Detecting and Dissecting A-C Covariance Using Polygenic Risk Scores in Twins and Parents

Alex Pugner

Department of Behavioural and Movement Sciences, Vrije Universiteit Amsterdam

P_MINTERN_1: Internship 1

Prof. Conor V. Dolan

20 June 2023

Abstract

In the literature many models have been proposed to account for cov(AC) in twin models, whether it comes from cultural transmission or sibling interaction (see. Keller et. al., 2009; Eaves, 1976). Incorporating polygenic risk scores has been shown to be a useful method to account for the covariance between additive genetic variables and shared environmental variables (see Dolan et. al., 2021; Selzam et. al., 2019; Kong et. Al., 2018; Bates et. Al., 2018 and Okbay et. Al., 2022). However, the major shortcoming of these models is that they cannot account for the source of cov(AC), i.e., cultural transmission or sibling interaction. Therefore, the present aim is to propose a parent-sibling model using polygenic risk scores that combine the models proposed in the literature to distinguish these sources of cov(AC) and to investigate the power to determine whether cov(AC) stems from sibling-interaction, cultural transmission, or both sources. Based on the simulated data, we fitted Generalized Estimation Equation regression models and tested their power to reject the null hypothesis in various scenarios. The tests were conducted in a dataset consisting of DZ family data and in in a dataset consisting of MZ and DZ family data. Our results suggest that the model detects cov(AC) with a power that ranges from 0.7 to 1 in most scenarios, where cov(AC) is present, the resolution of the model is considered high. However, one major shortcoming of the model is that if both sources are present (cultural transmission and sibling-interaction), its power is strongly dependent on the value of the cov(AC) effects. Further finding is that the power is not dependent on the sizes of the A, C, and E variance components. However further research with larger computational power is needed further elaborate these findings.

Keywords: A-C covariance, Twins, Polygenic Risk Scores, Cultural Transmission, Sibling-interaction

Introduction

The inheritance of alleles from parents to offspring determines the offspring genotypes. Considering heritable phenotypes, which are phenotypes that are subjects to genetic effects, we can expect that the parents and their offspring will resemble each other, moreover, there will be certain resemblance among the offspring in the family. However, shared environmental influences may also contribute to the phenotypic resemblance. One possible source of the shared environmental influence is that parents and offspring interact with each other. This gives rise to the hypothesis that the shared environmental influences are partially stemming from the parental and the offspring's behaviour. However, derived from the fact that parental behaviour is subject to genetic influences, it is plausible that the shared environmental influences provided by parental behaviour is influenced by parental genetic influences. It has been shown that not just the rearing behaviour of the parents (Klahr and Burt, 2014), but almost all human psychological traits, are at least partly heritable (Polderman et al, 2015). This potential contribution of heritable behaviour to the shared family environment gives rise to covariance between genetic factors and shared environmental factors. If we assume that the genetic influences, denoted as A, are additive, and there are shared environmental influences, denoted as C, we may consider the covariance between A and C denoted as cov(AC). There are two sources of cov(AC) stemming from the interaction among the family members. The first is called *cultural transmission* (Keller et al, 2009, Fulker, 1982), which denotes the shared effects of parental behaviour in the offsprings' phenotype. In other words, *cultural transmission* is the extent the parental behaviour contributes to the shared environment. The second possible source is *sibling-interaction*, which denotes the phenomenon that the heritable behaviour of the offspring contributes to the shared environment of the offspring (Eaves, et al, 1976; Carey, 1986). In this case, the

behaviour the offspring contributes to the shared environment by the mechanism that sibling 1's behaviour provides part of the environment of sibling 2. Therefore, the behaviour of sibling 1 induces a change in the behaviour of sibling 2, but this change is echoed back to the behaviour of sibling 1 (Carey, 1986).

These two different processes (sibling interaction and cultural transmission) can result in covariance between the additive genetic variable (A) and the shared environmental variable (C), which makes the AC covariance a sensible hypothesis in twin modelling. However, since AC covariance is known to be absorbed by the C variance in the classical twin design (Fulker, 1982), the introduction of specific models or extending the classical ACE model to detect AC covariance is crucial. Given the phenotypic data collected in monozygotic (MZ) and dizygotic (DZ) twins and their parents, several models have been proposed to model and estimate cov(AC). Fulker (1982, see also Keller et. al.,2009) proposed the Nuclear Family Twin Design (referred to as NFTD), which includes cultural transmission (described above) and genetic transmission (allele transmission from parent to offspring) in the model. This model is based on parental, monozygotic (MZ) and dizygotic (DZ) twin phenotypic data. With path tracing from the shared environmental variable to the offspring genotypes, we can derive the following path: Parental genetic factors have influence on the heritable parental behaviour, which contributes to the shared environmental factors. However, since the parental genetic factors also have influence on the offspring genetic factors, therefore the parental additive genetic factors are connected to the shared environment and the offspring phenotype. Given this path, we can estimate cov(AC) in the NFTD. These pathways are assumed to be equivalent in both twins. A further extension of this is the Stealth model, where the NFTD is extended to include the spouses of the twins, leading to greater power of parameter estimates. However, a shortcoming of this model is that it assumes AC covariance

stemming from cultural transmission, and does not consider other sources, such as sibling interaction (Keller et. al., 2009).

A model exists for the detection of AC covariance stemming from sibling-interaction, which was introduced by Eaves (1976). The model is based on the phenotypic data of the twins, and it considers the between-sibling interactions, namely the direct effect between the phenotype of twin 1 and twin 2, represented by a path form the phenotype of twin 1 to the phenotype of twin 2, and vice versa. His results show that the MZ total variance is larger than the DZ total variance, which is larger than the total variance of adopted siblings (Foster-siblings). Foster-siblings are offspring, who are genetically not related, but reared in the same environment. This model, however, will detect sibling-interaction, but also cultural transmission, which makes the differentiation between the two sources in the model impossible. This model was developed to handle longitudinal twin data (see Carey, 1986; Dolan et. al., 2014). However, a shortcoming of these models is that they assume sibling interaction, while the AC covariance may be due to cultural transmission.

With the rise of Genome-Wide Association Studies (GWAS), new perspectives opened in the usage of measured genetic information for studying AC covariance, especially the utilisation of polygenic risk scores (PRS). The PRS is based on genetic variants, mainly single nucleotide polymorphisms (SNPs), that are associated with a given phenotype. The SNPs can be viewed as diallelic genetic variants that are additively coded as 0 (aa), 1 (Aa or aA) or 2 (AA), where a and A denote the inherited alleles from each parent. The regression of the phenotype on the SNP gives the association, which can be subject to statistical testing. Given the set of associated SNPs and their regression coefficients from the afore-mentioned regression method one can calculate the PRS score, which is the sum of the SNPs weighted by their regression coefficients. This score can be interpreted as part of the additive genetic factor in the twin model.

The availability of genetic information, in the form of associated SNPs and the derived PRS, gave rise to new ways to estimate and test cov(AC). Dolan et.al. (2021) implemented an ACE model, where the classical ACE twin design is extended with the polygenic risk score of twins to estimate the AC covariance. The power analysis of the model showed that the power to reject the hypothesis of cov(AC) = 0 is partly dependent on the proportion of the PRS contribution to the full phenotypic variance. Although, this model assumes sibling interaction, but cannot rule out a role of cultural transmission.

A possible solution for incorporating PRS to detect cov(AC) stemming exclusively from cultural transmission is using the transmitted / non-transmitted allele design (discussed further as T/NT) proposed by Kong et.al. (2018) and Bates et. al. (2018). This design is based on the parental alleles that are non-transmitted to the offspring, and the assumption that these can have influence on the cognitive abilities of the child. The effect of the non-transmitted alleles or PRS cannot be due to genetic transmission. The effect is interpreted as a consequence of heritable parental behaviour contributing to the offspring environment, i.e., cultural transmission or genetic nurture, as it is discussed by Kong et. al. (2018). These results are consistent with the findings of Bates et. al. (2018) for educational attainment and parental SES moderation of offspring's educational attainment. In this design, the regression of the offspring phenotype on the PRS, which is based on the transmitted and non-transmitted alleles from the parents, provides the test for the AC covariance. However, the inclusion of non-transmitted alleles in the model can be avoided. Okbay et. al. (2022) demonstrated that the regression of the phenotype of the offspring onto the parental PRS and the offspring PRS gives an equivalent test of cov(AC) as the T/NT method. It is plausible, since, if cov(AC) is zero, the correlation between the parental PRS and the offspring phenotype is zero (conditional on the offspring PRS), since the parental PRS is mediated by the parental phenotype. This method has two advantages, that makes it a more favourable choice to the

T/NT design. The parental transmitted and non-transmitted alleles do not have to be determined, which makes it a labour-saving method. Moreover, it is capable of detecting AC covariance even if there is only genetic data from one parent.

A possible method to detect cov(AC) stemming from sibling interaction using the PRS has also been introduced by Selzam et al. (2019). In the paper, they used a mixed-linear model, where variables are the centred mean of the twin PRS (Within-family effect), the average PRS in the family (between-family effect) and residual random effects. In the model, the average family PRS regression coefficient is the test of cov(AC). This model works sufficiently in the case of cognitive traits such as educational attainment or IQ as Selzam et. al. (2019) concluded in their paper, taking up the cov(AC) that is usually absorbed as the C variance in the classical twin studies.

The PRS-based regression methods mentioned above (Kong, et al, Okbay et. al. and Selzam et. al.) can be used to detect cov(AC), but do not differentiate between the source of this AC covariance. Although these are useful methods, no such models have been developed so far in the literature that made the distinction between the various sources (cultural transmission, sibling interaction) possible. The present aim is to introduce such a model, where a PRS-based method is used that combines the methods proposed by Okbay et. al. (2022) and Selzam et. al. (2019). This method requires PRSs in parents and twins, and phenotypes in the twins. Below, we first present the model, and then investigate the resolution of the model to resolve cov(AC) stemming from cultural transmission and from sibling interaction. We investigate the resolution in a simulation study, in which we simulate data according to a given process, where cultural transmission, sibling interaction, or both are present, and fit the regression models of Obkay et. al. (2022), Selzam et. al. (2019) and the combined model.

Methods

To investigate the power of the parent-sibling model to resolve cov(AC) stemming from sibling interaction and/or cultural transmission, we used simulated data. We simulated data according to the path model shown in Figure 1. We denote the effect of cultural transmission as parameter g. We can model sibling-interaction in two ways. First, we can introduce a genotype-phenotype effect in the twins, where the genotype of twin 1 influences the phenotype of twin 2, and vice versa. We denote this as the level process. The effect of sibling-interaction in the level process is denoted as parameter b. Second, we can introduce the sibling interaction effect as phenotypic interaction between the twins (as in Eaves, 1976), i.e., direct mutual effects between the twins' phenotypes. We denote this effect as parameter x. Additive genetic effects, shared and unshared environmental effects are modelled using parameters a, c, and e, respectively (see Figure 1). Even though there are two sources of sibling-interaction, we set either b or x to zero in every parameter setting, since allowing the presence of the two effects at the same time would inflate the power to detect cov(AC) stemming from sibling interaction. The correlation between cov(AC) is dependent on the values of g and b or x, but also on the a and c effects, therefore cov(AC) can be interpreted as the function of g, b or x, a, and c.

Figure 1 illustrates the proposed model, where Am and Af denotes the additive genetic variance of the parents, A_{T1} and A_{T2} denotes the additive genetic value of the twins, C denotes the shared environmental factor of the twins, and ζ_{AT2} and ζ_{AT1} denotes the contribution of PRS score to the A of the twins. Ph_{T1} and Ph_{T2} denotes the offspring phenotypes and e_{T1} and e_{T2} denotes the variance that is not explained by A, C and cov(AC). Given the Mendelian inheritance, we expect that each parent transmits 1/2 of their alleles to the offspring. The path a coefficient from A_{T1} and A_{T2} to Ph_{T1} and Ph_{T2} denotes the additive genetic contribution to the phenotypic variance, while the path coefficient from C to Ph_{T1} and

Ph_{T2} denotes the shared environmental contribution to the phenotypic variance. Path coefficient g denotes the cultural transmission effect, while path coefficients b and x denote the sibling interaction effect, as stated above. Given T1 and T2, g, b and x are set to be equal in both twins in the family.

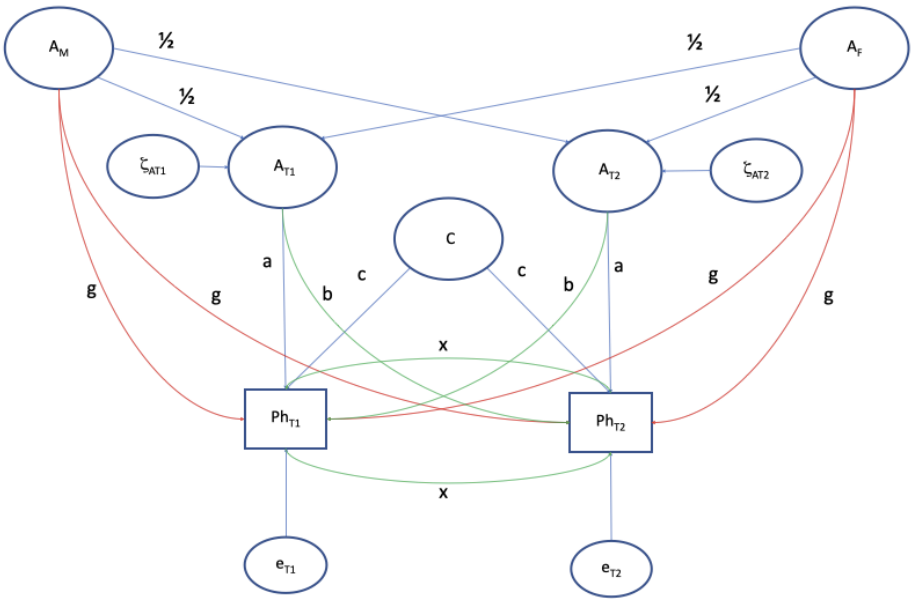

Figure 1. *Path diagram of the ACE parent-sibling model. The sources of cov(AC) are the parameter g (interpreted as the cultural transmission parameters), and b and x (interpreted as sibling interaction parameters*

We preset values by creating vector S, where S = {0, 0.025, 0.05}, and this is a vector with values assigned to parameter g^2, x^2, and b^2. To obtain all variations of the parameter values, we calculated the different variations in S with repetition. As S has n elements, and the number of variations over S are n^k, where n = 3 and k = 3, therefore the number of parameter settings is 27. However, since parameter settings, where $b \neq 0$ and $x \neq 0$ at the same time are mathematically possible, but not desirable for the present aim, we discarded these. As there

are two possible non-zero values (0.025, 0.05), this leads to four different cases where neither b^2 nor x^2 is zero. If we multiply this number with the possible values of g^2, which is 3, we see that we dropped 12 parameter cases. This results in 15 sets of parameter settings for b^2, x^2, and g^2, that we base the data simulation on. The given b^2, g^2, and x^2 are presented in Table 1B.

Although using the variation with repetition method with a^2, c^2, and e^2 values would have been informative, it would lead to 405 different parameter settings which, even with 500 repetitions per case, would take too long to simulate. Therefore, we chose 3 sets of values for a^2, c^2, and e^2 and paired them with all the parameter cases, resulting in 3*15 = 45 parameter settings for the model. These a^2, c^2, and e^2 values are presented in Table 1A.

Set	a^2	c^2	e^2
1	0,340	0,140	0,530
2	0,450	0,310	0,240
3	0,250	0,400	0,350

Table 1A. a^2, c^2 and e^2 parameter settings in the simulated dataset.

ID	g^2	b^2	x^2
1	0,000	0,000	0,000
2	0,000	0,000	0,025
3	0,000	0,000	0,050
4	0,000	0,025	0,000
5	0,000	0,050	0,000
6	0,025	0,000	0,000
7	0,025	0,000	0,025
8	0,025	0,000	0,050
9	0,025	0,025	0,000
10	0,025	0,050	0,000
11	0,050	0,000	0,000
12	0,050	0,000	0,025
13	0,050	0,000	0,050
14	0,050	0,025	0,000
15	0,050	0,050	0,000

Table 1B. The g^2, b^2 and x^2 parameter settings in the simulated dataset.

Based on the parameter settings, we simulated the data using R Studio (RStudio Team, 2020). We set the number of MZ twins and DZ twin pairs both to 2000. The number of diallelic loci is set to 50, while the number of loci comprising polygenic risk score (PRS) is set to 10. Derived from these values, the percentage of genetic variance explained by PRS is calculated as the number of loci comprising of PRS scores divided by the number of diallelic loci. This sets the genetic variance explained by the PRS to 0.2, meaning that it explains 20 % of the additive genetic variance, while 80% of it is explained by the 40 other loci. Given the afore-

mentioned settings, we simulated the genotypic data of the parents. Both the major allele frequency denoted as q and the minor allele frequency denoted as p are set to 0.5. Therefore, the variance of genetic variation per SNP is calculated by 2*p*q, which results in 0.5. The total number of families is 4000 based on the sum of the number of DZ and MZ twin pairs per parameter setting. The parental alleles are simulated based on p and q probability values, each allele coded as 0 and 1 based on the 50 unlinked loci, assuming no assortative mating. Given that offspring inherit either allele 0 or allele 1 from the father and mother, we simulate the allele inheritance based Mendelian inheritance. Based on the parental simulated alleles and the inherited alleles in the offspring, total additive genetic score can be calculated for the family members by summing the inherited allele values in the offspring, or in the case of the parents the simulated allele values. From the results, we can calculate the PRS score. The individual PRS score is calculated by taking the SNPs that have effect on the trait, multiplying them with the effect size of the given SNP, and sum these weighted values. The effect sizes are usually obtained from a previously performed GWAS study (Collister, Liu and Lei, 2022). Based on the aforementioned method, we calculated the PRS scores in the offspring, given that the effect-sizes were set to 1. We subsequently standardized the PRS (mean zero, standard deviation one). We simulated the C (shared environment) and E (non-shared environment), so these are normally distributed. Based on this, we can calculate the phenotypic score with the following equations in DZ twins:

$$ph1_{DZb} = a * A1 + c * C1 + e * E1 + g * Am + g * Af + b * A2$$

(Equation 1)

$$ph2_{DZb} = a * A2 + c * C2 + e * E2 + g * Am + g * Af + b * A1$$

(Equation 2)

where $ph1_{DZb}$ and $ph2_{DZb}$ denote the phenotypic score of DZ 1 and DZ 2, given that the sibling-interaction effect is stemming from the level process, with parameter b. Parameter a, c, e, g, and b are the parameters in the model, as explained above. A1, A2, C1, C1, E1, E2 are the normally distributed variables of additive genetic variables (A1, A2), shared environmental variables (C1, C2) and non-shared environmental variables (E1, E2) in the twins. Am denotes the additive genetic variable of the mother, Af denotes the additive genetic variable of the father. The terms b*A2 and b*A1 denote the contribution of the sibling interaction stemming from twin level process to the phenotypic value. In the case that the sibling interaction is modelled as mutual phenotypic interaction, with parameter b = 0 and parameter x estimated, we set b to zero in Equations 1 and 2 and introduce the sibling interaction as follows:.

$$ph1_{DZ} = ph1_{DZb} + x * ph2_{DZb}$$

(Equation 3)

$$ph2_{DZ} = ph2_{DZb} + x * ph1_{DZb}$$

(Equation 4)

where x * $ph2_{DZb}$ is the contribution to the phenotypic value of the sibling-interaction effect stemming from phenotype interaction in twin 1, and x * $ph1_{DZb}$ is the same contribution to the phenotypic score in twin 2 in DZ families. However, if parameter x is not zero, parameter

b is fixed to zero (and vice versa). Based on the above-mentioned equations we can write the following equations for the MZ twins to calculate the phenotypic score:

$$ph1_{MZb} = a * A1 + c * C1 + e * E1 + g * Am + g * Af + b * A1$$

(Equation 5)

$$ph2_{MZb} = a * A1 + c * C2 + e * E2 + g * Am + g * Af + b * A1$$

(Equation 6)

$$ph1_{MZ} = ph1_{MZb} + x * ph2_{MZb}$$

(Equation 7)

$$ph2_{MZ} = ph2_{MZb} + x * ph1_{MZb}$$

(Equation 8)

Given the simulated data, we fit regression models to detect cov(AC) (see below). The regression equation is based on the calculated PRS scores, which are observed additive genetic variables that account for part of the total additive genetic variance (i.e., the variance due to the total additive genetic variables A1 and A2). However, in fitting the regression models, we must consider the clustering of the data due to shared environmental and genetic effects. To correct the standard errors, we used Generalised Estimating Equation (GEE) regression models (Minica et. al., 2015). In applying GEE regression, we opted for the independence working correlation matrix, to correct the standard errors for the clustering (an

alternative is the exchangeable working correlation matrix, but the comparison of these options is beyond the present scope). The regression equation is

$$ph_{ij} = \beta_0 + \beta_1 * pgsmf_{ij} + \beta_2 * (pgst_{ij} - mpgst_{ij}) + \beta_3 * pgst_{ij} + \varepsilon_{ij}$$

(Equation 9; Model 3)

where *i* = {1,2} corresponds to the individual twin in family *j*. The ph_{ij} denotes the phenotype of the i-th twin in the j-th family, $pgst_{ij}$ is the PRS score of the given twin, pgsmf is sum of the PRS score of the mother and the father, ($pgst_{ij}$ - $mpgst_{ij}$) is the centred mean polygenic risk scores in twins, where mpgst is the mean PRS of the twins in the family and ε_{ij} represents the residuals. This model is the data generating model, which includes multiple sub-models as particular cases, to account for the dissimilar sources of cov(AC). In this model (Equation 9), β_0 is the intercept, β_1 provides the test of cov(AC) stemming from cultural transmission, and β_2 provides the test of cov(AC) stemming from sibling interaction regardless of it is induced by the b or the x effect, β_3 is the PRS effect on the phenotype. We name this Model 3, denoted as M3. In case that β_1 and β_2 are both zero, and β_3 is non-zero, the model is reduced to the following sub-model,

$$ph_{ij} = \beta_0 + \beta_1 * pgst_{ij} + \varepsilon_{ij}$$

(Equation 10; Model 0)

where *i* = {1,2} corresponds to the individual twin in family *j*. The variable ph_{ij} denotes the phenotype of the i-th twin in the j-th family, β_0 is the intercept, β_1 is the regression

coefficient, $pgst_{ij}$ is the polygenic risk score of the i-th twin in the j-th family and ε_{ij} is the residual. We call this model the 0-th model, denoted as M0. Model 0 provides the effect size of the PRS as a predictor of the phenotype.

In the case of the β_2 being zero, and all the other (β_1 and β_3) estimators are non-zero in M3, it is reduced to the following sub-model

$$ph_{ij} = \beta_0 + \beta_1 * (pgst_{ij} - mpgst_{ij}) + \beta_2 * pgst_{ij} + \varepsilon_{ij}$$

(Equation 11; Model 1)

where *i* = {1,2} corresponds to the individual twin in family *j*. The ph_{ij} denotes the phenotype of the i-th twin in the j-th family, β_0 is the intercept, β_1 and β_2 are the regression coefficients, $pgst_{ij}$ denotes the PRS of the twin, ($pgst_{ij}$ - $mpgst_{ij}$) denotes the centred mean of polygenic risk score value in the family, where mpgst is the mean PRS of the twins in the family and ε_{ij} is the residual. $\beta_1 = 0$ provides the test of cov(AC) stemming from sibling interaction based on the method proposed by Selzam et. al. (2009), while β_1 is the regression coefficient of centred mean of the twin PRS and β_2 is the regression coefficient of the PRS of the given twin. We call this Model 1, denoted as M1.

In the case β_2 and β_3 are non-zero in the data generating model, we can fit the following real model

$$ph_{ij} = \beta_0 + \beta_1 * pgsmf_{ij} + \beta_2 * pgst_{ij} + \varepsilon_{ij}$$

(Equation 12; Model 2)

where $i = \{1,2\}$ corresponds to the individual twin in family j. The ph_{ij} denotes the phenotype of the i-th twin in the j-th family, β_0 is the intercept, β_1 and β_2 are the regression coefficients, ε_{ij} is the residual, pgsmf is the sum of the maternal and paternal PRS score in twins in a family, and $pgst_{ij}$ is the polygenic risk score of twin i in the j-th family. The test of $\beta_1 = 0$ is the test of cov(AC) stemming from cultural transmission, while β_1 is the regression coefficient of the sum of the PRS of the mother and father and β_2 is the regression coefficient of the PRS of the given twin.

The design includes 45 (15 *3) cells, each representing a configuration of parameters. We simulated 500 replications for each cell. This resulted in 22,500 p-values, estimates, standard errors, and effect sizes, on which the power calculation was performed.

The power of the test is investigated by simply calculating the proportion of times the given p-value is smaller than our alpha. This gives a power to reject the H-null, which is that there is no covariance between AC, when AC covariance is present. The effect size is also included in the results.

The power is calculated with the following equation

$$\frac{\left(\sum_{i=1}^{I}[p_{ij} < 0.05]\right)}{I}$$

(Equation 13)

where i denotes the i-th repetition of the given set, j denotes the j-th parameter setting. The variable p_{ij} denotes the p-value of the model at the i-th repetition of the j-th parameter setting

and I is the number of repetitions per one parameter setting. The [] bracket denotes the Iverson bracket or, in other words, the indicator function, which returns 1 if the condition is met inside the bracket and 0 otherwise. The summation operator sums up the values of expression inside the brackets for all values of I, and this sum is divided by the alpha gives the power of the models at each parameter value. We set the alpha value to 0.05. In the full model, Model 3, the power to reject the hypothesis that a parameter is zero will equal the alpha if the parameter is truly zero.

To investigate the resolution of the model, we investigate the power of the different sub-models and the data generating model in all the datasets with the given parameter values. We identify 4 different cases which we base the power calculation on.

Case 1: If b, g, and x are all zero, the true model is M0. We expect that the power of the model to detect cov(AC) is equal or close to equal to our alpha, since in M3 $\beta_1 = 0$ and $\beta_2 = 0$.

Case 2: If b or x is present in the model, the true model is M1. Therefore, in that case, we calculate the power to reject hypotheses $\beta_1 = 0$ and $\beta_2 = 0$, where β_1 is the test of cov(AC) stemming from sibling interaction. However, in this setting it is important to fit M2 and investigate the power to reject hypotheses $\beta_1 = 0$ and $\beta_2 = 0$ in the model, since in this model β_1 is the test of cov(AC) but stemming from cultural transmission. Therefore, if we fit M1, we detect cov(AC) if it is present, but it <u>cannot</u> be decided, based on this single equation, whether it comes from sibling interaction or cultural transmission. Fitting M3 is also important since we expect $\beta_1 = 0$ and $\beta_2 \neq 0$. These results would mean that we detected cov(AC) and it is stemming exclusively from sibling interaction.

Case 3: $\beta_2 = 0$, $\beta_1 \neq 0$ and $\beta_3 \neq 0$ in M3, the true model is M2, which is the model testing cov(AC) stemming exclusively from cultural transmission. Therefore, we ask for the power to reject the hypothesis $\beta_1 = 0$. However, we also ask for the power in M1 to detect $\beta_1 \neq 0$ and

the power in M3 to detect $\beta_2 \neq 0$. This gives the test of cov(AC) stemming exclusively from cultural transmission.

Case 4: $\beta_1 \neq 0$, $\beta_2 \neq 0$ and $\beta_3 \neq 0$. In this case we fit M1 and M2 to demonstrate the power to detect cov(AC) in the two models regardless of the source, since in this case cov(AC) is stemming from both cultural transmission and sibling interaction. Moreover, we fit M3 to show the power of detecting $\beta_1 = 0$, $\beta_2 = 0$ and $\beta_1 = \beta_2 = 0$.

The power calculation results will show the true resolution of the model based on the for cases to detect cov(AC) stemming from cultural transmission and sibling-interaction, and its ability to differentiate between the sources from which the AC covariance is stemming from. We fit the above-mentioned GEE regression models in DZ families and both MZ and DZ families. Moreover, we investigate how the a, c and e effects affect the power of cov(AC) detection in the model.

Results

The results from Table 2 A, B and C represent the power values for each parameter setting for each model in the DZ data and DZ & MZ data. We assigned an ID to every parameter setting, therefore it is more convenient to refer to it through the case analysis, than referring to the cells in each table. We also named the 3 different sets of a, c and e parameters as set 1 ($a^2 = 0.45$, $c^2 = .0,31$, $e^2 = 0.24$ in Table 2A), set 2 ($a^2 = 0.25$, $c^2 = 0.40$, $e^2 = 0.35$ in Table 2B) and set 3 ($a^2 = 0.34$, $c^2 = 0.14$, $e^2 = 0.53$ in Table 2C). Moreover, since the tables are complicated to read because they contain many variables and data points, we visualized how the power values change with the parameter settings of a^2, c^2, e^2 and g^2, b^2 and x^2 (Figure 2-4), therefore tendencies in the change of the parameter values are easier to see visually. Given the results we can analyse the power based on the 4 cases given above (Methods section) considering how changes in a^2, c^2 and e^2 effects play a role in the power of the given models, how the changes of the g^2, b^2 and x^2 parameters affect the power of the tests in the models.

ID	a^2	c^2	e^2	g^2	b^2	x^2	M0 dz	M1 dz	M2 dz	M3 dz	M0 mz & dz	M1 mz & dz	M2 mz & dz	M3 mz & dz
1				0,000	0,000	0,000	0,052	0,046	0,056	0,038	0,040	0,046	0,044	0,042
2				0,000	0,000	0,025	0,220	0,868	0,036	0,536	0,122	0,976	0,044	0,862
3				0,000	0,000	0,050	0,354	0,994	0,078	0,792	0,162	1,000	0,048	0,996
4				0,000	0,025	0,000	0,494	0,998	0,062	0,880	0,228	1,000	0,060	0,986
5				0,000	0,050	0,000	0,760	1,000	0,068	0,992	0,374	1,000	0,050	1,000
6				0,025	0,000	0,000	0,972	0,852	0,784	0,050	1,000	0,898	0,996	0,044
7	0,45	0,31	0,24	0,025	0,000	0,025	0,998	1,000	0,774	0,448	1,000	1,000	0,994	0,818
8				0,025	0,000	0,050	0,998	1,000	0,776	0,732	1,000	1,000	0,986	0,980
9				0,025	0,025	0,000	1,000	1,000	0,714	0,784	1,000	1,000	0,968	0,984
10				0,025	0,050	0,000	0,996	1,000	0,656	0,970	1,000	1,000	0,968	1,000
11				0,050	0,000	0,000	1,000	0,986	0,950	0,052	1,000	0,992	1,000	0,060
12				0,050	0,000	0,025	1,000	1,000	0,938	0,416	1,000	1,000	1,000	0,758
13				0,050	0,000	0,050	1,000	1,000	0,938	0,652	1,000	1,000	1,000	0,948
14				0,050	0,025	0,000	1,000	1,000	0,912	0,724	1,000	1,000	0,998	0,980
15				0,050	0,050	0,000	1,000	1,000	0,872	0,972	1,000	1,000	0,998	1,000

Table 2A. The results of the power calculation in set 1. ID = identification number of the given parameter setting and power results for reference, a^2 = squared a effect parameter, c^2 = squared c effect parameter, e^2 = squared e effect parameter, g^2 = squared g effect parameter, b^2 = squared b effect parameter, x^2 = squared x effect parameter, M0 dz = Model 0 power results in DZ data , M1 dz = Model 1 power results in DZ data, M2 dz = Model 2 power results in DZ data, M3 dz = Model 3 power results in DZ data, M0 mz & dz = Model 0 power results in MZ-DZ data, M1 mz & dz = Model 1 power results in MZ-DZ data, M2 mz & dz = Model 2 power results in MZ-DZ data, M3 mz & dz = Model 3 power results in MZ-DZ data. For the description of the models, see Methods.

ID	a^2	c^2	e^2	g^2	b^2	x^2	M0 dz	M1 dz	M2 dz	M3 dz	M0 mz & dz	M1 mz & dz	M2 mz & dz	M3 mz & dz
16				0,000	0,000	0,000	0,044	0,036	0,040	0,030	0,058	0,048	0,058	0,054
17				0,000	0,000	0,025	0,134	0,616	0,040	0,322	0,096	0,802	0,050	0,580
18				0,000	0,000	0,050	0,234	0,886	0,060	0,506	0,132	0,990	0,054	0,886
19				0,000	0,025	0,000	0,540	0,998	0,040	0,862	0,258	1,000	0,062	0,996
20				0,000	0,050	0,000	0,766	1,000	0,056	0,988	0,412	1,000	0,046	1,000
21				0,025	0,000	0,000	0,962	0,834	0,816	0,048	0,996	0,868	0,994	0,054
22				0,025	0,000	0,025	1,000	1,000	0,786	0,304	1,000	1,000	1,000	0,546
23	0,25	0,40	0,35	0,025	0,000	0,050	0,998	1,000	0,794	0,434	1,000	1,000	0,980	0,840
24				0,025	0,025	0,000	0,998	1,000	0,722	0,774	1,000	1,000	0,980	0,984
25				0,025	0,050	0,000	1,000	1,000	0,714	0,974	1,000	1,000	0,972	1,000
26				0,050	0,000	0,000	1,000	0,972	0,958	0,052	1,000	0,986	1,000	0,066
27				0,050	0,000	0,025	1,000	1,000	0,954	0,280	1,000	1,000	1,000	0,552
28				0,050	0,000	0,050	1,000	1,000	0,956	0,390	1,000	1,000	1,000	0,804
29				0,050	0,025	0,000	1,000	1,000	0,916	0,762	1,000	1,000	1,000	0,986
30				0,050	0,050	0,000	1,000	1,000	0,914	0,960	1,000	1,000	1,000	1,000

Table 2B. The results of the power calculation in set 2. ID = identification number of the given parameter setting and power results for reference, a^2 = squared a effect parameter, c^2 = squared c effect parameter, e^2 = squared e effect parameter, g^2 = squared g effect parameter, b^2 = squared b effect parameter, x^2 = squared x effect parameter, M0 dz = Model 0 power results in DZ data , M1 dz = Model 1 power results in DZ data, M2 dz = Model 2 power results in DZ data, M3 dz = Model 3 power results in DZ data, M0 mz & dz = Model 0 power results in MZ-DZ data, M1 mz & dz = Model 1 power results in MZ-DZ data, M2 mz & dz = Model 2 power results in MZ-DZ data, M3 mz & dz = Model 3 power results in MZ-DZ data. For the description of the models, see Methods.

	a^2	c^2	e^2	g^2	b^2	x^2	M0 dz	M1 dz	M2 dz	M3 dz	M0 mz & dz	M1 mz & dz	M2 mz & dz	M3 mz & dz
31				0,000	0,000	0,000	0,040	0,048	0,030	0,050	0,064	0,058	0,062	0,060
32				0,000	0,000	0,025	0,190	0,714	0,054	0,402	0,122	0,858	0,068	0,726
33				0,000	0,000	0,050	0,394	0,976	0,044	0,698	0,156	0,990	0,044	0,952
34				0,000	0,025	0,000	0,522	0,990	0,050	0,870	0,244	0,998	0,052	0,980
35				0,000	0,050	0,000	0,808	1,000	0,046	0,980	0,440	1,000	0,042	1,000
36				0,025	0,000	0,000	0,974	0,788	0,820	0,054	1,000	0,760	0,992	0,048
37				0,025	0,000	0,025	1,000	1,000	0,852	0,338	1,000	1,000	0,996	0,624
38	0,34	0,14	0,53	0,025	0,000	0,050	1,000	1,000	0,834	0,602	1,000	1,000	0,994	0,912
39				0,025	0,025	0,000	0,998	1,000	0,740	0,816	0,996	1,000	0,982	0,980
40				0,025	0,050	0,000	1,000	1,000	0,722	0,980	1,000	1,000	0,966	1,000
41				0,050	0,000	0,000	1,000	0,954	0,968	0,050	1,000	0,968	1,000	0,042
42				0,050	0,000	0,025	1,000	1,000	0,972	0,358	1,000	1,000	1,000	0,632
43				0,050	0,000	0,050	1,000	1,000	0,972	0,558	1,000	1,000	1,000	0,886
44				0,050	0,025	0,000	1,000	1,000	0,950	0,758	1,000	1,000	0,998	0,970
45				0,050	0,050	0,000	1,000	1,000	0,918	0,964	1,000	1,000	1,000	1,000

Table 2C. The results of the power calculation in set 3. ID = identification number of the given parameter setting and power results, a^2 = squared a effect parameter, c^2 = squared c effect parameter, e^2 = squared e effect parameter, g^2 = squared g effect parameter, b^2 = squared b effect parameter, x^2 = squared x effect parameter, M0 dz = Model 0 power results in DZ data , M1 dz = Model 1 power results in DZ data, M2 dz = Model 2 power results in DZ data, M3 dz = Model 3 power results in DZ data, M0 mz & dz = Model 0 power results in MZ-DZ data, M1 mz & dz = Model 1 power results in MZ-DZ data, M2 mz & dz = Model 2 power results in MZ-DZ data, M3 mz & dz = Model 3 power results in MZ-DZ data. For the description of the models, see Methods.

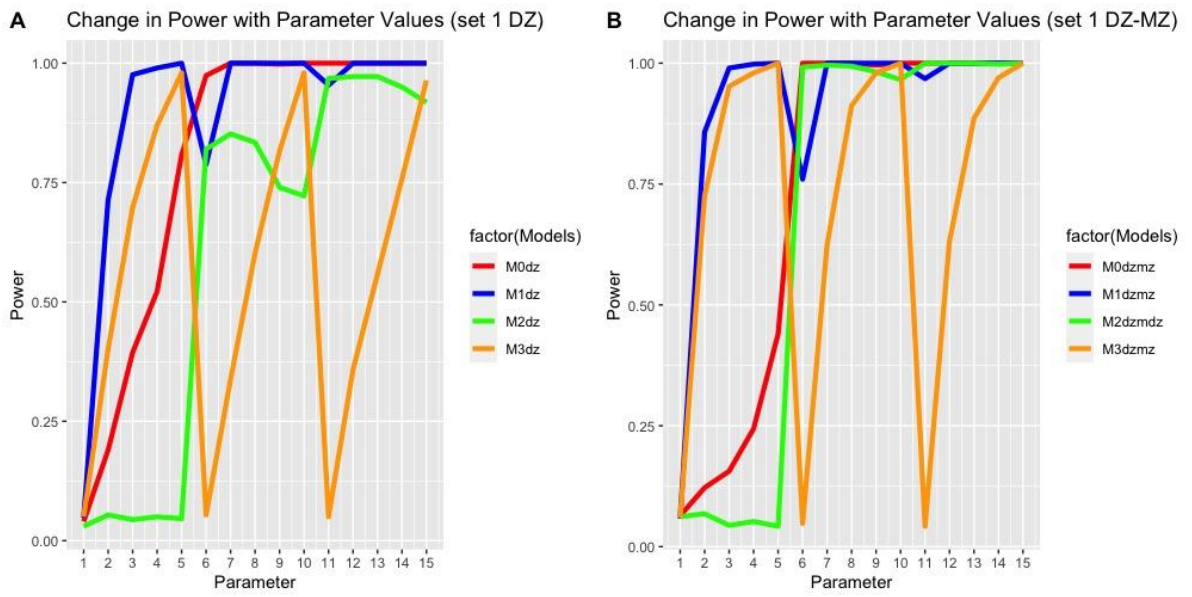

Figure 2. A: *Change in power with the change in parameter values in set 1 (a^2 = 0.45, c^2 =.0,31, e^2 = 0.24) in DZ data. On the x-axis, there are the parameter IDs denote the parameter values from Table 2A. On the y-axes the power values of the given test in the given models, ranging from 0 to 1.00. The factors are the models in the line graph, where M0dz = Model 0 power results in DZ dataset, M1dz = Model 1 power results in DZ dataset, M2dz = Model 2 power results in DZ dataset, M3dz = Model 3 power results in DZ dataset.* **B**: *Change in power with the change in parameter values in set 1 (a^2 = 0.45, c^2 =.0,31, e^2 = 0.24) in MZ-DZ dataset. On the x-axis, there are the parameter IDs denote the parameter values from Table 2A. On the y-axes the power values of the given test in the given models ranging, from 0 to 1.00. The factors are the models in the line graph, where M0dzmz = Model 0 power results in MZ-DZ dataset, M1dzmz = Model 1 power results in MZ-DZ dataset, M2dzmz = Model 2 power results in MZ-DZ dataset, M3dzmz = Model 3 power results in MZ-DZ dataset.*

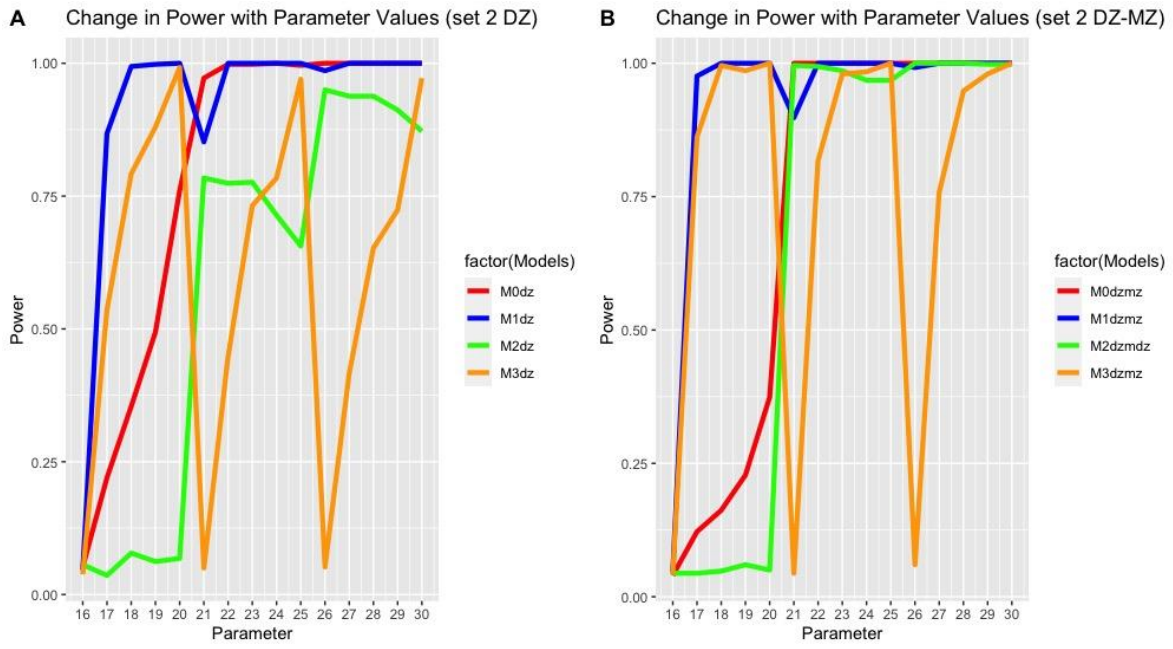

Figure 3. A: *Change in power with the change in parameter values in set 2 (a^2 = 0.25, c^2 = 0.40, e^2 = 0.35) in DZ data. On the x-axis, the parameter IDs denote the parameter values from Table 2B. On the y-axes the power values of the given test in the given models, ranging from 0 to 1.00. The factors are the models in the line graph, where M0dz = Model 0 power results in DZ dataset, M1dz = Model 1 power results in DZ dataset, M2dz = Model 2 power results in DZ dataset, M3dz = Model 3 power results in DZ dataset.* **B**: *Change in power with the change in parameter values in set 2 (a^2 = 0.25, c^2 = 0.40, e^2 = 0.35 in Table 2B) in MZ-DZ dataset. On the x-axis, there are the parameter IDs denote the parameter values from Table 2B. On the y-axes the power values of the given test in the given models ranging, from 0 to 1.00. The factors are the models in the line graph, where M0dzmz = Model 0 power results in MZ-DZ dataset, M1dzmz = Model 1 power results in MZ-DZ dataset, M2dzmz = Model 2 power results in MZ-DZ dataset, M3dzmz = Model 3 power results in MZ-DZ dataset.*

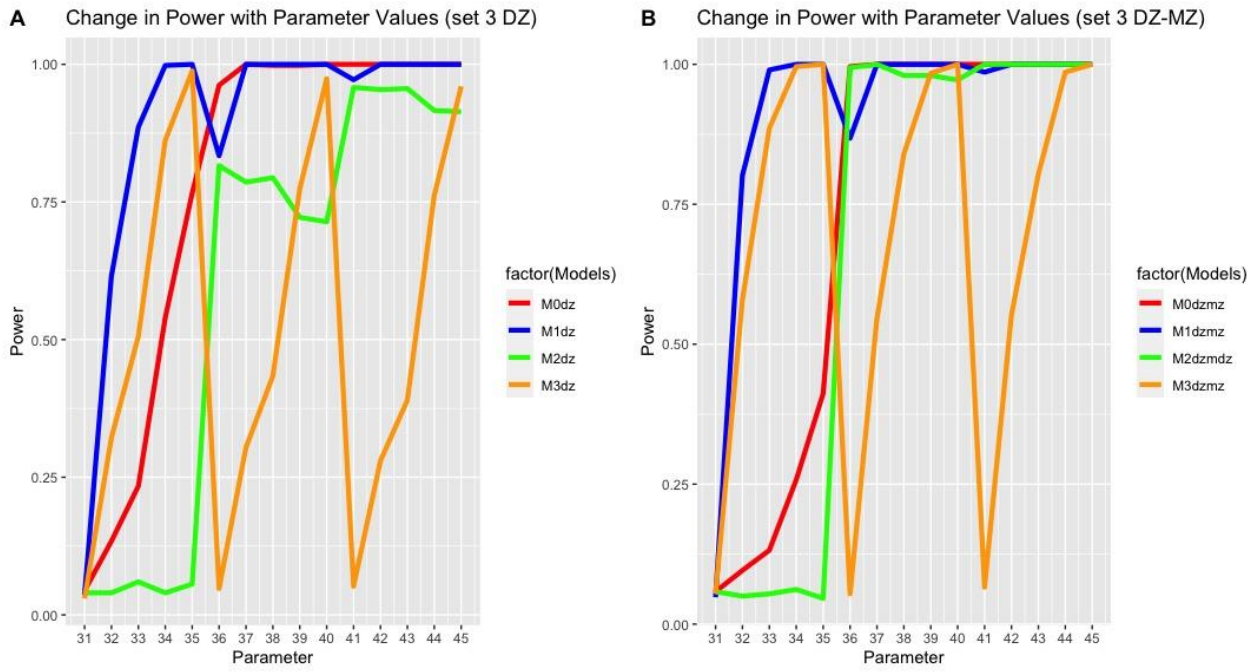

Figure 4. **A:** *Change in power with the change in parameter values in* set 3 ($a^2 = 0.34$, $c^2 = 0.14$, $e^2 = 0.53$) *in DZ data. On the x-axis, there are the parameter IDs from Table 2C, denoting the parameter values, On the x-axis, there are the parameter IDs denote the parameter values from Table 2C. On the y-axes the power values of the given test in the given models ranging from 0 to 1.00. The factors are the models in the line graph, where M0dz = Model 0 power results in DZ dataset, M1dz = Model 1 power results in DZ dataset, M2dz = Model 2 power results in DZ dataset, M3dz = Model 3 power results in DZ dataset.* **B**: *Change in power with the change in parameter values in* set 3 ($a^2 = 0.34$, $c^2 = 0.14$, $e^2 = 0.53$) *in MZ-DZ dataset. On the x-axis, there are the parameter IDs denote the parameter values from Table 2C. On the y-axes the power values of the given test in the given models ranging from 0 to 1.00. The factors are the models in the line graph, where M0dzmz = Model 0 power results in MZ-DZ dataset, M1dzmz = Model 1 power results in MZ-DZ dataset, M2dzmz = Model 2 power results in MZ-DZ dataset, M3dzmz = Model 3 power results in MZ-DZ dataset.*
For the description of the models, see Methods.

Case 1 results

In case 1, there is no effect that contributes to the cov(AC), therefore the power to reject the null hypothesis in the DZ tests and the MZ & DZ tests is expected to equal alpha, which was set before (see Methods). Therefore, cells 1, 16 and 31 in Table 2A, 2B and 2C show the results. We can see that both in the analysis of MZ & DZ data, and in the analysis of DZ data, the power of the test to detect A-C covariance using M0 is close to the alpha (row 1. M0dz = 0.052, M0mzdz= 0.040). As expected, variation in the a, c and e effects has no influence since in row 16 the power to reject the null hypothesis is around .05 (M0dz=0.044, M0mzdz= 0.058) and we can also deduct this from row 31 (M0dz=0.040, M0mzdz= 0.064). This simply shows that the power to detect a parameter giving rise to cov(AC) is close to the alpha of .05, given that the parameter is zero. That is, in the case of fitting all the other models (M1, M2, M3) in case 1 both in the DZ & MZ data and the DZ data, we can conclude that the power to reject the null hypothesis, given that there is no effect, is around the alpha. It is as assumed, since if no g and b or x effect is present, we know there is no cov(AC). Moreover, we can also see that in both DZ data and MZ & DZ data, regardless of the a, c, and e settings, fitting M0 produces low power of rejecting the null-hypothesis if only the b or the x effect is present in the model, however the power reaches 1 in every cell when cultural transmission (g) is not zero, even though the model is not designed to detect covariance between A-C, since it is simply a regression of the phenotype on the PRS.

Case 2 results

Given Table 2A, we see that cells where b^2 is present are ID 4,5,9,10,14,15 and x^2 is present in ID 2, 3 ,7,8,12,13. We see given b^2 is present, regardless if g^2 is present or not, that the power to reject the null-hypothesis is around 1, when fitting M1. This shows that this model detects cov(AC) given that sibling-interaction effect is present in the model. However, we can see that in ID 6, in Table 2A where only cultural transmission is present, power to reject

the null, therefore detecting cov(AC) is 0,852 and in ID 11 in the same table the power is 0,982. We can derive the same conclusions from Table 2B (ID 21 and ID 26) and 2C (ID 36 and 41) with only slight difference in the magnitude of power. We can derive from these results that if the cultural transmission effect is large enough, M1 will detect cov(AC) with almost equal power to the sibling-interaction. This is aligned with the assumption, that it also detects cov(AC) stemming from cultural transmission. However, as outlined in the methods, The power to reject $\beta_1 = 0$ and $\beta_2 = 0$ in M2 is also investigated, since it is the test of cov(AC) stemming from cultural transmission. In the case of ID 2,3 in Table 2A where only x effect is present with no g effect, moreover in the case of ID 4 and 5 in Table 2A, where only b effect is present with no g effect, the power to reject the null-hypothesis given we fit M2 in DZ twin data is around the alpha ranging from 0,04 to 0,06, which shows that the cov(AC) is stemming from sibling-interaction exclusively. We can deduct the same conclusions based on the corresponding rows in Table 2B and 2C. However, in cases where both effects are present (ID 7,8,9,10 and 12,13,14,15) the power is ranging from .97 to 1 when fitting M2 (see Table 2A). This means that M1 detects cov(AC) stemming from either sources, and M2 detects cov(AC) stemming from cultural transmission, therefore we can deduct that both sources contribute to the covariance between A and C. However, we can see in the case of ID 6 and ID 11 in Table 2A that, even-though there is only g present, the power to reject the null-hypothesises given that we fit M1 and M2 is close to 1 (ID 6(M1dz = 0.900, M2dz = 0.996), ID 11(M1dz=0.992, M2dz=1), when truly there is no sibling-interaction effect. The cause of the minuscule difference between the two power values is due to the settings that in ID 6 g^2 is set to 0.025, while in ID 11 $g^2 = 0.05$. However, to account for this, we can fit M3 as described in the Methods section above. We can see that both in the case of ID 6 (power = 0.44) and ID 11 (power=0.60), the power to reject the null-hypothesis is about equal to the alpha, while in the other cases, where cultural transmission and sibling-interaction are

present, or when exclusively sibling interaction is present, the power is much higher. These results hold both in the analysis of the DZ data and the MZ & DZ data, with slightly larger power in the case of MZ-DZ data, compared to DZ data (see ID 2,3 in Table 2A). However, we can also conclude that, while the power of rejecting the null hypothesis while fitting M1 and M2 is not dependent on the magnitude of either the g, b, or x effect, the power to reject the null hypothesis while fitting M3 is dependent on the magnitude of the parameters denoting sibling-interaction. We can see from ID 12 and ID 13, that with the change of the effect from ID 12 ($x^2 = 0.025$) to ID 13 ($x^2 =0.05$), the power goes from 0.758 to 0.948. However, we see this phenomenon only when x is present in the model. In the case, when the b effect is present as sibling-interaction, we cannot observe such difference in power between the two b parameters, namely $b^2 = 0.025$ and $b^2 = 0.05$. Given the three parameter settings for a, c and e effects, we can derive from the table that the size of these effects does not have a major influence on the power of detecting cov(AC) stemming from cultural transmission, since from both Table 2B and 2C we can derive the same conclusions as from Table 2A, given that the power values are just slightly different from the values given in Table 2A. We can also derive this conclusion if we investigate Figure 2, 3 and 4.

Case 3 results

As we stated in the methods section, we fit M2 as the true model in case 3. We can see in both the analysis of the DZ data and in the analysis of the MZ & DZ data from ID 1 to 5 (but also from ID 16 to 20 and ID 31 to 35) that when g is set to zero, the power of M2 is equal to the alpha value, regardless if the b or x effect is present. Unsurprisingly, where g is present and its value is $g^2 = 0.025$, the power of the test while fitting Model 2 is between .7 and .85. However, as the g^2 effect increases from 0.025 to 0.05 regardless of the parameters of a, c and e effects, the power also increases to between .9 and .96. However, we can also derive that, given the parameter settings for a, c and e in Table 2A and Table 2C, in the case of

lower a and c effects (a^2 = .34, c^2 =.14) the power of rejecting the null-hypothesis in M2 is slightly higher if .05 g^2 effect is present (power ranging from 0.918 to 0.972), compared to the setting where g^2=.05, but with larger a and c effects (a^2 =.45, c^2 = .35), where the power is lower (power is ranging between .87 to .95). However, this difference is not exceptionally large, since the power of the tests stay above 0.8. As we can read from Figure 2,3 and 4, the power of rejecting the null hypothesis given that we fit M2 is higher if we fit the model in the MZ & DZ data compared to fitting just in the DZ data. As we have seen it in the results of Case 2, if only g is present with b and x set to 0, the power to reject the null-hypothesis in case of fitting M3 is around the alpha value, therefore we can conclude based on these two power results that in these cases the covariance stemming exclusively from cultural transmission.

Case 4 results

In case 4, we tested the power of detecting cov(AC) in the given models if both g and b or x effect is present as described in the Methods section above. In Table 2A ID 7,8 and 12,13,14,15 present the cases where both effects are present. The power of the test given that if M1 is fitted is ranging from 0.98 to 1, which has already been discussed in the results sections of case 2. The power of this test is not dependent on either the magnitude of a,c and e parameter values or the magnitude of the g or b or x parameter values. In contrast, the power of rejecting the null hypothesis when fitting M2 is dependent on the size of the g parameter. When g^2 = 0.025, the power is around .8, while when g^2 = 0.05, it reaches .9. However, we consider the power of the test large enough, taking into consideration that 0.8 is a widely used and accepted power threshold. We can also note that in all a, c and e parameter settings given that both cultural transmission and sibling interaction is present, the power of the null-hypothesis rejection in M2 increases in the MZ-DZ dataset compared to the DZ dataset. The largest fluctuation in the power is shown by fitting M3 in Case 4. We can see

that M3 has a larger detection power if x is non-zero in the parameter compared to b is non-zero in the parameter. The afore-mentioned phenomena exists independently from the a, c, and e parameter settings. Moreover, we also note that in the MZ-DZ data the power of rejecting the null-hypothesis in M3 in the case of both sources of cov(AC) are present is higher compared to the power in the DZ data set. M3 is also sensitive to the magnitude of g effect regardless the a, c, and e parameter settings. The afore-mentioned results can also be read from Figure 2-4. We can derive from these figures that the values of a, c and e effects do not have major influence on the detection power and how the power increase between the two datasets (only DZ vs DZ & MZ). However, we also tested how the effect sizes (R^2s) change with the different parameter settings, because if the change in magnitude of the effect size is large between two parameter settings, that would mean that the results can be inflated by the fluctuation of the effect size. However, we can deduct from Table 3 that the effect sizes both in the DZ dataset and in MZ-DZ dataset fluctuate between 0.05 and 0.1, depending on the g, b, or x effect magnitudes. This shows that no abrupt magnitude change in the effect sizes occurs, therefore the power results are plausible.

ID	a^2	c^2	e^2	g^2	b^2	x^2	M0 mz & dz	M0 dz
1	0,45	0,31	0,24	0,000	0,000	0,000	0,067	0,067
2				0,000	0,000	0,025	0,073	0,069
3				0,000	0,000	0,050	0,075	0,071
4				0,000	0,025	0,000	0,084	0,078
5				0,000	0,050	0,000	0,090	0,082
6				0,025	0,000	0,000	0,088	0,089
7				0,025	0,000	0,025	0,095	0,091
8				0,025	0,000	0,050	0,097	0,092
9				0,025	0,025	0,000	0,102	0,095
10				0,025	0,050	0,000	0,105	0,097

11				0,050	0,000	0,000	0,095	0,095
12				0,050	0,000	0,025	0,101	0,097
13				0,050	0,000	0,050	0,103	0,098
14				0,050	0,025	0,000	0,106	0,100
15				0,050	0,050	0,000	0,111	0,103
16				0,000	0,000	0,000	0,091	0,091
17				0,000	0,000	0,025	0,091	0,088
18				0,000	0,000	0,050	0,092	0,087
19				0,000	0,025	0,000	0,105	0,100
20				0,000	0,050	0,000	0,110	0,102
21				0,025	0,000	0,000	0,109	0,109
22				0,025	0,000	0,025	0,111	0,107
23	0,25	0,40	0,35	0,025	0,000	0,050	0,111	0,106
24				0,025	0,025	0,000	0,121	0,115
25				0,025	0,050	0,000	0,124	0,116
26				0,050	0,000	0,000	0,115	0,115
27				0,050	0,000	0,025	0,116	0,112
28				0,050	0,000	0,050	0,116	0,110
29				0,050	0,025	0,000	0,124	0,118
30				0,050	0,050	0,000	0,128	0,120
31				0,000	0,000	0,000	0,050	0,050
32				0,000	0,000	0,025	0,052	0,049
33				0,000	0,000	0,050	0,052	0,048
34				0,000	0,025	0,000	0,067	0,061
35				0,000	0,050	0,000	0,073	0,064
36				0,025	0,000	0,000	0,072	0,072
37	0,34	0,14	0,53	0,025	0,000	0,025	0,074	0,072
38				0,025	0,000	0,050	0,074	0,070
39				0,025	0,025	0,000	0,085	0,079
40				0,025	0,050	0,000	0,091	0,082
41				0,050	0,000	0,000	0,079	0,079
42				0,050	0,000	0,025	0,082	0,079
43				0,050	0,000	0,050	0,082	0,078
44				0,050	0,025	0,000	0,093	0,086

45	0,050	0,050	0,000	0,097	0,088

Table 3. Effect sizes (R2s) given the parameter settings. ID = identification number of the given parameter setting, a^2 = squared a effect parameter, c^2 = squared c effect parameter, e^2 = squared e effect parameter, g^2 = squared g effect parameter, b^2 = squared b effect parameter, x^2 = squared x effect parameter. M0mzdz is the effect size given by M0 in the MZ-DZ between family design and M0dz is the effect size given by M0 in the DZ within-family design.

Discussion

The purpose of this study was to propose a model to detect covariance between additive genetic variables and shared environmental variables stemming from two different sources in twin models, namely from sibling-interaction and from cultural transmission. To this end, we introduced a parent-sibling model, where the regression equations include the offspring phenotypes as dependent variables and parental and / or offspring PRS scores as predictors. We incorporated the method of Selzam et. al. (2019) and Okbay et. al. (2022) in a combined model to detect cov(AC) stemming from sibling interaction and/or cultural transmission. Subsequently, we tested the power of the model given different g, b, and x, moreover a,c and e effects. The main question of interest concerns the power to detect cov(AC), generally, and to resolve the sources of cov(AC), specifically.

The results showed that the model is capable of detecting and dissecting cov(AC) with a power ranging from 0.7 to 1, given the parameter settings. Therefore, we can conclude that the model works sufficiently in detecting the covariance term, and that it can differentiate between the two sources, i.e., sibling-interaction or cultural transmission, as a function of the g, b, or x effects. Moreover, based on our results, we found that the power of the tests of cov(AC) did not depend appreciably on the size of the parameters a (additive genetic effect) or c (shared environmental effect). Some fluctuations can occur between the given sets of a,c and e effects, but as we saw from Table 2A, 2B and 2C, the effect on the power is small. Although, the model has acceptable resolution, given the present design settings, it also has some shortcomings. As we could see from the results, in the case when both sibling-interaction and cultural transmission is present, the power of rejecting the null-hypothesis when fitting M3 (Equation 9) is strongly dependent on the magnitude of the value of the sibling-interaction and the cultural transmission effect. Therefore, if the effect is not large

enough (in our settings, g^2=0.025 is the middle value) it might take up on both sources with smaller than 0.8 power. This means that there is a higher probability than 20 percent that we would come to the false conclusion, given that sibling-interaction is detected by the sub-models, that the covariance is stemming from sibling-interaction exclusively, when in reality it is stemming from both sources, just the contribution of cultural transmission effect to the total phenotypic variance is not detected by the model. Therefore, in this case the probability of falsely rejecting the alternative hypothesis is higher compared to other cases, where cov(AC) present and detected by the model with a power larger than or equal to 0.8.

Another interesting finding is regarding the sibling-interaction effect. We supposed that the sibling interaction effect also stems from two separate sources, but we set either the level-processes (parameter b) to zero or the phenotypic interaction (parameter x) to zero, assuming that the results will be inflated. However, we can see from the results that if we fit M3 and both the cultural transmission effect and the sibling interaction effect is present, there are differences in the power of the tests between g and b settings versus g and x settings. In these cases, the power of rejecting the null hypothesis x = 0 when fitting M3 is smaller compared to rejecting the null hypothesis b = 0. However the effect sizes (i.e.,the chosen values of the parameter b^2, g^2, x^2) in these cases are not significantly different from each other. Given these results, we can say that there is a differentiation between these two sibling-interaction effects, but investigating how we can address this issue is outside of the scope of this paper. We can conclude, however, that a model to account for the difference between the two sources will advance our understanding of cov(AC) detection.

Moreover, when fitting Model 2 and Model 3, the power to detect cov(AC) is higher in the MZ & DZ samples than in the DZ samples only, given that cov(AC) is present. This does not interfere with the results outlined above, since the power in the DZ dataset is sufficient if we take 0.8 as a threshold. However, with smaller sample sizes it might lead to falsely reject the

hypothesis that cov(AC) is (partially) stemming from cultural transmission with certain parameter settings, if we limit the analyses to the DZ sample.

Finally, we address the limitation that are present in the paper. The first limitation is that we did not take into consideration if dominance (denoted as D) is present in the model instead of C given that the design is limited to ACE in the paper. Investigating the possibility of estimating the presence of D in the ACE model is out of the scope of the current paper. The ADE model is, however, of interest to pursue in the future, as negative cov(AC), if unmodeled (i.e., in regular genetic covariance structure modelling of twin data), may give rise to spurious D variance. The second limitation is that we assumed assortative mating to be absent. However, we know that in the case of certain traits such as educational attainment (Baker et. al., 1996, Robinson et. al., 2017) or psychopathology (Nordsletten et. al., 2016), partners tend to resemble each other, which may influence the cov(AC) and our model's resolution (see Keller et al, 2009 for the parents and twin model which includes cultural transmittion and phenotypic assortment). Torvik et. al. (2022) proposed a model that shed light on the mechanisms of assortative mating. The third limitation is considering the sample size. Given that the sample sizes are equal, with 2000 MZ and 2000 DZ twins, and this ratio has influence on the power of the twin model (Visscher, 2004, see also Dolan et. al., 2021), the resolution of the model might change if the sample sizes of MZ and DZ twins are not equal. Moreover, the sample size of 2000 MZ and 2000 DZ twins may be considered to be large. The fourth and final limitation is that, as it has been mentioned already in the methods section, the possibility to investigate more a, c and e parameter settings. We concluded in the paper, that based on our preset values, the power of the model to detect cov(AC) is not dependent on the a, c and e values. Still, it would be useful to investigate the model with further a, c, and e effect settings, given that there might be certain, probably very specific cases, when the power to reject the null hypothesis in the model might decrease. However,

we suppose that our preset values are representative of the possible values of a, c and e in a twin study.

Ethical approval

This article does not contain any studies with human participants or animal subjects, therefore special ethical approval was not needed.

References:

Baker, L. A., Treloar, S. A., Reynolds, C. A., Heath, A. C. & Martin, N. G. (1996). Genetics of educational attainment in Australian twins: Sex differences and secular changes. B*ehav. Genet* **26**, 89–102 (1996). DOI: 10.1007/BF02359887

Bates, T. C., Maher, B. S., Medland, S. E., McAloney, K., Wright, M. J., Hansell, N. K., & Gillespie, N. A. (2018). *T*he nature of nurture: Using a virtual-parent design to test parenting effects on children's educational attainment in genotyped families. *Twin Research and Human Genetics*, 21(2), 73-83. DOI: 10.1017/thg.2017.92

Collister, J. A., Liu, X., & Clifton, L. (2022). Calculating polygenic risk scores (PRS) in UK Biobank: a practical guide for epidemiologists. *Frontiers in Genetics*, *13*, 105. DOI: 10.3389/fgene.2022.818574

Dolan, C.V., de Kort, J.M., van Beijsterveldt, T.C.E.M. et al. (2014). GE Covariance Through Phenotype to Environment Transmission: An Assessment in Longitudinal Twin Data and Application to Childhood Anxiety. *Behav Genet* 44, 240–253 (2014). https://doi.org/10.1007/s10519-014-9659-5

Dolan, C. V., Huijskens, R. C., Minică, C. C., Neale, M. C., & Boomsma, D. I. (2021). Incorporating polygenic risk scores in the ACE twin model to estimate A–C covariance. *Behavior Genetics*, 51(3), 237-249. DOI: 10.1007/s10519-020-10003-w

Eaves, L. (1976). A model for sibling effects in man. *Heredity*, 36(2), 205-214.

Fulker DW (1982) Extensions of the classical twin method. In: *Genetics Human (ed) Part A: the unfolding genome*. Alan R. Liss Inc., New York, pp 395–406

Keller, M. C., Medland, S. E., Duncan, L. E., Hatemi, P. K., Neale, M. C., Maes, H. H., & Eaves, L. J. (2009). Modeling extended twin family data I: Description of the Cascade model. *Twin Research and Human Genetics*, 12(1), 8-18. DOI: 10.1375/twin.12.1.8

Klahr, A. M., & Burt, S. A. (2014). Elucidating the etiology of individual differences in parenting: A meta-analysis of behavioral genetic research. *Psychological Bulletin*, 140(2), 544. DOI: 10.1037/a0033720

Kong, A., Thorleifsson, G., Frigge, M. L., Vilhjalmsson, B. J., Young, A. I., Thorgeirsson, T. E., ... , & Gudbjartsson, D. F. (2018). The nature of nurture: Effects of parental genotypes. *Science*, 359(6374), 424-428. DOI: 10.1126/science.aan6877

Minică, C. C., Dolan, C. V., Kampert, M., Boomsma, D. I., & Vink, J. M. Sandwich (2015). Corrected standard errors in family-based genome-wide association studies. *European journal of human genetics*, *23*(3), 388-394. DOI: 10.1038/ejhg.2014.94.

Nordsletten, A. E. et al. (2016). Patterns of nonrandom mating within and across 11 major psychiatric disorders. *JAMA Psychiatry* **73**, 354–361. DOI: 10.1001/jamapsychiatry.2015.3192

Okbay, A., Wu, Y., Wang, N., et al. (2022). Polygenic prediction of educational attainment within and between families from genome-wide association analyses in 3 million individuals. *Nature Genetics*, 54, 437-449. DOI: 10.1038/s41588-022-01016-z

Polderman, T. J., Benyamin, B., De Leeuw, C. A., Sullivan, P. F., Van Bochoven, A., Robinson, M. R. et al. (2017). Genetic evidence of assortative mating in humans. *Nat. Hum. Behav.* **1**, 0016. DOI: 10.1038/s41562-016-0016

RStudio Team (2020). RStudio: Integrated Development for R. RStudio, *PBC, Boston,* MA URL: http://www.rstudio.com/.

Selzam, S., Ritchie, S. J., Pingault, J. B., Reynolds, C. A., O'Reilly, P. F., & Plomin, R. (2019). Comparing within-and between-family polygenic score prediction. *The American Journal of Human Genetics*, *105*(2), 351-363. DOI: 10.1016/j.ajhg.2019.06.006

Torvik, F. A., Eilertsen, E. M., Hannigan, L. J., Cheesman, R., Howe, L. J., Magnus, P., ... & Ystrom, E. (2022). Modeling assortative mating and genetic similarities between partners, siblings, and in-laws. *Nature Communications*, *13*(1), 1108. DOI: https://doi.org/10.1038/s41467-022-28774-y

Visscher, P. M. (2004). Power of the classical twin design revisited. *Twin Research and Human Genetics*, *7*(5), 505-512. DOI: 10.1375/1369052042335250

Supplementary material

R code for data simulation

```
rm(list=ls(all=TRUE))
library(MASS)
library(geepack)
library(sys)
start_time=Sys.time()
# replaces gee()
#
# ------------------------------------------------------------------------ settings
#
# set.seed(2101)
#
nrep=500 # number of replication per parameter setting preferable 500 or 1000 (depending on time)
#
#
#
cmethod='independence'  # gee error cov structure
#
nmz=2000  # sample size
ndz=2000  # sample size
# parameters settings
par_as=p_as=sqrt(c(.25)) # 0.34-- based on south et. al. 2018, neuroticism A component, second karoline
schochousebo 2003 norway men bmi
par_cs=p_cs=sqrt(c(.40)) # 0.14 -- south 2018 c component, same
par_es=p_es=sqrt(c(.35)) #0.53 -- south 2018 e component, same
#par_as=sqrt(c(.34,.45,.25))
#par_cs=sqrt(c(.14,.31,.40))
#par_es=sqrt(c(.53,.24,.35))
#             sources of A-C covariance effect size based on phenotypic variance
par_gs=p_gs=sqrt(c(0,0.025,0.05))    # parent genotype to twins phenotype
par_bs=p_bs=sqrt(c(0,0.025,0.05))     # twin 1 (2) genotype to twin 2 (1) phenotype
par_xs=p_xs=sqrt(c(0,0.025,0.05))  # twin 1 (2) phenotype to twin 2 (1) phenotype - "sib interaction model"
#
# the number of setting in the factorial design
#
nset=
  length(p_as)*length(p_es)*length(p_cs)*
  length(p_gs)*length(p_bs)*length(p_xs)
nset
#
setkeep=matrix(NA,nset,38)   # to keep settings
reskeep=matrix(NA,nset*nrep,38)   # to keep results each data set
#
ng=50  # number of diallelic loci    ---q
ngp=10  # number of loci comprising polygenic score pgs (0 <= npg <= ng).---q
p_pgs=ngp/ng  # percentage of genetic variance explained by pgs ------q
p_pgs
#
p_A=1-p_pgs # not explained = A without pgs effect
VA1=p_A; VP=p_pgs;VC=1; VE=1
#
ii=0  # count sets
jj=0  # count from 1:(nset*nrep)
for (par_a in par_as) {
  for (par_c in par_cs) {
    for (par_e in par_es) {

      for (par_g in par_gs) {
        for (par_b in par_bs) {
          for (par_x in par_xs) {
            if (par_b != 0 && par_x !=0) next
            ii=ii+1 # count sets in factorial design
            setkeep[ii,1:11]=c(nmz,ndz, par_a, par_c, par_e, par_g, par_b, par_x, p_pgs, p_A,ii)
            #
            for (irep in 1:nrep) {  ###### terminates at line +/- 352
              jj=jj+1
              #
              # start ordinary simulation.
              #
              pal=.5     # maf all GVs same maf ... Minor Allele Freqs (maf)
              qal=1-pal  # major allele freq
              #
              varGV=2*pal*qal  # variance of GV per SNP
              #
              bs=rep(1,ngp) #
              #
              #
              NMZ=nmz
              NDZ=ndz   # number of mz and dz twin pairs
              nfam=nmz+ndz # number of families
              #
```

```
# ------------------------------------------------------------------------------  real data
sim start
# simulate parental alleles assuming random mating and linkage equilibrium
#
am=array(0,c(nfam,ng,2)) # mother alleles
af=array(0,c(nfam,ng,2)) # father alleles
#
gm=matrix(0,nfam,ng) # mother genotype
gf=matrix(0,nfam,ng) # father genotype
#
Am=matrix(0,nfam,1)  # mother pgs
Af=matrix(0,nfam,1)  # father pgs
#
pgsm=matrix(0,nfam,1) # mother polygenic scores
pgsf=matrix(0,nfam,1) # father polygenic scores
#
at1=array(0,c(nfam,ng,2))  # twin 1 alleles transmitted from m and f
at2=array(0,c(nfam,ng,2))  # twin 2 alleles transmitted from m and f
ant1=array(0,c(nfam,ng,2))  # t1 alleles not transmitted - ibd = 0 with t1
ant2=array(0,c(nfam,ng,2))  # t2 alleles not transmitted - ibd = 0 with t2
#
g1=matrix(0,nfam,ng)  # genotypes twin 1
g2=matrix(0,nfam,ng)  # genotypes twin 2
#
A1=matrix(0,nfam,2) # pgs tw 1 based on 1) transmitted, 2) based on non-transmitted
A2=matrix(0,nfam,2) # pgs tw 2 based on 1) transmitted, 2) based on non-transmitted
#
pgs1=matrix(0,nfam,2) # twin 1 polygenic scores 1=transmitted 2=not transmitted
pgs2=matrix(0,nfam,2) # twin 2 polygenic scores 1=transmitted 2=not transmitted
#
# parental alleles - simulated.
#
for (i in 1:ng) {
  am[,i,1] = sample(c(0,1),nfam,replace=T,prob=c(pal,qal))  # mother allele 1
  am[,i,2] = sample(c(0,1),nfam,replace=T,prob=c(pal,qal)) # mother allele 2
  af[,i,1] = sample(c(0,1),nfam,replace=T,prob=c(pal,qal)) # father allele 1
  af[,i,2] = sample(c(0,1),nfam,replace=T,prob=c(pal,qal)) # father allele 2
}
#
for (i in 1:ng) {
  #
  # offspring (twins) inherits alleles from father and mother
  #
  mt1=sample(c(1,2),nfam,replace=T,prob=c(.5,.5)) # sample maternal alleles for transmission
  ft1=sample(c(1,2),nfam,replace=T,prob=c(.5,.5)) # sample paternal alleles for transmission
  mnt1=3-1*mt1  # 1->2 2-> 1  # nontransmitted maternal alleles  #---- ? 1*
  fnt1=3-1*ft1 # 1->2 2-> 1  # nontransmitted paternal alleles
  # twin 2
  mt2=sample(c(1,2),nfam,replace=T,prob=c(.5,.5))
  ft2=sample(c(1,2),nfam,replace=T,prob=c(.5,.5))
  mnt2=3-1*mt2
  fnt2=3-1*ft2
  # offspring alleles transmitted and not transmitted
  for (k in 1:nfam) {
    at1[k,i,1] = am[k,i,mt1[k]]   # transm
    at1[k,i,2] = af[k,i,ft1[k]]   # transm
    ant1[k,i,1]= am[k,i,mnt1[k]]  # not transm
    ant1[k,i,2]= af[k,i,fnt1[k]]  # not transm
    at2[k,i,1] = am[k,i,mt2[k]]   # transm
    at2[k,i,2] = af[k,i,ft2[k]]   # transm
    ant2[k,i,1]= am[k,i,mnt2[k]]  # not transm
    ant2[k,i,2]= af[k,i,fnt2[k]]  # not transm
  } # nfam
} # ng
#
# A scores polygenic scores representing A = total add gen
#
for (i in 1:ng) {
  Am[,1]=Am[,1]+(am[,i,1]+am[,i,2])
  Af[,1]=Af[,1]+(af[,i,1]+af[,i,2])
  A1[,1]=A1[,1]+(at1[,i,1]+at1[,i,2])
  A1[,2]=A1[,2]+(ant1[,i,1]+ant1[,i,2])
  A2[,1]=A2[,1]+(at2[,i,1]+at2[,i,2])
  A2[,2]=A2[,2]+(ant2[,i,1]+ant2[,i,2])
}
#
# polygenic scores based on first ngp genetic variants
# weighted by bs
#
for (i in 1:ngp) {
  pgsm[,1]=pgsm[,1]+bs[i]*(am[,i,1]+am[,i,2])
  pgsf[,1]=pgsf[,1]+bs[i]*(af[,i,1]+af[,i,2])
  pgs1[,1]=pgs1[,1]+bs[i]*(at1[,i,1]+at1[,i,2])
  pgs1[,2]=pgs1[,2]+(ant1[,i,1]+ant1[,i,2])
  pgs2[,1]=pgs2[,1]+bs[i]*(at2[,i,1]+at2[,i,2])
  pgs2[,2]=pgs2[,2]+(ant2[,i,1]+ant2[,i,2])
}
#
```

```
              #  C
              #
              C1_=scale(rnorm(nfam,0,1))  # shared Environment of the twins  C1_ variance = 1 #scale what
is it exacly?
              #
              E1=scale(rnorm(nfam,0,1))  #   unshared Environment  variance = 1
              E2=scale(rnorm(nfam,0,1))  #   unshared Environment variance = 1
              #
              # scale A to variance = 1  parameter a is the effect
              #
              Am[,1]=scale(Am[,1])   # polygenic score mother standardized
              Af[,1]=scale(Af[,1])   # polygenic score father standardized
              A1[,1]=scale(A1[,1])   # polygenic score tw1 standardized (transmitted alleles)
              A1[,2]=scale(A1[,2])   # polygenic score tw1 standardized (not transmitted alleles)
              A2[,1]=scale(A2[,1])   # polygenic score tw2 standardized (transmitted alleles)
              A2[,2]=scale(A2[,2])   # polygenic score tw2 standardized (not transmitted alleles)
              #
              # scale polygenic scores
              #
              pgsm[,1]=scale(pgsm[,1])   # polygenic score mother standardized
              pgsf[,1]=scale(pgsf[,1])   # polygenic score father standardized
              pgs1[,1]=scale(pgs1[,1])   # polygenic score tw1 standardized (transmitted alleles)
              pgs1[,2]=scale(pgs1[,2])   # polygenic score tw1 standardized (not transmitted alleles)
              pgs2[,1]=scale(pgs2[,1])   # polygenic score tw2 standardized (transmitted alleles)
              pgs2[,2]=scale(pgs2[,2])   # polygenic score tw2 standardized (not transmitted alleles)
              #
              # A C E
              #          DZ
              C11= C1_ # twin 1
              C12= C1_ # twin 2  ....
              #           A              C          E     AC cov        AC cov         AC cov (not due to
parents)
              ph1_=par_a*A1[,1] + par_c*C11 + par_e*E1 + par_g*Am[,1] + par_g*Af[,1] + par_b*A2[,1] #
offspring 1 pheno
              ph2_=par_a*A2[,1] + par_c*C12 + par_e*E2 + par_g*Am[,1] + par_g*Af[,1] + par_b*A1[,1] #
offspring 1 pheno# offspring 2 pheno
              # siblinf interaction: mutual influence
              ph1=ph1_+par_x*ph2_
              ph2=ph2_+par_x*ph1_
              #
              # 1:ndz
              #
              phdatdz=matrix(0,ndz,2)
              phdatdz[,1]=ph1[1:ndz]
              phdatdz[,2]=ph2[1:ndz]
              cov(phdatdz)
              cor(phdatdz)
              #
              # MZ       MZ
              #
              C11= C1_ # twin 1
              C12= C1_ # twin 2  ....
              #          A1 and A1                                                                  A1 and A1
              ph1_=par_a*A1[,1] + par_c*C11 + par_e*E1 + par_g*Am[,1] + par_g*Af[,1] + par_b*A1[,1] #
offspring 1 pheno
              ph2_=par_a*A1[,1] + par_c*C12 + par_e*E2 + par_g*Am[,1] + par_g*Af[,1] + par_b*A1[,1]  #
offspring 1 pheno# offspring 2 pheno
              # interaction
              ph1=ph1_+par_x*ph2_
              ph2=ph2_+par_x*ph1_
              #
              # ndz+1 : nfam .... second half
              phdatmz=matrix(0,nmz,2)
              phdatmz[,1]=ph1[(ndz+1):nfam]  # ndz+1 to nfam MZs
              phdatmz[,2]=ph2[(ndz+1):nfam]  # ndz+1 to nfam MZs
              #
              #cov(phdatmz)
              #cor(phdatmz)
              #
              # phdatmz ... add polygenic scores
              #                 mother     father      dz1 t, nt   dz2 t,nt
              pgdatdz=cbind(pgsm[1:ndz,],pgsf[1:ndz,],pgs1[1:ndz,],pgs2[1:ndz,])
              #                 mother            father                   mz1 t, nt =   mz2 t, nt
(duplicate 1)

pgdatmz=cbind(pgsm[(1+ndz):nfam,],pgsf[(1+ndz):nfam,],pgs1[(1+ndz):nfam,],pgs1[(1+ndz):nfam,])
              #
              # stochastic simulated data
              phdatmz=as.data.frame(cbind(phdatmz,pgdatmz))
              phdatdz=as.data.frame(cbind(phdatdz,pgdatdz))
              # add sum and mean
              colnames(phdatdz) = colnames(phdatmz) =  vnames1 =
                c('pht1','pht2',
                  'pgsm','pgsf','pgst1','pgsnt1','pgst2','pgsnt2')
              adddz=cbind(phdatdz$pgsm+phdatdz$pgsf, (phdatdz$pgst1+phdatdz$pgst2)/2)
              colnames(adddz) = c('pgsmf','mpgst')
              phdatdz = cbind(phdatdz, adddz)
              addmz=cbind(phdatmz$pgsm+phdatmz$pgsf, (phdatmz$pgst1+phdatmz$pgst2)/2)
```

```
              colnames(addmz) = c('pgsmf','mpgst')
              phdatmz = cbind(phdatmz, addmz)
              #
              c(1,2,3,4,5,7,9,10) -> i1

              #
              apply(phdatmz,2,var)
              round(cor(phdatmz),3)
              #
              apply(phdatdz,2,var)
              round(cor(phdatdz),3)
              #
              # [1] "pht1"   "pht2"   "pgsm"   "pgsf"   "pgst1"  "pgsnt1" "pgst2"  "pgsnt2" "pgsmf"
"mpgst"
              # Organize data in long format simulated data
              #
              phdatmzL = matrix(0,nmz*2,8)
              phdatmzL[,2]=c(c(1:nmz),c(1:nmz))
              phdatmzL[,3]=c(phdatmz$pht1,phdatmz$pht2)
              phdatmzL[,4]=c(phdatmz$pgsm,phdatmz$pgsm)
              phdatmzL[,5]=c(phdatmz$pgsf,phdatmz$pgsf)
              phdatmzL[,6]=c(phdatmz$pgst1,phdatmz$pgst2)
              phdatmzL[,7]=c(phdatmz$pgsmf,phdatmz$pgsmf)
              phdatmzL[,8]=c(phdatmz$mpgst,phdatmz$mpgst)
              #
              ix_ = sort.int(phdatmzL[,2], index.return=T)
              phdatmzL = phdatmzL[ix_$ix,]
              colnames(phdatmzL)=c("zyg","famnr","ph","pgsm","pgsf","pgst","pgsmf","mpgst")
              phdatmzL=as.data.frame(phdatmzL)
              #
              # Long format simulated data
              phdatdzL = matrix(1,ndz*2,8)
              phdatdzL[,2]=nmz + c(c(1:ndz),c(1:ndz))
              phdatdzL[,3]=c(phdatdz$pht1,phdatdz$pht2)
              phdatdzL[,4]=c(phdatdz$pgsm,phdatdz$pgsm)
              phdatdzL[,5]=c(phdatdz$pgsf,phdatdz$pgsf)
              phdatdzL[,6]=c(phdatdz$pgst1,phdatdz$pgst2)
              phdatdzL[,7]=c(phdatdz$pgsmf,phdatdz$pgsmf)
              phdatdzL[,8]=c(phdatdz$mpgst,phdatdz$mpgst)
              #
              ix_ = sort.int(phdatdzL[,2], index.return=T)
              phdatdzL = phdatdzL[ix_$ix,]
              colnames(phdatdzL)=c("zyg","famnr","ph","pgsm","pgsf","pgst","pgsmf","mpgst")
              phdatdzL=as.data.frame(phdatdzL)
              #
              phdatL=rbind(phdatmzL, phdatdzL)
              #
              # simulated stochastically
              #                  wide        wide     long      long      long mz+dz
              # data sets are  phdatdz and phdatdz, phdatdzL, phdatmzL, phdatL
              #                  wide          wide        long       long        long mz+dz
              # simulated exactly
              # data sets are  phdatdz_e and phdatdz_e, phdatdzL_e, phdatmzL_e, phdatL_e
              #              pheno t1 pheno t2 mother    father   twin 1  nt twin1  twin2   nt twin2  m+f
prs  mean twin prs
              # colnames [1] "pht1"   "pht2"   "pgsm"   "pgsf"   "pgst1"  "pgsnt1" "pgst2"  "pgsnt2"
"pgsmf"  "mpgst"
              #
              # ------------------------------------------------- end data sim
              #
              # regression analyses. based on simulated data (not exact) ...
              #
              # DZ twin 1 only ... just pgsmf
              #
              M0dz=lm(pht1~pgst1, data=phdatdz)#)$coefficients               #  single twin dz 1    ...
pheno on pgs ...just regression
              M1dz=lm(pht1~pgsmf+pgst1, data=phdatdz)#)$coefficients         #  single twin dz 1    ...
pheno on pgs + m&f pgs test of cov(AC)
              #M2dz=lm(pht1~mpgst+pgst1, data=phdatdz)#)$coefficients        #    single twin dz 1   ...
pheno on pgs + twin mean pgs test of cov AC ... not important
              #M3dz=lm(pht1~pgsmf+mpgst+pgst1, data=phdatdz)#)$coefficients #     single twin dz 1 ...
pheno on pgs, m&f pgs + twin mean pgs test of cov AC
              #
              # 2 dz twins regression using gee
              #
              geeM0dzL=geeglm(ph~pgst, id=famnr, corstr=cmethod, data=phdatdzL)#)$coefficients #  on pgs
              geeM1dzL=geeglm(ph~pgsmf+pgst, id=famnr, corstr=cmethod, data=phdatdzL)#)$coefficients #
m+f pgs + twin pgs #Kong, Bates
              geeM2dzL=geeglm(ph~mpgst+pgst, id=famnr, corstr=cmethod,data=phdatdzL)#)$coefficients  #
mean twin pgs + twin pgs #Selzam
              geeM3dzL=geeglm(ph~pgsmf+mpgst+pgst, id=famnr, corstr=cmethod,data=phdatdzL)#)$coefficients
#    # mf+mean+twin #test of X
              #
              geeM0mzdzL=geeglm(ph~pgst, id=famnr, corstr=cmethod,data=phdatL)#)$coefficients #
same sequence of analysis
              geeM1mzdzL=geeglm(ph~pgsmf+pgst, id=famnr, corstr=cmethod,data=phdatL)#)$coefficients #
              geeM2mzdzL=geeglm(ph~mpgst+pgst, id=famnr, corstr=cmethod,data=phdatL)#)$coefficients  #
```

```
              geeM3mzdzL=geeglm(ph~pgsmf+mpgst+pgst, id=famnr, corstr=cmethod,data=phdatL)#)$coefficients
#
              R2M0mzdzL=summary(lm(ph~pgst,data=phdatL))$r.squared #          same sequence of analysis
              R2M1mzdzL=summary(lm(ph~pgsmf+pgst,data=phdatL))$r.squared #
              R2M2mzdzL=summary(lm(ph~mpgst+pgst,data=phdatL))$r.squared  #
              R2M3mzdzL=summary(lm(ph~pgsmf+mpgst+pgst,data=phdatL))$r.squared  #     # mf+mean+twin
              #
              R2M0dzL=summary(lm(ph~pgst,data=phdatdzL))$r.squared #          same sequence of analysis
              R2M1dzL=summary(lm(ph~pgsmf+pgst,data=phdatdzL))$r.squared #
              R2M2dzL=summary(lm(ph~mpgst+pgst,data=phdatdzL))$r.squared  #
              R2M3dzL=summary(lm(ph~pgsmf+mpgst+pgst,data=phdatdzL))$r.squared  #
              #
              # reskeep=matrix(NA,nrep,30)   # to keep results
              # get power exact power pgsmf, mpgst, pgsmf+mpgst: 3 tests ...  dz pairs,   dz + mz pairs
              # get power exact power pgsmf: 1 tests ...  dz singles
              #
              # dz 1 pgsmf test ... test of mpgst does not apply given 1 dz
              # model M1dz=lm(pht1~pgsmf+pgst1, data=phdatdz) ... 1dz test of pgsmf
              tmp=summary(M1dz)$coefficients
              M1dzest=tmp[2,1]; M1dzse=tmp[2,2]; M1dzp=tmp[2,4]
              reskeep[jj,1:3]=c(M1dzest,M1dzse,M1dzp) # est st and power
              #
              # dz 1+2 test power
              # model geeM1dzL=geeglm(ph~pgsmf+pgst, id=famnr, corstr=cmethod, data=phdatdzL) #
              # pgsmf
              tmp=summary(geeM1dzL)$coefficients
              geeM1dzest=tmp[2,1]; geeM1dzse=tmp[2,2]; geeM1dzp=tmp[2,4]
              reskeep[jj,4:6]=c(geeM1dzest,geeM1dzse,geeM1dzp) # est st and power
              # dz 1+2 test power
              #geeM2dzL=geeglm(ph~mpgst+pgst, id=famnr, corstr=cmethod,data=phdatdzL)#)$coefficients  #
              # mpgst
              tmp=summary(geeM2dzL)$coefficients
              geeM2dzest=tmp[2,1]; geeM2dzse=tmp[2,2]; geeM2dzp=tmp[2,4]
              reskeep[jj,7:9]=c(geeM2dzest,geeM2dzse,geeM2dzp) # est st and power
              #
              # geeM3dzL=geeglm(ph~pgsmf+mpgst+pgst, id=famnr, corstr=cmethod,data=phdatdzL)#)$coefficients
#
              # pgsmf in presence of mfpgs
              tmp=summary(geeM3dzL)$coefficients
              geeM3dzest1=tmp[2,1]; geeM3dzse1=tmp[2,2]; geeM3dzp1=tmp[2,4];
              reskeep[jj,10:12]=c(geeM3dzest1,geeM3dzse1,geeM3dzp1) # est st and power
              # mpgst
              geeM3dzest2=tmp[3,1]; geeM3dzse2=tmp[3,2]; geeM3dzp2=tmp[3,4]
              reskeep[jj,13:15]=c(geeM3dzest2,geeM3dzse2,geeM3dzp2) # est st and power
              #
              # mz dz 1+2 test power
              # geeM1mzdzL=geeglm(ph~pgsmf+pgst, id=famnr, corstr=cmethod,data=phdatL_e)#)$coefficients #
              # pgsmf
              tmp=summary(geeM1mzdzL)$coefficients
              geeM1mzdzest=tmp[2,1]; geeM1mzdzse=tmp[2,2]; geeM1mzdzp=tmp[2,4];
              reskeep[jj,16:18]=c(geeM1mzdzest,geeM1mzdzse,geeM1mzdzp) # est st and power
              # dz 1+2 test power
              # egeeM2mzdzL=geeglm(ph~mpgst+pgst, id=famnr, corstr=cmethod,data=phdatL_e)#)$coefficients  #
              # mpgst
              tmp=summary(geeM2mzdzL)$coefficients
              geeM2mzdzest=tmp[2,1]; geeM2mzdzse=tmp[2,2]; geeM2mzdzp=tmp[2,4]
              reskeep[jj,19:21]=c(geeM2mzdzest,geeM2mzdzse,geeM2mzdzp) # est st and power
              #
              # geeM3mzdzL=geeglm(ph~pgsmf+mpgst+pgst, id=famnr,
corstr=cmethod,data=phdatL_e)#)$coefficients  # #
              # pgsmf in presence of mfpgs
              tmp=summary(geeM3mzdzL)$coefficients
              geeM3mzdzest1=tmp[2,1]; geeM3mzdzse1=tmp[2,2]; geeM3mzdzp1=tmp[2,4];
              reskeep[jj,22:24]=c(geeM3mzdzest1,geeM3mzdzse1,geeM3mzdzp1) # est st and power
              # mpgst
              geeM3mzdzest2=tmp[3,1]; geeM3mzdzse2=tmp[3,2]; geeM3mzdzp2=tmp[3,4];
              reskeep[jj,25:27]=c(geeM3mzdzest2,geeM3mzdzse2,geeM3mzdzp2) # est st and power
              #
              reskeep[jj,28:35]=c(R2M0mzdzL,R2M1mzdzL,R2M2mzdzL,R2M3mzdzL,R2M0dzL,R2M1dzL,R2M2dzL,R2M3dzL)
              reskeep[jj,36:38]=c(ii,irep,jj)
            } # irep         line 352 about
          }}}}}} # the factorial design

ip=c(seq(3,27,3))
ie=c(seq(1,25,3))
is=c(seq(2,26,3))
colnames(reskeep) = c('dz1mfpgs_e','dz1mfpgs_s','dz1mfpgs1_p',
                      'dz2mfpgs_e','dz2mfpgs_s','dz2mfpgs_p',
                      'dz2mpgsm_e','dz2mpgsm_s','dz2mpgsm_p',
                      'dz2mfpgs_m_e','dz2mfpgs_m_s','dz2mfpgs_m_p',
                      'dz2mpgst_mf_e','dz2mpgst_mf_s','dz2mpgst_mf_p',
                      'mzdzmfpgs_e','mzdzmfpgs_s','mzdzmfpgs_p',
                      'mzdzmpgsm_e','mzdzmpgsm_s','mzdzmpgsm_p',
                      'mzdzmfpgs_m_e','mzdzmfpgs_m_s','mzdzmfpgs_m_p',
                      'mzdzmpgst_mf_e','mzdzmpgst_mf_s','mzdzmpgst_mf_p',
                      'R2M0mzdzL','R2M1mzdzL','R2M2mzdzL','R2M3mzdzL','R2M0dzL',
                      'R2M1dzL','R2M2dzL','R2M3dzL','set','irep','tcounter')
```

```
#------------------deleting NA rows (caused by the conditional statement in the for loop)
#------------------and creating 3 different matrices with standard errors, estimates and p-values)
head(reskeep)
tail(reskeep)
reskeep=na.omit(reskeep)
reskeep
reskeep=as.data.frame(reskeep)
reskeep
counter=reskeep[,36:38]

reskeep_p_values=cbind(reskeep[,ip],counter)
reskeep_p_values
reskeep_est=cbind(reskeep[,ie],counter)
reskeep_est
reskeep_se=cbind(reskeep[,is],counter)
reskeep_se
reskeep_R2=cbind(reskeep[,28:35],counter)
reskeep_R2
setkeep
setkeep=na.omit(setkeep[,1:10])
setkeep
setkeep=as.data.frame(setkeep)
setkeep
setkeep=cbind(setkeep,NA)
setkeep
colnames(setkeep)=c("nmz","ndz","par_a",'par_c', 'par_e', 'par_g', 'par_b', 'par_x', 'p_pgs', 'p_A','set')
setkeep
write.csv(setkeep,"path.csv")
#creating as much copies of each rows in setkeep as much repetitions we have
setkeep_full <- setkeep[rep(seq_len(nrow(setkeep)), each = nrep), ]
setkeep_full
#check if all good
head(reskeep)
check=as.data.frame(setkeep_full[,11] == reskeep[,36])
colnames(check) = 'BOOL'
check
check_BOOL=check[which(check$BOOL == FALSE),]
check_BOOL
#creating now the final dataframes ready fo the power calculations and further analysis
final_df_p = cbind(setkeep_full,reskeep_p_values, reskeep_R2)
final_df_p
final_df_est = cbind(setkeep_full, reskeep_est)
final_df_se = cbind(setkeep_full,reskeep_se)
final_df=cbind(setkeep_full,reskeep)
tail(final_df)
final_df=final_df[,-11]
R2_testing_table=final_df[,38:45]
R2_testing_table
mean_R2_test=apply(R2_testing_table,2,mean)
mean_R2_test
```

R code for power analysis and data visualization

```
df = data.frame(read.csv('path'.csv))
df2=data.frame(read.csv('path.csv'))
df2=df2[,39:46]
df2
df_se=df2
df_se
rep = 500
nrowdf=nrow(df)
nrowdf/rep
# Split the original dataset into smaller datasets
row_indices <- seq_len(nrow(df))
nrep <- ceiling(nrow(df) / rep)
df_list <- split(df, rep(1:nrep, each = rep, length.out = nrow(df)))
dim(df_list$`1`)
# Calculate the power for each column in each dataset
alpha <- 0.05
power_list <- lapply(df_list, function(x) {
  apply(x, 2, function(y) {
    sum(y < alpha) / length(y)
  })
})
power_list
df
power_table <- do.call(rbind, power_list)
power_table
setkeep_for_final_bind= data.frame(read.csv('path.csv'))
setkeep_for_final_bind
final_power_table=cbind.data.frame(setkeep_for_final_bind,power_table)
final_power_table=round(final_power_table,8)
final_power_table
write.csv(final_power_table,"path.csv”)

df_list_R <- split(df2, rep(1:nrep, each = rep, length.out = nrow(df))
df_list_Rdf_list_R = df_list_R[,39:46]
R2_list=lapply(df_list_R,colMeans)
R2_list
R2_table=do.call(rbind, R2_list)
R2_table
final_R2_table=cbind.data.frame(setkeep_for_final_bind,R2_table)
final_R2_table
write.csv(final_R2_table,"path.csv)
power_values_dz=final_power_table[14:17]
power_values_dz
power_values_dzmz=final_power_table[18:21]
power_values_dzmz
#df_se=data.frame(read.csv('/Users/macszerez.com/Desktop/VU GBH/Genes in Health and Behaviour 1st
year/set_1_full.csv'))
#df_se

library(ggplot2)
library(ggpubr)

# Define the parameter values (x-axis)
parameter_values <- 31:45

# Define colors for the power values
power_colors <- c("red", "blue", "green", "orange")

# Create a data frame with parameter, power values, and colors
data <- data.frame(Parameter = rep(parameter_values, each = 4),
                   Power = as.vector(t(power_values_dz)),
                   Models = rep(1:4, times = 15))

# Create a scatter plot with colored points
dz=ggplot(data, aes(x = Parameter, y = Power, color = factor(Models))) +
  geom_line(size = 1.5) +
  scale_color_manual(values = power_colors,
                     labels = c('M0dz','M1dz','M2dz','M3dz')) +
  scale_x_continuous(breaks = parameter_values, labels = parameter_values) +
  xlab("Parameter") +
  ylab("Power") +
  ggtitle("Change in Power with Parameter Values (set 3 DZ)")


# Create a data frame with parameter, power values, and colors
data <- data.frame(Parameter = rep(parameter_values, each = 4),
                   Power = as.vector(t(power_values_dzmz)),
                   Models = rep(1:4, times = 15))
```

```
# Create a scatter plot with colored points
mz=ggplot(data, aes(x = Parameter, y = Power, color = factor(Models))) +
  geom_line(size = 1.5) +
  scale_color_manual(values = power_colors,
                     labels = c('M0dzmz','M1dzmz','M2dzmdz','M3dzmz')) +
  scale_x_continuous(breaks = parameter_values, labels = parameter_values) +
  xlab("Parameter") +
  ylab("Power") +
  ggtitle("Change in Power with Parameter Values (set 3 DZ-MZ)")

figure <- ggarrange(dz,mz,
                    labels = c("A", "B"),
                    ncol = 2, nrow = 1)
figure
```