

Supplementary Materials

Additional file 1 — Stan code for the Bayesian multivariate hierarchical model

```
1 data {
2   int<lower=0> N; // number of observations
3   int<lower=0> D; // number of of binary outcomes
4   int<lower=2> L; // number of WHO categories
5   int<lower=0> P_main; // number of pre-treatment characteristics in the
      main effects term
6   int<lower=0> P_inter; // number of pre-treatment characteristics in the
      interaction effects term
7   int<lower=1,upper=L> y_ord[N]; // vector of ordinal outcomes
8   int<lower=0,upper=1> y_b[N,D]; // matrix of D binary outcomes (N x D matrix)
9   int<lower=0,upper=1> A[N]; // treatment or control
10  row_vector[P_main] x_main[N]; // pre-treatment characteristics in the main
      effects term (N x P_main matrix)
11  row_vector[P_inter] x_inter[N]; // pre-treatment characteristics in the
      interaction effects term (N x P_inter matrix)
12 }
13
14 parameters {
15   ordered[L-1] tau; // cut-points for cumulative odds model
16   vector<lower=0>[(P_inter + 1)] sigma_beta; // sd of outcome-specific treatment
      main effect and interaction effect
17   vector[D] beta_0; // outcome-specific intercepts for D binary
      outcomes
18   matrix[P_main,(D + 1)] beta_1; // covariates main effect for D binary outcomes
      and 1 ordinal outcome (P_main x (D + 1) matrix)
19   vector[(P_inter + 1)] beta_star; // pooled treatment main effect and pooled
      interaction effect across all outcomes
20
21   // non-central parameterization
22
23   matrix[(P_inter + 1),(D + 1)] z_beta_int;
24 }
25
26 transformed parameters {
27   matrix[(P_inter + 1),(D + 1)] beta_int; // outcome-specific treatment main effect
      and interaction effect ((P_inter + 1) x (D + 1) matrix)
28   vector[D] yhat_b[N];
29   real yhat_ord[N];
30
31   for (j in 1:(P_inter + 1))
32     for (k in 1:(D + 1)){
33       beta_int[j,k] = beta_star[j] + sigma_beta[j] * z_beta_int[j,k];
34     }
35
36   for (i in 1:N){
37     for (k in 1:D){
38       yhat_b[i,k] = beta_0[k] + x_main[i] * beta_1[,k] + (append_col(1, x_inter[
39         i]) * beta_int[,k]) * A[i];
40     }
41     yhat_ord[i] = x_main[i] * beta_1[,D+1] + (append_col(1, x_inter[i]) *
42       beta_int[,D+1]) * A[i];
43   }
44 }
45 model {
46
47   // priors
48
49   sigma_beta ~ exponential(1);
50   beta_star ~ normal(0,2.5);
```

```

51 to_vector(beta_1) ~ normal(0,2.5);
52 to_vector(z_beta_int) ~ std_normal();
53
54 for (l in 1:(L-1)){
55   tau[l] ~ student_t(3,0,8);
56 }
57
58   for (k in 1:D){
59     beta_0[k] ~ student_t(3,0,8);
60   }
61
62   // outcome model
63
64   for (i in 1:N){
65     y_ord[i] ~ ordered_logistic(yhat_ord[i], tau);
66     for (k in 1:D){
67       y_b[i,k] ~ bernoulli_logit(yhat_b[i,k]);
68     }
69   }
70 }

```