

SUPPLEMENTARY MATERIAL

Simulation Details: Factors and Levels

1. H_0 : treatment inferior, full compliance, 20% missing, MAR/MNAR, same/opposite, early/late separation trajectory, correlation=0.3/0.7. Total cases: 16
2. H_0 : treatment inferior, 80% compliance, 0% missing + (20% missing, MAR/MNAR, same/opposite), early/late separation trajectory, correlation=0.3/0.7. Total cases: 20
3. H_0 : treatment inferior, full and 80% compliance, missing range (5%, 50%), MAR/MNAR, same/opposite, early/late separation trajectory, correlation=0.3/0.7. Total cases: 32
4. H_1 : treatment non-inferior, full compliance, 20% missing, MAR/MNAR, same/opposite, same trajectories, correlation=0.3/0.7. Total cases: 8
5. H_1 : treatment non-inferior, 80% compliance, 0% missing + (20% missing, MAR/MNAR, same/opposite), same trajectories, correlation=0.3/0.7. Total cases: 10
6. H_1 : treatment non-inferior, full and 80% compliance, missing range (5%, 50%), MAR/MNAR, same/opposite, same trajectories, correlation=0.3/0.7. Total cases: 16

Simulation Details: Variance-covariance matrix, Σ_ρ , for data generation model with $\rho=0.3$ and 0.7.

$$\Sigma_{0.3} = \begin{pmatrix} 2.25 & 0.7875 & 0.30375 & 0.10125 \\ & 3.0625 & 1.18125 & 0.39375 \\ & & 5.0625 & 1.6875 \\ & & & 6.25 \end{pmatrix}$$
$$\Sigma_{0.7} = \begin{pmatrix} 2.25 & 1.8375 & 1.65375 & 1.28625 \\ & 3.0625 & 2.75625 & 2.14375 \\ & & 5.0625 & 3.9375 \\ & & & 6.25 \end{pmatrix}$$

SAS Code: Mixed Model for Repeated Measures

```
/* where the data are stored in long form */
/* the outcome y is collected at 4 times: 0,1,2,3 */
/* treat=1 for experimental treatment */
/* treat=0 for active control */
/* estimates change from baseline at final
time point difference between groups */
```

```

/* for NI test with alpha = 0.05 */

proc mixed data=dat_long;
class treat time;
model y=time treat time*treat/solution;
repeated time/ sub=id type=un;
estimate 'AminT_CFB' time*treat -1 0 0 1 1 0 0 -1/e cl alpha=0.1;
run;

```

SAS Code: Multiple Imputation

```

/* impute missing data in variables time2, time3 */
/* variable 'nc' is fully observed compliance indicator */

proc mi data=dat_wide out=out_mi;
class treat nc;
monotone regression;
var treat nc time0 time1 time2 time3;
run;

/* wide to long */
data temp1;
set out_mi;
id=_N_;
proc sort data=temp1;
by id treat _imputation_;
proc transpose data=temp1 out=temp2;
by id treat _imputation_;
data out_milong;
set temp2 (rename=(col1=y));
time=input(substr(_name_, 5), 5.);
drop _name_;
run;

/* run analysis model */
proc mixed data=out_milong;
class treat time;
model y=time treat time*treat/solution;
repeated time/ sub=id type=un;
by _imputation_;
estimate 'AminT_CFB' time*treat -1 0 0 1 1 0 0 -1/e cl alpha=.1;
ods output Estimates=ests;

```

```

/* save estimates and standard errors */
proc sort data=ests;
by _imputation_ label;

proc transpose data=ests out=ests_e prefix=e;
id label;
by k rep _imputation_;
var estimate;

proc transpose data=ests out=ests_s prefix=s;
id label;
by _imputation_;
var stderr;

data ests_wide(drop=_Label_);
merge ests_e(drop=_Name_) ests_s(drop=_Name_);
by _imputation_;

/* pool results */
proc mianalyze data=ests_wide;
modeleffects eAminT_CFB;
stderr sAminT_CFB;
run;

```

SAS Code: Multiple Imputation with MNAR Control Pattern

```

/* remove data at final timepoint if non-compliant */
data dat_wide;
set dat_wide;
if nc = 1 then time3 = '.';

proc sort data=dat_wide;
by treat;

/* impute missing data with compliers (nc=0) as control */
proc mi data=dat_wide out=out_mi;
class treat nc;
monotone regression;
mnar model(time2 time3 / modelobs=(nc='0'));

```

```

var treat time0 time1 time2 time3;

/* wide to long */
data temp1;
set out_mi;
id=_N_;

proc sort data=temp1;
by id treat _imputation_;

proc transpose data=temp1 out=temp2;
by id treat _imputation_;

data out_milong;
set temp2 (rename=(col1=y));
time=input(substr(_name_, 5), 5.);
drop _name_;
run;

/*
 * Run analysis model
 */
proc mixed data=out_milong;
class treat time;
model y=time treat time*treat/solution;
repeated time/ sub=id type=un;
by _imputation_;
estimate 'AminT_CFB' time*treat -1 0 0 1 1 0 0 -1/e cl alpha=.1;
ods output Estimates=ests;
run;

/* save estimates and standard errors */
proc sort data=ests;
by _imputation_ label;

proc transpose data=ests out=ests_e prefix=e;
id label;
by _imputation_;
var estimate;

proc transpose data=ests out=ests_s prefix=s;
id label;
by _imputation_;
var stderr;

```

```
data ests_wide(drop=_Label_);  
merge ests_e(drop=_Name_) ests_s(drop=_Name_);  
by _imputation_;  
  
/* pool results */  
proc mianalyze data=ests_wide alpha=0.1;  
modeleffects eAminT_CFB;  
stderr sAminT_CFB;  
run;
```

Supplementary Tables and Figures