```
/* This is a version of the SAS script used to conduct meta-analyses published in:
Hoeksema JD and SE Forde. 2008. A meta-analysis of factors affecting local adaptation
between interacting species. The American Naturalist 171:275-290. (DOI: 10.1086/527496)
*/

/* data step to read in data file and calculate several new variables for analysis */
data local_adapt;
infile /* insert path to a data file here, bracketed by 'single quotes' */
delimiter = ',' firstobs = 2;
input COMPARIS PAPER $ INTXNTYP $ HOSTTYPE $ Complex Hostcell Parcells GEN_SIMA GEN_SIMB
DISPSIMA DISPSIMB TAX_SIMA TAX_SIMB HOSTDISA HOSTDISB SYM_DISA SYM_DISB RESCUED $ TYPE1ALL
TYPE2ALL TYPE3ALL TYPE4SYM ALLOS SYMPS PAIRINGS HOSDEMES PARDEMES syminfSD2 aloinfSD2
symvirSD2 alovirSD2 PropNonrecip RECIP $ SHORT $ MAXALLO lnMAXALL REVVIR REVINFEC
SYMINF SYMINFSD SYMINFN ALO_INF ALOINFSD ALOINFN
SYMVIR SYMVIRSD SYMVIRN ALO_VIR ALOVIRSD ALOVIRN ;
/*Calculate log response ratio as effect size, and
multiply it by a reversal variable so that it always reflects parasite adaptation */
lnRRinf=(log(syminf/alo_inf))*revinfec;
lnRRvir=(log(symvir/alo_vir))*revvir;
/* Calculate the estimated within-study variance, and call it 'est' */
var=(((syminfsd)**2)/((syminfn)*((syminf)**2)))+(((aloinfsd)**2)/((aloinfn)*((alo_inf)**2)));
/*if syminfSD2 = '.' then delete;*/
var2=(((syminfSD2)**2)/((symps)*((syminf)**2)))+(((aloinfsd2)**2)/((allos)*((alo_inf)**2)));
if pairings = '.' then delete;
est=1/pairings; /* Here, the inverse of the number of sympatric plus allopatric pairings
is used as a proxy for the variance, and thus as the inverse weight for each study */
/* Calculate weighting variable as the reciprocal of the within-study variance estimate */
VarWt=1/est;
studyid=_n_;
run;
data full; /* full dataset */
set local_adapt;
if lnRRinf='.' then delete;
keep studyid paper lnRRinf lnRRvir est dispsimb hosttype gen_sima tax_sima complex recip lnmaxall comp
run;
data recip; /* only reciprocally-designed studies and rescued reciprocal data */
set full;
where RECIP = 'Yes';
studyid=_n_;
keep studyid paper lnRRinf lnRRvir est dispsimb hosttype gen_sima tax_sima complex;
run;
data parasites; /* does not contain studies on mutualisms */
set full;
where INTXNTYP = 'parasiti';
run;
data recip_par; /* only reciprocal studies, no studies on mutualisms */
set parasites;
where RECIP = 'Yes';
run;

/*Macros for conducting randomization tests to obtain p-values */
/*  indata = name of input dataset
    outdata = name you want to assign to the randomized output dataset
    depvar = the dependent effect-size variable
    var = the variance estimate that should be paired with each effect size estimate; this
        should usually be named 'est'
```

```
    numreps = the number of iterations for the randomization test; should be at least 1000
        and ideally 10000
    seed = the seed for the random number generator; zero gives a different seed every time
        while a non-zero integer will give the same set of random numbers each time
*/
%macro rand_gen(indata=,outdata=,depvar=,var=,numreps=,seed=);
/* Get size of input dataset into macro
variable &NUMRECS */
proc sql noprint;
select count(*) into :numrecs from
&INDATA;
quit;
/* Generate &NUMREPS random numbers for each
record, so records can be randomly
sorted within each replicate */
data __temp_1;
retain seed &SEED ; drop seed;
set &INDATA;
do replicate = 1 to &NUMREPS;
call ranuni(seed,rand_dep);
output;
end;
run;
proc sort data=__temp_1;
by replicate rand_dep;
run;
/* Now append the new re-orderings to the
original dataset. Label the original
as Replicate=0, so the %RAND_ANL macro
will be able to pick out the correct
p-value. Then use the ordering of
__counter within each replicate to
write the original values of &DEPVAR and &VAR,
thus creating a randomization of the
dependent variable and its corresponding variance in every replicate. */
data &OUTDATA ;
array deps{ &NUMRECS } $ _temporary_ ;
array vars{ &NUMRECS } $ _temporary_ ;
set &INDATA(in=in_orig)
__temp_1(drop=rand_dep);
if in_orig then do;
replicate=0;
deps{_n_} = &DEPVAR ;
end;
else
&DEPVAR = deps{ 1+ mod(_n_,&NUMRECS) };
if in_orig then do;
replicate=0;
vars{_n_} = &VAR ;
end;
else
&VAR = vars{ 1+ mod(_n_,&NUMRECS) };
run;
%mend rand_gen;
%macro rand_anl(randdata=,where=,teststat=,testlabel=);
data _null_;
```

```
retain observed numsig numtot 0;
set &RANDDATA end=endofile;
%if "&WHERE" ne ""
%then where &WHERE %str(;) ;
if Replicate=0 then observed = &TESTSTAT ;
else do;
numtot+1;
numsig + ( &TESTSTAT >= observed ); /* Use >= when &TESTSTAT is a test statistic such as SS or F */
end;                              /* but use <= when &TESTSTAT is the p-value itself */
/* Also, when using p-value as the test stat, should round all p-values, or multiply them */
/* by a large number, so that numerical operators perform accurately in SAS */
/* e.g. very small p-values can sometimes be inaccurately compared */
if endofile then do;
ratio = numsig/numtot;
put "Randomization test for &TESTLABEL "
%if "&WHERE" ne "" %then "where &WHERE";
" has significance level of "
ratio 12.10;
end;
run;
%mend rand_anl;

/* Mixed-effects maximum likelihood model, full dataset, species traits main effects */
/* modified from Section 5 of van Houwelingen et al. 2002:
van Houwelingen, H. C., L. R. Arends, and T. Stijnen. 2002. Advanced methods in meta-analysis:
Multivariate approach and meta-regression. Statistics in Medicine 21:589-624.
*/

data betvar; /* gives a starting value of 0.05 for the between-studies variance component */
infile cards;
input est;
cards;
0.05
;
run;
data vars;
set full;
keep est;
run;
proc append base=betvar data=vars;
run;
proc sql noprint;
select count(*)
   into :nobs
   from betvar;
quit;

/*obtains estimate of between-studies variance component, and overall weighted mean
effect size, using maximum likelihood*/
proc mixed cl method=ml data=full ;
parms / parmsdata=betvar eqcons=2 to &nobs;
class studyid;
model lnRRinf=  / s cl;
random int / subject=studyid;
repeated / group=studyid;
run;
```

```
/* Mixed-effects meta-regression model */
proc mixed cl method=ml data=full ;
parms / parmsdata=betvar eqcons=2 to &nobs;
class studyid dispsimb hosttype;
model lnRRinf=dispsimb hosttype gen_sima tax_sima complex /
s cl covb ddf=1000,1000,1000,1000,1000 solution;
random int/ subject=studyid s;
repeated / group=studyid;
run;
quit;

/* invoke macro to obtain randomized dateset for randomization tests */
%rand_gen(indata=full,outdata=outrand,depvar=lnRRinf,var=est,numreps=10,seed=0);
proc sort data=outrand;
by replicate studyid;
run;

/* create dataset called randvars for use with parmdata, during randomization iterations
from mixed model below*/

data randvars;
set outrand;
where studyid = 1;
est=0.05;
order=0;
keep order replicate est;
run;
data startvars;
set outrand;
order = _n_;
keep order replicate est;
run;
proc append base=randvars data=startvars;
run;
proc sort data=randvars;
by replicate order;
run;
data randvars;
set randvars;
drop order;
run;

/*Randomization tests to obtain p-values from mixed-effects max likelihood model */
ods output Tests3=Tests;
ods listing close;
options nonotes;

proc mixed cl method=ml data=outrand;
by replicate;
parms / parmsdata=randvars eqcons=2 to &nobs;
class studyid dispsimb hosttype;
model lnRRinf=dispsimb hosttype gen_sima tax_sima complex/ s cl covb ddf=1000,1000,1000,1000,1000;
random int/ subject=studyid s;
repeated / group=studyid;
run;
```

```
ods output close;
ods listing;
options notes;

%rand_anl(randdata=Tests,where=Effect='DISPSIMB',
teststat=FValue,testlabel= test of significance );
%rand_anl(randdata=Tests,where=Effect='HOSTTYPE',
teststat=FValue,testlabel= test of significance );
%rand_anl(randdata=Tests,where=Effect='GEN_SIMA',
teststat=FValue,testlabel=test of significance );
%rand_anl(randdata=Tests,where=Effect='TAX_SIMA',
teststat=FValue,testlabel=test of significance );
%rand_anl(randdata=Tests,where=Effect='Complex',
teststat=FValue,testlabel=test of significance );

quit;

/* Mixed-effects meta-regression model, full dataset, effects of experimental design */
proc mixed cl method=ml data=full ;
parms / parmsdata=betvar eqcons=2 to &nobs;
class studyid recip;
model lnRRinf=recip lnmaxall / s cl covb ddf=1000,1000;
random int/ subject=studyid s;
repeated / group=studyid;
run;
quit;

/*Randomization tests to obtain p-values from mixed-effects max likelihood model */
ods output Tests3=TestsB;
ods listing close;
options nonotes;

proc mixed cl method=ml data=outrand;
by replicate;
parms / parmsdata=randvars eqcons=2 to &nobs;
class studyid recip;
model lnRRinf=recip lnmaxall / s cl covb ddf=1000,1000;
random int/ subject=studyid s;
repeated / group=studyid;
run;

ods output close;
ods listing;
options notes;
quit;

%rand_anl(randdata=TestsB,where=Effect='RECIP',
teststat=FValue,testlabel= test of significance );
%rand_anl(randdata=TestsB,where=Effect='lnMAXALL',
teststat=FValue,testlabel= test of significance );
quit;

/*Mixed-effects model using maximum likelihood, reciprocal dataset, species trait effects */
/* modified from van Houwelingen et al. 2002 Sect 5 */
```

```
data betvar2; /* gives a starting value of 0.05 for the between-studies variance component */
infile cards;
input est;
cards;
0.05
;
run;
data vars2;
set recip;
keep est;
run;
proc append base=betvar2 data=vars2;
run;
proc sql noprint;
select count(*)
   into :nobs
   from betvar2;
quit;

/*obtain estimate of between-studies variance component, and overall weighted mean
effect size, using maximum likelihood*/
proc mixed cl method=ml data=recip;
parms / parmsdata=betvar2 eqcons=2 to &nobs;
class studyid;
model lnRRinf=  / s cl;
random int / subject=studyid;
repeated / group=studyid;
run;

/* Mixed-effects meta-regression model using max likelihood */

proc mixed cl method=ml data=recip ;
parms / parmsdata=betvar2 eqcons=2 to &nobs;
class studyid dispsimb hosttype;
model lnRRinf=dispsimb hosttype gen_sima tax_sima complex / s cl covb ddf=1000,1000,1000,1000,1000;
random int/ subject=studyid s;
repeated / group=studyid;
run;
quit;

/* randomization tests for above model */

%rand_gen(indata=recip,outdata=outrand2,depvar=lnRRinf,var=est,numreps=10,seed=0);
proc sort data=outrand2;
by replicate studyid;
run;

/* create dataset called randvars2 for use with parmdata, during randomization iterations
from mixed model below*/

data randvars2;
set outrand2;
where studyid = 1;
est=0.05;
order=0;
keep order replicate est;
```

```
run;
data startvars2;
set outrand2;
order = _n_;
keep order replicate est;
run;
proc append base=randvars2 data=startvars2;
run;
proc sort data=randvars2;
by replicate order;
run;
data randvars2;
set randvars2;
drop order;
run;

/* Randomization tests to obtain p-values from mixed-effects max likelihood model*/

ods output Tests3=Tests_b;
ods listing close;
options nonotes;

proc mixed method=ml data=outrand2;
by replicate;
parms / parmsdata=randvars2 eqcons=2 to &nobs;
class studyid dispsimb hosttype;
model lnRRinf=dispsimb hosttype gen_sima tax_sima complex/ s cl covb ddf=1000,1000,1000,1000,1000;
random int/ subject=studyid s;
repeated / group=studyid;
run;

ods output close;
ods listing;
options notes;

%rand_anl(randdata=Tests_b,where=Effect='DISPSIMB',
teststat=FValue,testlabel= test of significance );
%rand_anl(randdata=Tests_b,where=Effect='HOSTTYPE',
teststat=FValue,testlabel= test of significance );
%rand_anl(randdata=Tests_b,where=Effect='GEN_SIMA',
teststat=FValue,testlabel=test of significance );
%rand_anl(randdata=Tests_b,where=Effect='TAX_SIMA',
teststat=FValue,testlabel=test of significance );
%rand_anl(randdata=Tests_b,where=Effect='Complex',
teststat=FValue,testlabel=test of significance );

quit;
```