association between the jth SNV and the trait;
Second, it leads to a simple interpretation and a guidance: as the value of γ increases, we up-weight more and more the larger components of the score vector while gradually ignoring the remaining components. Such process smoothly combines the variable weighting and variable selection schema. Besides, an even integer of γ automatically eliminates the effect of opposite signs of U_j, thus avoid power loss due to opposite direction effects canceling out each other; an odd integer of γ might be more appropriate, as in SPU(1), Sum test or other burden tests, when the SNV effects are all in the same direction.

In our experience, SPU(γ) test with a large $\gamma > 8$ usually gave similar results as that of SPU(∞) test [**?**], thus we will only use $\gamma \in \Gamma = \{1, 2, \ldots, 8, \infty\}$ for the whole dissertation work. Suppose the sample size is large enough or MAF of SNV is large enough for the asymptotic normal distribution of score vector to hold under null hypothesis, we will use a simulation method to calculate the p-value from each $T_{SPU(\gamma)}$ [**?**, **?**]. Specifically, suppose

T is short notation of $T_{SPU(\gamma)}$ for a specific γ and $\hat{\Sigma}_{.2}$ is the covariance matrix of the score vector $U_{.2}$ based on original data (see Equation 1). We draw B samples of the score vector from its null distribution: $U_{.2}^{(b)} \sim MVN\left(0, \hat{\Sigma}_{.2}\right)$, with $b = 1, 2, \ldots, B$, and thus obtain a statistics under null hypothesis: $T^{(b)} = \sum_{j=1}^{p} U_{.2,j}^{(b)\gamma}$. We then can calculate the p-value of $T_{SPU(\gamma)}$ as $P_{SPU(\gamma)} = \sum_{b=1}^{B} \frac{I(T^{(b)} \geq T^{obs})+1}{B+1}$.

The aSPU test

Although we have a list of SPU(γ) statistics and p-values, we are not sure which one is the most powerful in a specific data situation. Thus, it will be convenient to have a test which data-adaptively and automatically select/combine the best SPU(γ) test(s). We hereby propose an adaptive SPU (aSPU) test to achieve such purpose. There are a list of combining methods, such as exponential combine [**?**], linear combine, quadratic combine and fisher's combine methods [**?**, **?**, **?**], however in this dissertation work we will use minimum-p combining method exclusively with room left for trying other combining methods. As for different γ, it is difficult to characterize the power curve of an SPU test in real data situation, we will use the p-value of a SPU test to approximate its power; this idea has been prevalent in practice. Accordingly, we will have the aSPU test statistic:

$$T_{aSPU} = \min_{\gamma \in \Gamma} P_{SPU(\gamma)},$$

where $P_{SPU(\gamma)}$ is the p-value of a specific SPU(γ) test.

Similarly as the above simulation method to get p-value of $T_{SPU(\gamma)}$, the *same strategy* can be applied to get the p-value of T_{aSPU} and actually it fully utilizes the previous simulated intermediate result, hereby saves another *unnecessary* simulation work. Specifically, at the SPU test stage we already have the $U_{.2}^{(b)}$ for $b = 1, 2, \ldots, B$. We then calculate the corresponding

SPU test statistics $T^{(b)}_{SPU(\gamma)}$ and p-value

$$P^{(b)}_{SPU(\gamma)} = \sum_{b_1 \neq b}^{B} \frac{I(T^{(b_1)}_{SPU(\gamma)} \geq T^{(b)}_{SPU(\gamma)}) + 1}{(B-1)+1}$$

for every γ and every b. Then, we will have $T^{(b)}_{aSPU} = \min_{\gamma \in \Gamma} P^{(b)}_{SPU(\gamma)}$, and the final p-value of aSPU test is:

$$P_{aSPU} = \sum_{b=1}^{B} \frac{I(T^{(b)}_{aSPU} \leq T^{obs}_{aSPU}) + 1}{B+1}.$$

It is worth noting again that the same B simulated score vectors have been used in calculating the P_{aSPU}.

In practice for genome wide scan purpose, we can use a "data-adaptive" aSPU test strategy that is: we first start with a smaller B, say $B = 1000$, to scan the genomes, then gradually increase B to say 10^6 for a few groups of SNVs, e.g. specific genes or windows, which pass an pre-determined significance cutoff (e.g. p-value $\leq 5/B$) in the previous step; repeat this process according to user's specific need until satisfying the significance level accuracy, e.g. a p-value of $\leq 10^{-7}$ requires $B \geq 10^7$. In this "data-adaptive" way of implementing the simulation based p-value calculating method for aSPU test, we will be able to apply the aSPU test to GWA data.

Other versions of aSPU test

- **aSPUw test**

  The SPUw test is a *diagonal-variance-weighted* version of the SPU test, defined as:

$$T_{SPUw(\gamma)} \quad = \quad \sum_{j=1}^{p} \left(\frac{U_{.2,j}}{\sqrt{\hat{\Sigma}_{.2,jj}}} \right)^{\gamma}$$

  Accordingly, **the aSPUw test** statistic is defined as

$$T_{aSPUw} \quad = \quad \underset{\gamma \in \Gamma}{min} P_{SPUw(\gamma)}$$

where $P_{SPUw(\gamma)}$ is the p-value from $T_{SPUw(\gamma)}$. The procedures of getting these values are exactly the same as in above **aSPU** test based on simulation. Finally, aSPUw p-value can be get by:

$$P_{aSPUw} = \sum_{b=1}^{B} \frac{I(T_{aSPUw}^{(b)} \le T_{aSPUw}^{obs}) + 1}{B+1},$$

again the same formula as **aSPU** test. It is worth noting that **aSPU** and **aSPUw** test can be implemented once using the same simulated score vector, which makes the computation more efficient.

The **aSPUw** test is designed to complement the performance of aSPU test. As the standard deviations of SNVs in a region may vary a lot, there is possibility that a *non-informative* SNV has *larger* standard deviations than other associated SNVs, and the SPU test statistic will be dominated by the noise coming from the null but with larger standard deviation SNV, thus leads to concealing association signals and eventually reduce the test power. Another advantage **aSPUw** brings about is it makes jointly analyze the effect of RVs and CVs possible by giving them an inverse-standard-deviation weight closely related to MAF.

The **aSPUw** test also has disadvantages, otherwise, we will not keep mentioning **aSPU** test as our flagship test within the aSPU test family (including aSPU, aSPUw, and below aSPU.Score and aSPUw.Score tests). When **variance** of SNVs are quite **homogeneous**, put a variance-based weight (always positive) in the denominator will shrink the test statistics and thus lead to less power. In brief, there will be some scenarios, the aSPU test will dominate aSPUw test, and vice versa. Therefore, it is worth generating both test results for all real-data scenarios of which we don't know the underlying SNV variance situation (homogeneous or heterogeneous). We can compare the results afterwards. The best thing is already mentioned earlier: the two tests can be executed at the same time without extra computation burden.

- **aSPU(w).Score test**

  Although the **GEE Score test** will lose power in some scenario of gene-based GWA analysis as mentioned before, it still has the unique advantage in some scenarios when the correlation structure among SNVs really matters. GEE Score test in the form of $T = U'_{.2}\Sigma_{.2}^{-1}U_{.2}^{-1}$ will keep the covariance matrix in the denominator, which preserves the information of possible linkage disequilibrium among SNVs. To combine the pros of GEE Score test and aSPU(w) test, we propose to adopt the minimum p-value combining strategy again, yielding the aSPU(w).Score test with test statistic:

  $$T_{aSPU.Score} = \min\Big\{\min_{\gamma\in\Gamma} P_{SPU(\gamma)}, P_{Score}\Big\},$$

  where P_{Score} is the p-value of the Score test. To calculate the p-value of the aSPU(w).Score test, it is just as simple as to include the Score test p-value along with the other SPU(γ) p-values, select the minimum p-value among them to form the new statistic $T_{aSPU.Score}$, then use the same simulation algorithm as discussed earlier to get the the $P_{aSPU.Score}$.

  The advantage of **aSPU(w).Score** test is we only need to sacrifice a little bit test performance in all scenarios (based on our extensive simulation studies, which is though not shown here), to exchange for a huge improved stability in maintaining a high power in all scenarios (usually when aSPU family performs not so impressive, and Score test happen to be on the edge due to its retaining of the LD information among SNVs).

0.2 Aim One (1b): Longitudinal aSPU family tests on Rare Variants

0.2.1 Statistical Modeling

In the previous section 0.1.1 we discussed the methodology development of aSPU family tests on common variants with a longitudinal trait. In this section, we will discuss the extension of the new methods to rare variants.

While MAF of RVs are usually low, e.g. between 0.001 to 0.01, the asymptotically Normal distribution of either *beta* coefficient or score vector may or may not hold. The simulation-based p-value calculating method as proposed in CV scenario is not sufficient in RV case and need modification. Specifically, in last section, we have:

$$U_{.2}^{(b)} \sim MVN\left(0, \hat{\Sigma}_{.2}\right)$$

with $b = 1, 2, \ldots, B$, and thus obtain a statistics under null hypothesis: $T^{(b)} = \sum_{j=1}^{p} U_{.2,j}^{(b)\gamma}$. We then calculate the p-value of $T_{SPU(\gamma)}$ as $P_{SPU(\gamma)} = \sum_{b=1}^{B} \frac{I(T^{(b)} \geq T^{obs})+1}{B+1}$.

The above algorithms will hold in RV case by large, except that the $U_{.2}^{(b)}$ may not follow the multivariate Normal distribution any longer. As a remedy, we propose a permutation algorithm that generates the empirical null distribution of $U_{.2}^{(b)}$ and in the same time maintain the relationship between longitudinal traits and possible covariates such as age, gender, etc, for subject i. The algorithm is required to be also robust to missing data as this is a usual case in longitudinal data settings. The permutation algorithm can be implemented as follows:

1. identify the max k across all n subjects, which is the number of longitudinal measurements, e.g. $k = 4$ as used in simulation study in section **??**.

2. detect if the data has missing values, if yes, fill the missing value with NA to complement the data dimension (for example, subject i with $Y_i = (y_{i,1},,,y_{i,4})'$ has two missing measurements at time 2 and time 3. After missing value complementing, it becomes $Y_i = (y_{i,1}, \text{NA}, \text{NA}, y_{i,4})')$. Now we should have all the subjects with each Y_i of dimension equal to $k \times 1$.

3. complement H_i to be of full dimension, i.e. $k \times (p + q + 1)$, for covariates and SNVs. Now we should have $\begin{pmatrix} Y_i & H_i \end{pmatrix}$ as an augmented matrix of dimension $k \times (p + q + 2)$ for each subject i, where $H_i = (Z_i, X_i)$. For total n subjects, we have row-wise binded

matrix

$$M = \begin{pmatrix} Y_1 & H_1 \\ Y_2 & H_2 \\ \vdots & \vdots \\ Y_n & H_n \end{pmatrix}$$

of dimension $nk \times (p+q+2)$.

4. permute the SNV chunk among different individuals, i.e. the X_i in $\begin{pmatrix} Y_i & Z_i, X_i \end{pmatrix}$ with the X_j in $\begin{pmatrix} Y_j & Z_j, X_j \end{pmatrix}$, where $i \neq j$.

5. with permuted

$$M^{*(b)} = \begin{pmatrix} Y_1 & Z_1, X_1^{*(b)} \\ Y_2 & Z_1, X_2^{*(b)} \\ \vdots & \vdots \\ Y_n & Z_1, X_n^{*(b)} \end{pmatrix}$$

   we refit the GEE model and get the $U_{.2}^{*(b)}$

6. repeat step 4 - 5 B times to produce $U_{.2}^{*(b)}$ with $b = 1, 2, \ldots, B$.

After we get enough $U_{.2}^{*(b)}$ to form an empirical null distribution, the left work of aSPU test for RVs will be exactly the same as we did on CVs. The only difference is, previously we get simulation based null distribution of score vector under CVs situation, but now we rely on special permutation algorithm in the longitudinal data settings to generate the null distribution of score vector.