Summary of package validation results

May 02, 2022

Introduction

We completed extensive validation checks to ensure our power calculation procedures are correct. We compared three different methods of estimating power:

- `PUMP`: we denote these power estimates $\hat{p}_{pump}$.
- `PowerUpR`: we denote these power estimates $\hat{p}_{pow}$.
- Full Monte Carlo simulations: we denote these power estimates $\hat{p}_{sim}$.

For individual, unadjusted power, we compare `PUMP` to both `PowerUpR` and Monte Carlo simulations. `PowerUpR` does not support other types of power definitions and MTP adjustments, for all other definitions and adjustments we only compare the estimates from `PUMP` to Monte Carlo simulations.

To produce Monte Carlo estimates, we follow the full simulation approach outlined in detail in Section **??**. For each of $S = 5,000$ iterations, we simulate full dataset, apply the assumed model, and calculate p-values for each of the outcomes. This process gives us a $M \times S$ matrix of p-values. This matrix is then used to calculate empirical estimates of all definitions of power. To produce 95% confidence intervals for the power estimates, we construct a confidence interval assuming a conservative standard error estimate of $\sqrt{0.25/S}$.

The full validation code and results are in a supplementary GitHub repository pump_validate.

Validation scenarios

We conducted validation for all designs and models supported by `PowerUpR`. For each design and model, we ran several scenarios with varying parameter values. For most scenarios, we vary only one parameter at a time. Thus, to test varying ρ, we set $\rho = 0.2$ with all other parameters being set to the default values, and try another scenario with $\rho = 0.8$ with all other parameters being set to the default values. Table shows the default parameter values, and the other values we try to test out varying that parameter.

We do not vary:

- `M = 3` for all designs and models.
- `J` and `K` are fixed for all scenarios for a particular design and model.
- The scalar grand mean of control outcomes is always assumed to be zero.

Parameter	Default	Additional values
school size $\bar{n}$	50	75, 100
R^2	0.1	0.6
ρ	0.5	0.2, 0.8
MDES	(0.125, 0.125, 0.125)	(0.125, 0, 0)
ICC	0.2	0.7
ω	0.1	0.8

Table 1: Validation parameters

- The correlations between random effects and random impacts at a particular level are assumed to be zero.

In the `PUMP` package, the user chooses single parameter `rho` that specifies the correlation between test statistics. In order to generate a full simulated data set, there are a variety of correlation parameters that inform the correlation between outcomes. In order to ensure that the test statistics have correlation `rho`, we use `rho` as the correlation for all variables in the simulation process. For example, we may assume that each school has its own intercept and impact for each outcome. The correlation between the intercept columns for each pair of outcomes is assumed to be `rho`. The same strategy is used for the following correlations for each level: the correlation between covariates, the correlation between random intercepts (or residuals for level 1), and the correlation between random impacts. We further discuss the relationship between the correlation between variables and the correlation between test statistics in Section **??**.

Power validation results

Due to the higher computational burden of Westfall-Young procedures, we first consider validation for the Bonferroni, Holm, and Benjamini-Hochberg procedures. We consider Westfall-Young procedures in a separate section below.

Figure 1 is an example of a graph we use for validation. The green dots are `PUMP` estimate of power, the red dot is the `PowerUpR` estimate of power, and the 95% confidence intervals based on the Monte Carlo simulations are shown in blue. To validate that `PUMP` produces the expected result, we want to see the red and green points match, and for the red point to be within the blue intervals. Figure 1 shows the results across different types of power and different MTPs. We repeat this plot for a variety of different parameter values for each design and model.

The full set of these visualizations can be found in the github repository. Here, we summarize the general results and patterns.

Simulated power intervals. The Monte Carlo simulations produce a 95% confidence interval, and we check whether that interval contains the `PUMP` estimates. In other words, we check if $\hat{p}_{pump}$ is within $\hat{p}_{sim} \pm 1.96\sqrt{0.25/5000}$.

Simulated power point estimates. We compare the point estimates from the simulations to `PUMP`. To do so, we calculate the absolute difference between the power estimates from the simulations and from `PUMP`:

$$b_{sim} = |\hat{p}_{sim} - \hat{p}_{pump}|.$$

PowerUpR power point estimates. We compare the point estimates from the `PowerUpR` to `PUMP`. This comparison is only conducted for individual, unadjusted power. Similar to the last metric, we calculate the absolute difference between the power estimates from `PowerUpR` and from `PUMP`:

$$b_{pow} = |\hat{p}_{pow} - \hat{p}_{pump}|.$$

Discrepancies between PUMP and simulations.

In some scenarios, our validation results show some discrepancies. We discuss the broad categories here.

Expected discrepancies For all `d2.1_m2fc` and `d2.1_m2ff` designs and models, `PowerUpR` assumes `ICC.2 = 0`, but we do not make that assumption here. Thus, we expect to see a discrepancy between `PUMP` and `PowerUpR` except for scenarios when we assume `ICC.2 = 0`. PowerUpR! does allow for a non-zero `ICC.2` for `d2.1_m2fr`.

Unexpected discrepancies For some of the scenarios, the `PUMP` estimate is slightly outside the range of the simulation intervals. Our hypothesis is that for these scenarios, the `PUMP` estimate is accurate, but the simulation are producing inaccurate estimates. All scenarios with unexpected discrepancies occur for random effects models. In particular, discrepancies occur for the contexts `d2.1_m2fr` (2 scenarios) and `d3.2_m3rr2rc`

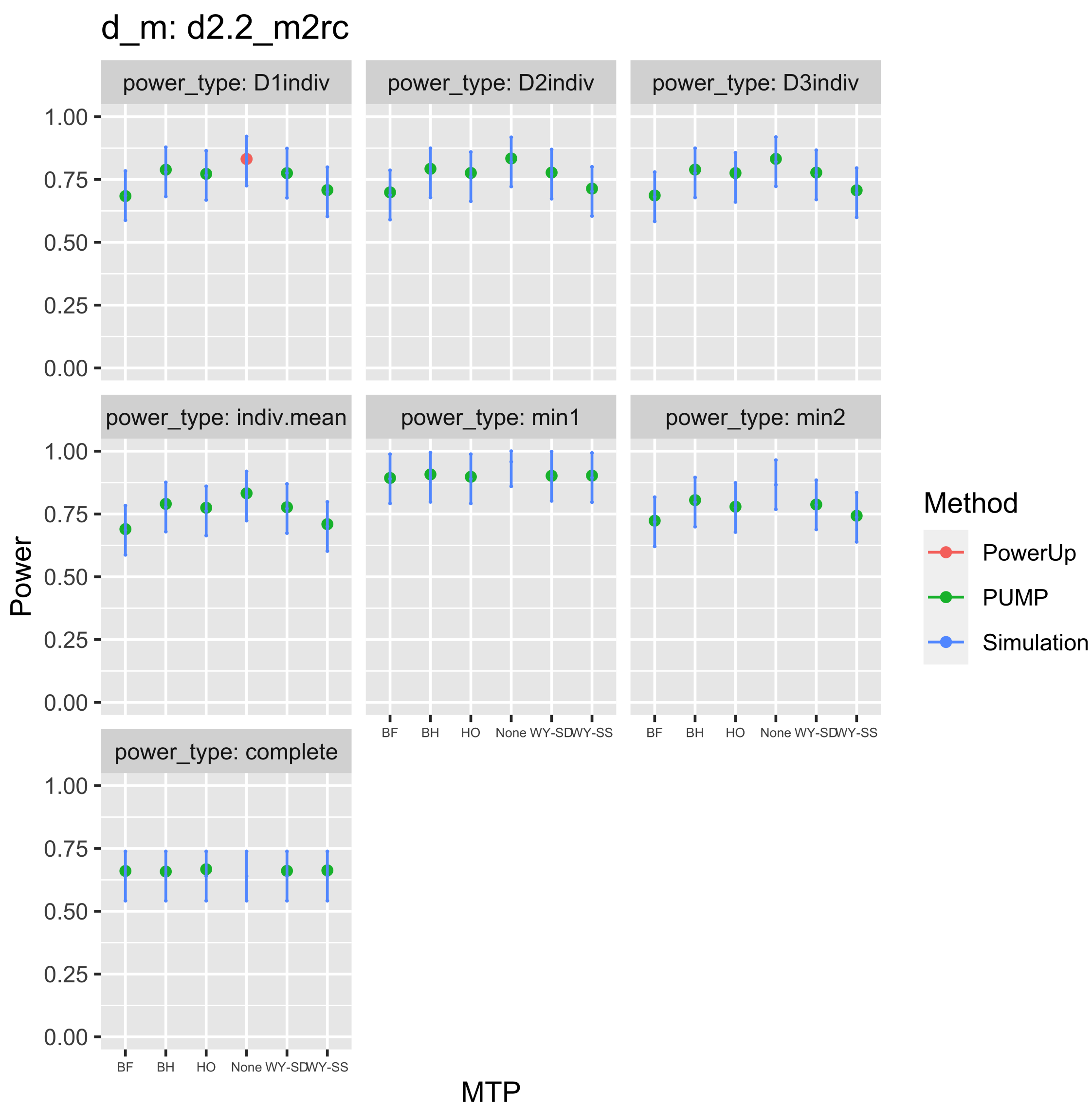

Figure 1: Validation plot

(6 scenarios). In general, we find that random effects models are more difficult to fit and less stable than fixed or constant effects models. For example, for many of the random effects models, the `lme4` package warns that the model has not converged, or has a singular fit. This instability could result in the simulations achieving poor estimates of power.

In particular, we find that discrepancies occur when random effects models are being fit on models where it is questionably appropriate to apply such a model. For the context `d2.1_m2fr`, we find that the `PUMP` estimate is slightly outside of the simulation interval for scenarios where either `ICC.2 = 0` or `omega.2 = 0`. If either of these parameters is zero, there is no treatment effect variation, but the model is attempting to fit random treatment impacts. Thus, it is not surprising that the simulation models do not perform well in this setting, and we do not necessarily trust power estimates for this scenario. Similarly, discrepancies occur for the `d3.2_m3rr2rc` context when either `omega.3 = 0` or `ICC.3 = 0`. In addition, we find three default scenarios for the `d3.2_m3rr2rc` context where the `PUMP` estimate lies outside the simulation interval when neither of these values values is zero. However, the discrepancy does not occur for all types of power, but only `min1` power, and in one case `D1indiv` power. For these scenarios, we find that increasing from `K = 10` to `K = 20` reduces or eliminates the discrepancy, lending credence to our hypothesis that the discrepancies are due to poor model fit.

Point estimates

For each design and model, we calculate a variety of estimates of b_{sim} and b_{pow}: we generate an estimate for each MTP and definition of power, across all validation scenario as described above (varying values of design and model parameters). In the validation github repository, we summarize the mean values of b_{sim} for each design and model, MTP, and definition and power, taking the average across all scenarios. These full results can be found in the "summary" document in the results folder. For the sake of brevity, we further summarize here by taking the mean and max values of b_{sim} across all definitions of power and MTPs The quantity b_{pow} can only be calculated for individual power with no adjustment, so we summarize it only across validation scenarios.

We find a high concordance between the simulated, `PUMP`, and `PowerUp` estimates. The summarized results are shown in Table 2. In the worst-case scenario, there is a discrepancy of 4% between the `PUMP` and simulation power estimates, and a discrepancy of 2% between the `PUMP` and `PowerUp` estimates. A conservative estimate of the Monte Carlo standard error is $\sqrt{0.25}/5000 = 0.0001$ (where 5000 is the number of simulation iterations).

d_m	mean.b.sim	max.b.sim	mean.b.pow
d2.1_m2fc	0.006	0.008	0.002
d2.1_m2ff	0.007	0.010	0.002
d2.1_m2fr	0.016	0.024	0.013
d2.2_m2rc	0.007	0.011	0.006
d3.1_m3rr2rr	0.008	0.013	0.006
d3.2_m3ff2rc	0.005	0.006	0.008
d3.2_m3rr2rc	0.019	0.039	0.020
d3.3_m3rc2rc	0.010	0.025	0.011

Table 2: Point estimate results for power

Westfall-Young procedures

Above, we have already shown `PUMP` estimates are valid across a wide variety of designs, models, and scenarios. One remaining piece is to validate that the Westfall-Young procedures are correct. Because Westfall-Young adjustments are computationally intensive, we consider a smaller number of scenarios for these procedures, but if we validate that the procedure is correct, it should generalize to all designs, models, and scenarios.

First, we wrote a series of careful unit tests ensuring that our code worked as expected for each step of the WY procedure. One of these tests compared results from our procedure to the WY procedure in the multtest package, and found it matched quite well.

Second, we also tested the WY procedures using the same simulation procedure described above. For constant effect and fixed effect models, the power estimation matches well between `PUMP` and the simulations. However, for random effects models, the two methods diverge in some cases. We find that the simulated power matches the `PUMP`-calculated power only when (1) there is a sufficiently large number of blocks/clusters, and (2) the user has a large number of WY permutations. Below, we discuss the single-step procedure because it is simpler, although the same concepts hold for the step-down procedure. Although it is difficult to verify why this discrepancy occurs, our hypothesis is that this behavior occurs due to the combination of the sensitivity of the WY procedure to departures from the true null distribution, and the instability of the random effects model. Below, we discuss the WY procedure in more detail to better understand why WY adjustment might not produce accurate power estimates in certain settings.

The goal of the WY procedure is to estimate the adjusted p-value for outcome i

$$\tilde{p}_i = Pr\left(\min_{1\leq j\leq k} P_j \leq p_i \mid H_0^C\right).$$

where P_j is a random variable representing a p-value for outcome j, and lowercase p_i is the observed realization for outcome i. We assume a set of k tests with corresponding null hypotheses H_{0i} for $i = 1, \cdots, k$. The complete null hypothesis is the setting where all the null hypotheses are true: $H_0^C = \cap_{i=1}^k H_{0i} = \{\text{all } H_i \text{ are true}\}$. In order to estimate this p-value, the p-value is calculated across B permutations

$$\tilde{p}_i = \frac{1}{B}\sum_{b=1}^{B} 1(\min_{1\leq j\leq k} p^\star_{b,j} \leq p_i).$$

In order for the procedure to be correct, we need to generate the permuted p-values $p^\star_{b,j}$ from the true distribution of p-values under the null hypothesis. However, if our estimation procedure is incorrect, we could be generating p-values that do not come from the true null distribution. For example, with a random effects model that does not converge or has a singular fit, the p-values from that model may be inaccurate.

For WY procedures, we are particularly concerned with discrepancies between the true and generated p-values that occur in the tails of the distribution. We generally are testing outcomes that are truly significant, so the observed p-values are small. Then, we are estimating a tail probability: is the observed p-value less than the minimum of the generated p-values over the multiple outcomes? At each permutation, we set a binary indicator for whether this occurs, so a small change in p-values can flip from an indicator of 0 to 1. Thus, a small discrepancy in the tails between the generated null distribution and the true distribution could have a large impact on the adjusted p-value.

Given that we are relying on tail behavior, this explains why we need both a sufficiently large number of permutations, and a sufficiently large number of blocks/clusters. The large number of permutations is required because we are trying to estimate a rare event. With a significant outcome, it is rare that the observed p-value will be less than the minimum of p-values generated from the null distribution. The large number of blocks/clusters is required in order for the model to generate accurate p-value estimates. With a small number of blocks/clusters, a random effects model has difficulty estimating the spread of the random effects distribution, which could result in inaccurate p-values.

Remark due to computational burden, for certain designs and models we only tested WY-SS and not WY-SD. In some cases, we had to reduce the number of simulations, number of permutations, or the number of units in order to run a validation that was not computationally intractable. We run a single scenario for each design and model, and do not test varying design and model parameters.

Discrepancies. After having adjusted all scenarios to have sufficiently large number of WY permutations and to have enough blocks/clusters, we find that for all scenarios the `PUMP` estimate falls within the simulation interval. We did not determine a strict rule for what is a sufficient sample size for stable WY estimates. For the `d3.1_m3rr2rr` context, we found that `J = 5` and `K = 5` produced a poor power estimate, but `J = 10` and `K = 10` produced a good estimate. For `d3.3_m3rc2rc`, `J = 10` and `K = 10` produced a poor estimate, while `J = 20` and `K = 20` produced a good estimate.

Point estimates. As with other MTPs, we find concordance between the point estimates for simulations and `PUMP`. The results can be found in Table 3.

Validating MDES and Sample size

For MDES and sample size calculations, we take a different approach. If the power calculation is correct is across all scenarios, then our MDES and sample size estimates should also be correct across all scenarios as long as the general procedure is correct. Thus, we choose one default scenario for each design and model and validate MDES and sample size for that single scenario. First, given a particular MDES, we use `PUMP` to estimate the power. Then, we reverse the procedure: we use `PUMP` to calculate the MDES given the power

d_m	MTP	mean.b.sim	max.b.sim
d2.1_m2fc	WY-SD	0.003	0.004
d2.1_m2fc	WY-SS	0.004	0.005
d2.1_m2ff	WY-SD	0.031	0.046
d2.1_m2ff	WY-SS	0.022	0.033
d2.1_m2fr	WY-SD	0.020	0.035
d2.1_m2fr	WY-SS	0.025	0.051
d2.2_m2rc	WY-SD	0.006	0.021
d2.2_m2rc	WY-SS	0.011	0.023
d3.1_m3rr2rr	WY-SD	0.026	0.037
d3.1_m3rr2rr	WY-SS	0.040	0.056
d3.2_m3ff2rc	WY-SS	0.048	0.066
d3.2_m3rr2rc	WY-SS	0.077	0.110
d3.3_m3rc2rc	WY-SS	0.018	0.036

Table 3: Point estimate results for power using WY procedures

estimate we just found. Finally, we check whether the output MDES is the same as the original input MDES. We follow the same approach for sample size validation, but repeat the process for each level relevant to that design (`nbar`, `J`, and `K` depending on design).

In Table , we provide an illustrative example of validation for a single scenario. The first column shows the calculated MDES, the middle column is the power we plugged into the calculation, and the last column shows the MDES that we are targeting. Thus, ideally we want the first and last columns to match.

MTP	Adjusted MDES	D1indiv Power	Target MDES
Bonferroni	0.122	0.447	0.125
BH	0.127	0.578	0.125
Holm	0.125	0.540	0.125

Table 4: MDES validation

PUMP mdes point estimates. We summarize the MDES performance by calculating the absolute difference between the `PUMP` estimate and the target MDES.

$$b_{mdes} = |\hat{m}_{pump} - \hat{m}_{target}|.$$

For MDES, we only run one scenario per design and model, and we take the mean across the different MTPs. The results are summarized below. We find that the `PUMP`-generated MDES is very close to the target MDES. The results can be found in Table 5.

Similarly, we validate our sample size calculations. Using our found power, we see if `pump_sample()` returns the original sample size. In Table , we are targeting a sample size of $J = 20$.

It is difficult to summarize sample size performance like we do for power and MDES. For many designs and models, a range of sample sizes may result in the same power. Thus, our procedure may produce a valid sample size that is very far off from the input sample size, and summarizing the difference is not illuminating here. For cases when the sample sizes have a large discrepancy, we plot the power curves to check that they are flat, and visually verify that the generated sample size has a similar power to the input sample size. These plots can be found in the individual validation documents for each scenario.

d_m	b_{mdes}
d2.1_m2fc	0.005
d2.1_m2ff	0.004
d2.1_m2fr	0.004
d2.2_m2rc	0.012
d3.1_m3rr2rr	0.005
d3.2_m3ff2rc	0.004
d3.2_m3rr2rc	0.006
d3.3_m3rc2rc	0.006

Table 5: Point estimate results for MDES

MTP	Sample.type	Sample.size	D1indiv.power
Bonferroni	J	21	0.500
BH	J	21	0.580
Holm	J	20	0.544

Table 6: Sample size validation

Summary of validation coverage results

d_m	MTP	power_type	cover
d2.1_m2fc	BF	D1indiv	1
d2.1_m2fc	BF	indiv.mean	1
d2.1_m2fc	BF	min1	1
d2.1_m2fc	BF	complete	1
d2.1_m2fc	BF	D2indiv	1
d2.1_m2fc	BF	D3indiv	1
d2.1_m2fc	BF	min2	1
d2.1_m2fc	BH	D1indiv	1
d2.1_m2fc	BH	indiv.mean	1
d2.1_m2fc	BH	min1	1
d2.1_m2fc	BH	complete	1
d2.1_m2fc	BH	D2indiv	1
d2.1_m2fc	BH	D3indiv	1
d2.1_m2fc	BH	min2	1
d2.1_m2fc	HO	D1indiv	1
d2.1_m2fc	HO	indiv.mean	1
d2.1_m2fc	HO	min1	1
d2.1_m2fc	HO	complete	1
d2.1_m2fc	HO	D2indiv	1
d2.1_m2fc	HO	D3indiv	1
d2.1_m2fc	HO	min2	1
d2.1_m2fc	None	D1indiv	1
d2.1_m2fc	None	indiv.mean	1
d2.1_m2fc	None	min1	–
d2.1_m2fc	None	complete	–
d2.1_m2fc	None	D2indiv	1
d2.1_m2fc	None	D3indiv	1
d2.1_m2fc	None	min2	–

d_m	MTP	power_type	cover
d2.1_m2ff	BF	D1indiv	1
d2.1_m2ff	BF	indiv.mean	1
d2.1_m2ff	BF	min1	1
d2.1_m2ff	BF	complete	1
d2.1_m2ff	BF	D2indiv	1
d2.1_m2ff	BF	D3indiv	1
d2.1_m2ff	BF	min2	1
d2.1_m2ff	BH	D1indiv	1
d2.1_m2ff	BH	indiv.mean	1
d2.1_m2ff	BH	min1	1
d2.1_m2ff	BH	complete	1
d2.1_m2ff	BH	D2indiv	1
d2.1_m2ff	BH	D3indiv	1
d2.1_m2ff	BH	min2	1
d2.1_m2ff	HO	D1indiv	1
d2.1_m2ff	HO	indiv.mean	1
d2.1_m2ff	HO	min1	1
d2.1_m2ff	HO	complete	1
d2.1_m2ff	HO	D2indiv	1
d2.1_m2ff	HO	D3indiv	1
d2.1_m2ff	HO	min2	1
d2.1_m2ff	None	D1indiv	1
d2.1_m2ff	None	indiv.mean	1
d2.1_m2ff	None	min1	–
d2.1_m2ff	None	complete	–
d2.1_m2ff	None	D2indiv	1
d2.1_m2ff	None	D3indiv	1
d2.1_m2ff	None	min2	–

d_m	MTP	power_type	cover
d2.1_m2fr	BF	D1indiv	0.875
d2.1_m2fr	BF	indiv.mean	0.875
d2.1_m2fr	BF	min1	0.750
d2.1_m2fr	BF	complete	0.875
d2.1_m2fr	BF	D2indiv	0.875
d2.1_m2fr	BF	D3indiv	0.875
d2.1_m2fr	BF	min2	0.750
d2.1_m2fr	BH	D1indiv	0.750
d2.1_m2fr	BH	indiv.mean	0.750
d2.1_m2fr	BH	min1	0.750
d2.1_m2fr	BH	complete	1.000
d2.1_m2fr	BH	D2indiv	0.875
d2.1_m2fr	BH	D3indiv	0.750
d2.1_m2fr	BH	min2	0.750
d2.1_m2fr	HO	D1indiv	0.750
d2.1_m2fr	HO	indiv.mean	0.750
d2.1_m2fr	HO	min1	0.750
d2.1_m2fr	HO	complete	1.000
d2.1_m2fr	HO	D2indiv	0.875
d2.1_m2fr	HO	D3indiv	0.750
d2.1_m2fr	HO	min2	0.750
d2.1_m2fr	None	D1indiv	0.750
d2.1_m2fr	None	indiv.mean	0.750
d2.1_m2fr	None	min1	–
d2.1_m2fr	None	complete	–
d2.1_m2fr	None	D2indiv	0.875
d2.1_m2fr	None	D3indiv	0.750
d2.1_m2fr	None	min2	–

d_m	MTP	power_type	cover
d2.2_m2rc	BF	D1indiv	1
d2.2_m2rc	BF	indiv.mean	1
d2.2_m2rc	BF	min1	1
d2.2_m2rc	BF	complete	–
d2.2_m2rc	BF	D2indiv	1
d2.2_m2rc	BF	D3indiv	1
d2.2_m2rc	BF	min2	1
d2.2_m2rc	BH	D1indiv	1
d2.2_m2rc	BH	indiv.mean	1
d2.2_m2rc	BH	min1	1
d2.2_m2rc	BH	complete	–
d2.2_m2rc	BH	D2indiv	1
d2.2_m2rc	BH	D3indiv	1
d2.2_m2rc	BH	min2	1
d2.2_m2rc	HO	D1indiv	1
d2.2_m2rc	HO	indiv.mean	1
d2.2_m2rc	HO	min1	1
d2.2_m2rc	HO	complete	–
d2.2_m2rc	HO	D2indiv	1
d2.2_m2rc	HO	D3indiv	1
d2.2_m2rc	HO	min2	1
d2.2_m2rc	None	D1indiv	1
d2.2_m2rc	None	indiv.mean	1
d2.2_m2rc	None	min1	–
d2.2_m2rc	None	complete	–
d2.2_m2rc	None	D2indiv	1
d2.2_m2rc	None	D3indiv	1
d2.2_m2rc	None	min2	–

d_m	MTP	power_type	cover
d3.1_m3rr2rr	BF	D1indiv	1
d3.1_m3rr2rr	BF	indiv.mean	1
d3.1_m3rr2rr	BF	min1	1
d3.1_m3rr2rr	BF	complete	–
d3.1_m3rr2rr	BF	D2indiv	1
d3.1_m3rr2rr	BF	D3indiv	1
d3.1_m3rr2rr	BF	min2	1
d3.1_m3rr2rr	BH	D1indiv	1
d3.1_m3rr2rr	BH	indiv.mean	1
d3.1_m3rr2rr	BH	min1	1
d3.1_m3rr2rr	BH	complete	–
d3.1_m3rr2rr	BH	D2indiv	1
d3.1_m3rr2rr	BH	D3indiv	1
d3.1_m3rr2rr	BH	min2	1
d3.1_m3rr2rr	HO	D1indiv	1
d3.1_m3rr2rr	HO	indiv.mean	1
d3.1_m3rr2rr	HO	min1	1
d3.1_m3rr2rr	HO	complete	–
d3.1_m3rr2rr	HO	D2indiv	1
d3.1_m3rr2rr	HO	D3indiv	1
d3.1_m3rr2rr	HO	min2	1
d3.1_m3rr2rr	None	D1indiv	1
d3.1_m3rr2rr	None	indiv.mean	1
d3.1_m3rr2rr	None	min1	–
d3.1_m3rr2rr	None	complete	–
d3.1_m3rr2rr	None	D2indiv	1
d3.1_m3rr2rr	None	D3indiv	1
d3.1_m3rr2rr	None	min2	–

d_m	MTP	power_type	cover
d3.2_m3ff2rc	BF	D1indiv	1
d3.2_m3ff2rc	BF	indiv.mean	1
d3.2_m3ff2rc	BF	min1	1
d3.2_m3ff2rc	BF	complete	–
d3.2_m3ff2rc	BF	D2indiv	1
d3.2_m3ff2rc	BF	D3indiv	1
d3.2_m3ff2rc	BF	min2	1
d3.2_m3ff2rc	BH	D1indiv	1
d3.2_m3ff2rc	BH	indiv.mean	1
d3.2_m3ff2rc	BH	min1	1
d3.2_m3ff2rc	BH	complete	–
d3.2_m3ff2rc	BH	D2indiv	1
d3.2_m3ff2rc	BH	D3indiv	1
d3.2_m3ff2rc	BH	min2	1
d3.2_m3ff2rc	HO	D1indiv	1
d3.2_m3ff2rc	HO	indiv.mean	1
d3.2_m3ff2rc	HO	min1	1
d3.2_m3ff2rc	HO	complete	–
d3.2_m3ff2rc	HO	D2indiv	1
d3.2_m3ff2rc	HO	D3indiv	1
d3.2_m3ff2rc	HO	min2	1
d3.2_m3ff2rc	None	D1indiv	1
d3.2_m3ff2rc	None	indiv.mean	1
d3.2_m3ff2rc	None	min1	–
d3.2_m3ff2rc	None	complete	–
d3.2_m3ff2rc	None	D2indiv	1
d3.2_m3ff2rc	None	D3indiv	1
d3.2_m3ff2rc	None	min2	–

d_m	MTP	power_type	cover
d3.2_m3rr2rc	BF	D1indiv	0.875
d3.2_m3rr2rc	BF	indiv.mean	0.875
d3.2_m3rr2rc	BF	min1	0.688
d3.2_m3rr2rc	BF	complete	0.875
d3.2_m3rr2rc	BF	D2indiv	0.875
d3.2_m3rr2rc	BF	D3indiv	0.875
d3.2_m3rr2rc	BF	min2	0.875
d3.2_m3rr2rc	BH	D1indiv	0.875
d3.2_m3rr2rc	BH	indiv.mean	0.875
d3.2_m3rr2rc	BH	min1	0.750
d3.2_m3rr2rc	BH	complete	0.875
d3.2_m3rr2rc	BH	D2indiv	0.875
d3.2_m3rr2rc	BH	D3indiv	0.875
d3.2_m3rr2rc	BH	min2	0.875
d3.2_m3rr2rc	HO	D1indiv	0.875
d3.2_m3rr2rc	HO	indiv.mean	0.875
d3.2_m3rr2rc	HO	min1	0.688
d3.2_m3rr2rc	HO	complete	0.875
d3.2_m3rr2rc	HO	D2indiv	0.875
d3.2_m3rr2rc	HO	D3indiv	0.875
d3.2_m3rr2rc	HO	min2	0.875
d3.2_m3rr2rc	None	D1indiv	0.875
d3.2_m3rr2rc	None	indiv.mean	0.875
d3.2_m3rr2rc	None	min1	–
d3.2_m3rr2rc	None	complete	–
d3.2_m3rr2rc	None	D2indiv	0.875
d3.2_m3rr2rc	None	D3indiv	0.875
d3.2_m3rr2rc	None	min2	–

d_m	MTP	power_type	cover
d3.3_m3rc2rc	BF	D1indiv	1
d3.3_m3rc2rc	BF	indiv.mean	1
d3.3_m3rc2rc	BF	min1	1
d3.3_m3rc2rc	BF	complete	–
d3.3_m3rc2rc	BF	D2indiv	1
d3.3_m3rc2rc	BF	D3indiv	1
d3.3_m3rc2rc	BF	min2	1
d3.3_m3rc2rc	BH	D1indiv	1
d3.3_m3rc2rc	BH	indiv.mean	1
d3.3_m3rc2rc	BH	min1	1
d3.3_m3rc2rc	BH	complete	–
d3.3_m3rc2rc	BH	D2indiv	1
d3.3_m3rc2rc	BH	D3indiv	1
d3.3_m3rc2rc	BH	min2	1
d3.3_m3rc2rc	HO	D1indiv	1
d3.3_m3rc2rc	HO	indiv.mean	1
d3.3_m3rc2rc	HO	min1	1
d3.3_m3rc2rc	HO	complete	–
d3.3_m3rc2rc	HO	D2indiv	1
d3.3_m3rc2rc	HO	D3indiv	1
d3.3_m3rc2rc	HO	min2	1
d3.3_m3rc2rc	None	D1indiv	1
d3.3_m3rc2rc	None	indiv.mean	1
d3.3_m3rc2rc	None	min1	–
d3.3_m3rc2rc	None	complete	–
d3.3_m3rc2rc	None	D2indiv	1
d3.3_m3rc2rc	None	D3indiv	1
d3.3_m3rc2rc	None	min2	–

Coverage discrepancies

We summarize below the scenarios where the simulation intervals do not cover the PUMP value. For brevity, we only display results for Bonferroni adjustments.

First, the scenarios themselves.

d_m	numZero	J	K	nbar	omega.2	omega.3	R2.1	R2.2	R2.3	ICC.2	ICC.3	rho
d2.1_m2fr	0	20	1	50	0.0	–	0.1	–	–	0.2	–	0.5
d2.1_m2fr	0	20	1	50	0.1	–	0.1	–	–	0.0	–	0.5
d3.2_m3rr2rc	0	30	10	50	–	0.0	0.1	0.1	–	0.2	0.2	0.5
d3.2_m3rr2rc	0	30	10	50	–	0.1	0.1	0.1	–	0.0	0.2	0.5
d3.2_m3rr2rc	0	30	10	50	–	0.1	0.1	0.1	–	0.2	0.0	0.5
d3.2_m3rr2rc	0	30	10	50	–	0.1	0.1	0.1	–	0.2	0.7	0.5
d3.2_m3rr2rc	0	30	10	50	–	0.1	0.1	0.6	–	0.2	0.2	0.5

Next, the power estimates from these scenarios.

d_m	MTP	type	omega.2	omega.3	ICC.2	ICC.3	R2.2	pump	sim	low	up
d2.1_m2fr	BF	D1indiv	0.0	–	0.2	–	–	0.38	0.31	0.25	0.38
d2.1_m2fr	BF	D2indiv	0.0	–	0.2	–	–	0.38	0.31	0.25	0.37
d2.1_m2fr	BF	D3indiv	0.0	–	0.2	–	–	0.39	0.31	0.25	0.37
d2.1_m2fr	BF	indiv.mean	0.0	–	0.2	–	–	0.38	0.31	0.25	0.37
d2.1_m2fr	BF	min1	0.0	–	0.2	–	–	0.63	0.55	0.49	0.61
d2.1_m2fr	BF	min2	0.0	–	0.2	–	–	0.36	0.28	0.22	0.34
d2.1_m2fr	BF	complete	0.0	–	0.2	–	–	0.34	0.28	0.21	0.34
d2.1_m2fr	BF	min1	0.1	–	0.0	–	–	0.51	0.44	0.38	0.50
d2.1_m2fr	BF	min2	0.1	–	0.0	–	–	0.26	0.20	0.13	0.26

d_m	MTP	type	omega.2	omega.3	ICC.2	ICC.3	R2.2	pump	sim	low	up
d3.2_m3rr2rc	BF	D1indiv	–	0.0	0.2	0.2	0.1	0.34	0.24	0.18	0.31
d3.2_m3rr2rc	BF	D2indiv	–	0.0	0.2	0.2	0.1	0.34	0.25	0.19	0.31
d3.2_m3rr2rc	BF	D3indiv	–	0.0	0.2	0.2	0.1	0.34	0.25	0.19	0.31
d3.2_m3rr2rc	BF	indiv.mean	–	0.0	0.2	0.2	0.1	0.34	0.25	0.19	0.31
d3.2_m3rr2rc	BF	min1	–	0.0	0.2	0.2	0.1	0.57	0.47	0.41	0.53
d3.2_m3rr2rc	BF	min2	–	0.0	0.2	0.2	0.1	0.31	0.21	0.15	0.27
d3.2_m3rr2rc	BF	complete	–	0.0	0.2	0.2	0.1	0.34	0.23	0.17	0.30
d3.2_m3rr2rc	BF	min1	–	0.1	0.0	0.2	0.1	0.66	0.73	0.67	0.79
d3.2_m3rr2rc	BF	D1indiv	–	0.1	0.2	0.0	0.1	0.33	0.23	0.17	0.29
d3.2_m3rr2rc	BF	D2indiv	–	0.1	0.2	0.0	0.1	0.33	0.25	0.18	0.31
d3.2_m3rr2rc	BF	D3indiv	–	0.1	0.2	0.0	0.1	0.32	0.24	0.18	0.30
d3.2_m3rr2rc	BF	indiv.mean	–	0.1	0.2	0.0	0.1	0.33	0.24	0.18	0.30
d3.2_m3rr2rc	BF	min1	–	0.1	0.2	0.0	0.1	0.57	0.45	0.39	0.51
d3.2_m3rr2rc	BF	min2	–	0.1	0.2	0.0	0.1	0.30	0.21	0.14	0.27
d3.2_m3rr2rc	BF	complete	–	0.1	0.2	0.0	0.1	0.33	0.23	0.17	0.29
d3.2_m3rr2rc	BF	min1	–	0.1	0.2	0.7	0.1	0.15	0.21	0.15	0.28
d3.2_m3rr2rc	BF	min1	–	0.1	0.2	0.2	0.6	0.46	0.54	0.48	0.60

Summary of validation "bias" results

d_m	MTP	power_type	mean.b.sim	mean.b.pow
d2.1_m2fc	BF	D1indiv	0.007	–
d2.1_m2fc	BF	indiv.mean	0.004	–
d2.1_m2fc	BF	min1	0.005	–
d2.1_m2fc	BF	complete	0.006	–
d2.1_m2fc	BF	D2indiv	0.006	–
d2.1_m2fc	BF	D3indiv	0.005	–
d2.1_m2fc	BF	min2	0.004	–
d2.1_m2fc	BH	D1indiv	0.004	–
d2.1_m2fc	BH	indiv.mean	0.006	–
d2.1_m2fc	BH	min1	0.006	–
d2.1_m2fc	BH	complete	0.007	–
d2.1_m2fc	BH	D2indiv	0.005	–
d2.1_m2fc	BH	D3indiv	0.007	–
d2.1_m2fc	BH	min2	0.005	–
d2.1_m2fc	HO	D1indiv	0.005	–
d2.1_m2fc	HO	indiv.mean	0.006	–
d2.1_m2fc	HO	min1	0.005	–
d2.1_m2fc	HO	complete	0.008	–
d2.1_m2fc	HO	D2indiv	0.005	–
d2.1_m2fc	HO	D3indiv	0.008	–
d2.1_m2fc	HO	min2	0.006	–
d2.1_m2fc	None	D1indiv	0.006	0.002
d2.1_m2fc	None	indiv.mean	0.005	–
d2.1_m2fc	None	D2indiv	0.006	–
d2.1_m2fc	None	D3indiv	0.006	–

d_m	MTP	power_type	mean.b.sim	mean.b.pow
d2.1_m2ff	BF	D1indiv	0.009	–
d2.1_m2ff	BF	indiv.mean	0.007	–
d2.1_m2ff	BF	min1	0.008	–
d2.1_m2ff	BF	complete	0.004	–
d2.1_m2ff	BF	D2indiv	0.009	–
d2.1_m2ff	BF	D3indiv	0.006	–
d2.1_m2ff	BF	min2	0.009	–
d2.1_m2ff	BH	D1indiv	0.006	–
d2.1_m2ff	BH	indiv.mean	0.005	–
d2.1_m2ff	BH	min1	0.007	–
d2.1_m2ff	BH	complete	0.004	–
d2.1_m2ff	BH	D2indiv	0.008	–
d2.1_m2ff	BH	D3indiv	0.007	–
d2.1_m2ff	BH	min2	0.005	–
d2.1_m2ff	HO	D1indiv	0.006	–
d2.1_m2ff	HO	indiv.mean	0.005	–
d2.1_m2ff	HO	min1	0.010	–
d2.1_m2ff	HO	complete	0.006	–
d2.1_m2ff	HO	D2indiv	0.006	–
d2.1_m2ff	HO	D3indiv	0.004	–
d2.1_m2ff	HO	min2	0.005	–
d2.1_m2ff	None	D1indiv	0.010	0.002
d2.1_m2ff	None	indiv.mean	0.010	–
d2.1_m2ff	None	D2indiv	0.010	–
d2.1_m2ff	None	D3indiv	0.004	–

d_m	MTP	power_type	mean.b.sim	mean.b.pow
d2.1_m2fr	BF	D1indiv	0.020	–
d2.1_m2fr	BF	indiv.mean	0.019	–
d2.1_m2fr	BF	min1	0.024	–
d2.1_m2fr	BF	complete	0.013	–
d2.1_m2fr	BF	D2indiv	0.022	–
d2.1_m2fr	BF	D3indiv	0.017	–
d2.1_m2fr	BF	min2	0.015	–
d2.1_m2fr	BH	D1indiv	0.011	–
d2.1_m2fr	BH	indiv.mean	0.011	–
d2.1_m2fr	BH	min1	0.016	–
d2.1_m2fr	BH	complete	0.020	–
d2.1_m2fr	BH	D2indiv	0.019	–
d2.1_m2fr	BH	D3indiv	0.012	–
d2.1_m2fr	BH	min2	0.010	–
d2.1_m2fr	HO	D1indiv	0.014	–
d2.1_m2fr	HO	indiv.mean	0.013	–
d2.1_m2fr	HO	min1	0.022	–
d2.1_m2fr	HO	complete	0.019	–
d2.1_m2fr	HO	D2indiv	0.022	–
d2.1_m2fr	HO	D3indiv	0.012	–
d2.1_m2fr	HO	min2	0.014	–
d2.1_m2fr	None	D1indiv	0.013	0.013
d2.1_m2fr	None	indiv.mean	0.011	–
d2.1_m2fr	None	D2indiv	0.021	–
d2.1_m2fr	None	D3indiv	0.010	–

d_m	MTP	power_type	mean.b.sim	mean.b.pow
d2.2_m2rc	BF	D1indiv	0.009	–
d2.2_m2rc	BF	indiv.mean	0.007	–
d2.2_m2rc	BF	min1	0.011	–
d2.2_m2rc	BF	complete	0.006	–
d2.2_m2rc	BF	D2indiv	0.006	–
d2.2_m2rc	BF	D3indiv	0.009	–
d2.2_m2rc	BF	min2	0.005	–
d2.2_m2rc	BH	D1indiv	0.007	–
d2.2_m2rc	BH	indiv.mean	0.007	–
d2.2_m2rc	BH	min1	0.009	–
d2.2_m2rc	BH	complete	0.008	–
d2.2_m2rc	BH	D2indiv	0.007	–
d2.2_m2rc	BH	D3indiv	0.007	–
d2.2_m2rc	BH	min2	0.006	–
d2.2_m2rc	HO	D1indiv	0.008	–
d2.2_m2rc	HO	indiv.mean	0.007	–
d2.2_m2rc	HO	min1	0.010	–
d2.2_m2rc	HO	complete	0.007	–
d2.2_m2rc	HO	D2indiv	0.007	–
d2.2_m2rc	HO	D3indiv	0.008	–
d2.2_m2rc	HO	min2	0.006	–
d2.2_m2rc	None	D1indiv	0.009	0.006
d2.2_m2rc	None	indiv.mean	0.006	–
d2.2_m2rc	None	D2indiv	0.006	–
d2.2_m2rc	None	D3indiv	0.008	–

d_m	MTP	power_type	mean.b.sim	mean.b.pow
d3.1_m3rr2rr	BF	D1indiv	0.008	–
d3.1_m3rr2rr	BF	indiv.mean	0.008	–
d3.1_m3rr2rr	BF	min1	0.013	–
d3.1_m3rr2rr	BF	complete	0.008	–
d3.1_m3rr2rr	BF	D2indiv	0.008	–
d3.1_m3rr2rr	BF	D3indiv	0.008	–
d3.1_m3rr2rr	BF	min2	0.006	–
d3.1_m3rr2rr	BH	D1indiv	0.007	–
d3.1_m3rr2rr	BH	indiv.mean	0.006	–
d3.1_m3rr2rr	BH	min1	0.011	–
d3.1_m3rr2rr	BH	complete	0.009	–
d3.1_m3rr2rr	BH	D2indiv	0.008	–
d3.1_m3rr2rr	BH	D3indiv	0.007	–
d3.1_m3rr2rr	BH	min2	0.005	–
d3.1_m3rr2rr	HO	D1indiv	0.007	–
d3.1_m3rr2rr	HO	indiv.mean	0.007	–
d3.1_m3rr2rr	HO	min1	0.013	–
d3.1_m3rr2rr	HO	complete	0.008	–
d3.1_m3rr2rr	HO	D2indiv	0.007	–
d3.1_m3rr2rr	HO	D3indiv	0.008	–
d3.1_m3rr2rr	HO	min2	0.006	–
d3.1_m3rr2rr	None	D1indiv	0.006	0.006
d3.1_m3rr2rr	None	indiv.mean	0.006	–
d3.1_m3rr2rr	None	D2indiv	0.004	–
d3.1_m3rr2rr	None	D3indiv	0.006	–

d_m	MTP	power_type	mean.b.sim	mean.b.pow
d3.2_m3ff2rc	BF	D1indiv	0.005	–
d3.2_m3ff2rc	BF	indiv.mean	0.003	–
d3.2_m3ff2rc	BF	min1	0.004	–
d3.2_m3ff2rc	BF	complete	0.004	–
d3.2_m3ff2rc	BF	D2indiv	0.005	–
d3.2_m3ff2rc	BF	D3indiv	0.004	–
d3.2_m3ff2rc	BF	min2	0.004	–
d3.2_m3ff2rc	BH	D1indiv	0.004	–
d3.2_m3ff2rc	BH	indiv.mean	0.003	–
d3.2_m3ff2rc	BH	min1	0.004	–
d3.2_m3ff2rc	BH	complete	0.004	–
d3.2_m3ff2rc	BH	D2indiv	0.003	–
d3.2_m3ff2rc	BH	D3indiv	0.005	–
d3.2_m3ff2rc	BH	min2	0.004	–
d3.2_m3ff2rc	HO	D1indiv	0.004	–
d3.2_m3ff2rc	HO	indiv.mean	0.004	–
d3.2_m3ff2rc	HO	min1	0.005	–
d3.2_m3ff2rc	HO	complete	0.005	–
d3.2_m3ff2rc	HO	D2indiv	0.005	–
d3.2_m3ff2rc	HO	D3indiv	0.005	–
d3.2_m3ff2rc	HO	min2	0.005	–
d3.2_m3ff2rc	None	D1indiv	0.004	0.007
d3.2_m3ff2rc	None	indiv.mean	0.003	–
d3.2_m3ff2rc	None	D2indiv	0.005	–
d3.2_m3ff2rc	None	D3indiv	0.005	–

d_m	MTP	power_type	mean.b.sim	mean.b.pow
d3.2_m3rr2rc	BF	D1indiv	0.019	–
d3.2_m3rr2rc	BF	indiv.mean	0.018	–
d3.2_m3rr2rc	BF	min1	0.031	–
d3.2_m3rr2rc	BF	complete	0.006	–
d3.2_m3rr2rc	BF	D2indiv	0.018	–
d3.2_m3rr2rc	BF	D3indiv	0.018	–
d3.2_m3rr2rc	BF	min2	0.015	–
d3.2_m3rr2rc	BH	D1indiv	0.018	–
d3.2_m3rr2rc	BH	indiv.mean	0.017	–
d3.2_m3rr2rc	BH	min1	0.030	–
d3.2_m3rr2rc	BH	complete	0.006	–
d3.2_m3rr2rc	BH	D2indiv	0.017	–
d3.2_m3rr2rc	BH	D3indiv	0.015	–
d3.2_m3rr2rc	BH	min2	0.012	–
d3.2_m3rr2rc	HO	D1indiv	0.022	–
d3.2_m3rr2rc	HO	indiv.mean	0.021	–
d3.2_m3rr2rc	HO	min1	0.031	–
d3.2_m3rr2rc	HO	complete	0.006	–
d3.2_m3rr2rc	HO	D2indiv	0.022	–
d3.2_m3rr2rc	HO	D3indiv	0.020	–
d3.2_m3rr2rc	HO	min2	0.018	–
d3.2_m3rr2rc	None	D1indiv	0.014	0.019
d3.2_m3rr2rc	None	indiv.mean	0.011	–
d3.2_m3rr2rc	None	D2indiv	0.013	–
d3.2_m3rr2rc	None	D3indiv	0.011	–

d_m	MTP	power_type	mean.b.sim	mean.b.pow
d3.3_m3rc2rc	BF	D1indiv	0.011	–
d3.3_m3rc2rc	BF	indiv.mean	0.011	–
d3.3_m3rc2rc	BF	min1	0.025	–
d3.3_m3rc2rc	BF	complete	0.014	–
d3.3_m3rc2rc	BF	D2indiv	0.011	–
d3.3_m3rc2rc	BF	D3indiv	0.012	–
d3.3_m3rc2rc	BF	min2	0.007	–
d3.3_m3rc2rc	BH	D1indiv	0.006	–
d3.3_m3rc2rc	BH	indiv.mean	0.005	–
d3.3_m3rc2rc	BH	min1	0.019	–
d3.3_m3rc2rc	BH	complete	0.013	–
d3.3_m3rc2rc	BH	D2indiv	0.007	–
d3.3_m3rc2rc	BH	D3indiv	0.005	–
d3.3_m3rc2rc	BH	min2	0.006	–
d3.3_m3rc2rc	HO	D1indiv	0.009	–
d3.3_m3rc2rc	HO	indiv.mean	0.009	–
d3.3_m3rc2rc	HO	min1	0.025	–
d3.3_m3rc2rc	HO	complete	0.013	–
d3.3_m3rc2rc	HO	D2indiv	0.010	–
d3.3_m3rc2rc	HO	D3indiv	0.008	–
d3.3_m3rc2rc	HO	min2	0.005	–
d3.3_m3rc2rc	None	D1indiv	0.007	0.011
d3.3_m3rc2rc	None	indiv.mean	0.006	–
d3.3_m3rc2rc	None	D2indiv	0.007	–
d3.3_m3rc2rc	None	D3indiv	0.007	–

WY Summary

d_m	MTP	mean.b.sim	max.b.sim
d2.1_m2fc	WY-SD	0.003	0.004
d2.1_m2fc	WY-SS	0.004	0.005
d2.1_m2ff	WY-SD	0.031	0.046
d2.1_m2ff	WY-SS	0.022	0.033
d2.1_m2fr	WY-SD	0.020	0.035
d2.1_m2fr	WY-SS	0.025	0.051
d2.2_m2rc	WY-SD	0.006	0.021
d2.2_m2rc	WY-SS	0.011	0.023
d3.1_m3rr2rr	WY-SD	0.026	0.037
d3.1_m3rr2rr	WY-SS	0.040	0.056
d3.2_m3ff2rc	WY-SS	0.048	0.066
d3.2_m3rr2rc	WY-SS	0.077	0.110
d3.3_m3rc2rc	WY-SS	0.018	0.036

MDES summary

MTP	Adjusted MDES	D1indiv Power	Target MDES	d_m
BF	0.125	0.475	0.125	d2.1_m2fc
BH	0.124	0.557	0.125	d2.1_m2fc
HO	0.126	0.552	0.125	d2.1_m2fc
BF	0.125	0.473	0.125	d2.1_m2ff
BH	0.125	0.567	0.125	d2.1_m2ff
HO	0.126	0.555	0.125	d2.1_m2ff
BF	0.125	0.266	0.125	d2.1_m2fr
BH	0.125	0.351	0.125	d2.1_m2fr
HO	0.124	0.318	0.125	d2.1_m2fr
BF	0.125	0.164	0.125	d2.2_m2rc
BH	0.125	0.209	0.125	d2.2_m2rc
HO	0.121	0.183	0.125	d2.2_m2rc
BF	0.125	0.721	0.125	d3.1_m3rr2rr
BH	0.127	0.842	0.125	d3.1_m3rr2rr
HO	0.125	0.810	0.125	d3.1_m3rr2rr
BF	0.124	0.522	0.125	d3.2_m3ff2rc
BH	0.125	0.624	0.125	d3.2_m3ff2rc
HO	0.126	0.610	0.125	d3.2_m3ff2rc
BF	0.125	0.155	0.125	d3.2_m3rr2rc
BH	0.125	0.222	0.125	d3.2_m3rr2rc
HO	0.127	0.199	0.125	d3.2_m3rr2rc
BF	0.249	0.211	0.250	d3.3_m3rc2rc
BH	0.251	0.284	0.250	d3.3_m3rc2rc
HO	0.247	0.250	0.250	d3.3_m3rc2rc

Collapsed Summaries

d_m	mean.b.sim	max.b.sim	mean.b.pow
d2.1_m2fc	0.006	0.008	0.002
d2.1_m2ff	0.007	0.010	0.002
d2.1_m2fr	0.016	0.024	0.013
d2.2_m2rc	0.007	0.011	0.006
d3.1_m3rr2rr	0.008	0.013	0.006
d3.2_m3ff2rc	0.004	0.005	0.007
d3.2_m3rr2rc	0.017	0.031	0.019
d3.3_m3rc2rc	0.010	0.025	0.011

d_m	mean.b.mdes
d2.1_m2fc	0.007
d2.1_m2ff	0.004
d2.1_m2fr	0.004
d2.2_m2rc	0.012
d3.1_m3rr2rr	0.005
d3.2_m3ff2rc	0.004
d3.2_m3rr2rc	0.006
d3.3_m3rc2rc	0.006