

```

1 % N census from Hypergeom of biallelic haploid SNP alleles, full spectrum
2 % Based on NcHyper230810RandP2pass.m, spectrum of p each locus
3 % Simulates sampling inds&loci, uses jackknifed subsamples
4 % HypExp from HypExactExp230607.m, test (O-Ehyp) scaled 0-1 each locus
5 filename=input('outfile=', 's');
6 %%%%%%%%%%%%%%% Later make loops for samp etc
7 % Jackknife Sampling
8 samp=input('Largest fraction to sample, eg 0.1 to 0.9 = ');
9 jj=input('Prop of sample in each subsamp = ');
10
11 % Increments Ptry, Ntry (hyps. for joint est. of Nc & p)
12 % 11 possible Ncensus values to trial (hyps for joint estimate Nc & p)
13 minNtry=input('minNtry to hypothesise '); maxNtry=3*minNtry;
14 Ninc=round((maxNtry-minNtry)/10); % fine increment, to give 11 Ntry values
15 Ntry=[minNtry:Ninc:maxNtry]; % array of hypothetical Nc values to trial
16 disp('N values to try'); Ntry
17 % Actual Nc value
18 Nit=input('Real Nit integer b/n min&maxN = ');
19
20 % actual p value, called 'pp' for each locus, adjust to 'p' later
21 ppinc=0.05;% Increment from one pp value to next
22 ppcutter=[ppinc:ppinc:1-ppinc]; % the values
23 ppreps=10; % number of loci with each pp value
24 pp= sort(repmat(ppcutter,[1,ppreps])); % one value for each locus
25 L=length(pp); % Number of loci, index 'l'
26 for l=1:L
27 target=round((pp(l))*minNtry); p(l)=target/minNtry;%integral nbr targets
28 end
29
30 % hypothetical p values, Ptry; not same as p(l) above, esp. after jackknife
31 Ptry=[ppinc:ppinc:1-ppinc];
32
33 %% Simulate Pop size Nit: individuals in population, each locus
34 preindit=zeros(L,Nit); % Rows Loci, Cols Inds
35 indit=zeros(L,Nit); % Rows Loci, Cols Inds
36 for l=1:L % popsetup-loci 1:L(row),inds (col - Nit of these)
37     targetit=round(p(l).*Nit); % nbr. target alleles in actual pop., locus 'l'
38     preindit(l,1:targetit)=1; % set target inds to unity, rest still zero
39     tempit=preindit(l,:);%vector of inds for locus 'l', length Nit
40     idx = randperm(length(tempit));%idx:vec of shuffled indices of 'tempit'
41     tempitperm = tempit(idx);%chooses inds (0 or 1) from 'tempit'
42     indit(l,:)=tempitperm;%adds to matrix of L loci(row)* individs (col)
43 end % End locus loop to simulate population size Nit, L loci
44
45 % Loop for Ntry values, setting adaptive dimensions
46 %NB ensures n<Nit or n<Ntry, whichever is smaller
47 for Nindex=1:(length(Ntry)); % loop trial N's each locus, adaptive sampling
48     %Set sample size n,jackknife subsampsize j(Nindex)
49     if Nit<Ntry(Nindex);
50         n(Nindex)=round(Nit*samp); j(Nindex)=round(n(Nindex)*jj);
51     else;
52         n(Nindex)=round(samp*Ntry(Nindex)); j(Nindex)=round(n(Nindex)*jj);
53     end; % end adaptive sampling
54     jmax(Nindex)=round(jj*round(maxNtry*samp)); % Common DIM all O & E vectors

```

```

55 end; % end Ntry loop for setting adaptive dimensions
56 % Fixed DIMS, zero fill to jmaxALL, for all "l", "Ntry".
57 jmaxALL=max(jmax);
58 HypExp=zeros(length(Ntry),length(Ptry),jmaxALL+1);
59 propdetNit=zeros(L,jmaxALL+1);
60 BIAS1=zeros(length(Ntry),length(Ptry),L);% BIAS for each locus
61 % Loop for Ntry values, calc Hyp Exps for N,n,p
62 for Nindex=1:(length(Ntry)); % loop trial N's each locus,
63     for pindex=1:(length(Ptry)); %loop Hyp Exp in sample of 'j', iTry230831n N✓
loop
64         parg=Ptry(pindex);Narg=Ntry(Nindex);narg=j(Nindex);%Arguments for HypExp✓
next
65         [HypExpTEMP1(1:(j(Nindex)+1))]=HypExactExp230607(parg,Narg,narg);
66         HypExp(Nindex,pindex,(1:j(Nindex)+1))=HypExpTEMP1;%filled to jmax with zeros
67     end; % end loop for Hyp Exp
68 end; % end Loop for Ntry values, calc Hyp Exps for N,n,p
69
70 % Main Loop for Ntry values
71 for Nindex=1:(length(Ntry)); % loop trial N's each locus,
72     % Bin edges to tally targets-detected, increment 1; -0.5 to +(n+0.5)
73     counter1=1;
74     for e=-0.5:1:(j(Nindex)+0.5);%"j(Nindex)+1" bins, centres 0...j(Nindex)
75         edges(counter1)=e;
76         counter1=counter1+1;
77     end; % end e loop
78
79     %% Major Locus loop begins
80 %% Jackknife sample actual population size Nit
81 for l=1:L; %sample loci Nit. Inds random in locus*popsize matrix
82     %ploc(l)=(sum(indit(l,1:n)))/n;%allele prop locus "l", from sample of n
83     for jndx=1:(n(Nindex)-j(Nindex)+1);% overlap jac of j(Nindex),loc'l'
84         inditTEMP(l,(1:j(Nindex)))=indit(l,(jndx:(jndx+j(Nindex)-1)));
85         detNit(l,jndx)=sum(inditTEMP(l,1:j(Nindex)));%TrgtSampd row-loc,col-jack
86     end; % end loop for 'n-j(Nindex)+1' subsamples each sample size j(Nindex)
87 end; % end loop for sampling over loci 'l'
88
89 % Tally results in 'freqdetNit', popsize Nit, & each locus
90 for l=1:L % loop loci for histogram data.
91     HistFreq=histcounts(detNit(l,:),edges);%in n-j(Nindex)+1 Jackknife reps
92 % row vec, bins number detected 0,1,...,j(Nindex)+1
93 PropNit=HistFreq./sum(HistFreq);
94 propdetNit(l,(1:j(Nindex)+1))=PropNit;%filled to jmaxALL with zeros
95 end; % end target-det; dims: locus, bins (0's,1's,...n detected)
96 end; % end sampling fot that Ntry array
97
98 %% BIAS for each locus, & each hypothesised Ntry, Ptry
99 for l=1:L; % locus loop, each needs to examine all Ntrys
100     for Nindex=1:(length(Ntry)); % loop trial N's each locus,
101         for pindex=1:(length(Ptry)); % begin joint estimate p & Nc, loop p values
102             % AbsBias, Absolute value:Obs-HypergeoExps, n-1 inds (count 0,1,...n)
103             HypExpTEMP2=(zeros(1,jmaxALL+1))';
104             aa=HypExp(Nindex,pindex,(1:jmaxALL+1));
105             HypExpTEMP2=(squeeze(aa))'; % row vec length j(Nindex)
106             PropTEMP=propdetNit;

```

```

107     BiasTEMP1=abs(PropTEMP-HypExpTEMP2);%fill to jmax with 0
108     BiasTEMP2=sum(BiasTEMP1(1,:)); % abs bias over all bins, this locus
109     BIAS1(Nindex,pindex,1)=BiasTEMP2;% AbsBias all bins, this locus
110     end; % end pindex loop
111 end; % End loop for simulating and testing Ntry values.
112 end; %end locus loop.
113
114 % Joint Pest-Nest diagnosis each locus, then Nest averaged over all loci
115 %Diagnose p,N this locus, search min BIAS1, index gives Ptry Ntry indices
116 for l=1:L;
117 locBIAS=squeeze(BIAS1(:,:,l)); %bias for this locus (3rd dim)
118 Min=min(locBIAS,[],"all");
119 [row,col]=find(locBIAS==Min); % shows row and col of min value
120 Nest(l)=mean(Ntry(row)); Pest(l)=mean(Ptry(col));% Av if>1 val
121 end; % end locus loop
122 % Over loci, AVE and SE of Nest
123 AveNest=mean(Nest); SeNest=(std(Nest))/sqrt(L);
124
125 % For investigating loci where Pest Biased up so Nest Biased down.
126 NestBias(:)=(Nest(:)-Nit)./Nit;
127 PestBias(:)=(Pest(:)-p(:))./p(:);
128 SDdetNit=(std(detNit,1,2))';
129 AvedetNit=(mean(detNit,2))';
130 CVdetNit=((SDdetNit(:))./(AvedetNit(:)))';
131 % InvCVdetNit=((AvedetNit(:))./(SDdetNit(:)))';
132
133 %% Now do Nest diagnosis with low CV loci only
134 LocNbr(1,1:L)=[1:L];% serialnumbers for loci
135 CVNest=([CVdetNit(1:L);Nest(1:L);LocNbr(1:L)])';%CVcol1,NestCol2;LocNbrCol3
136 CVNestSort=(sortrows(CVNest,1))';%sorts by CV, all rows (CV,Nest,LocNbr)
137 NestLoCV=CVNestSort(2,(1:round(L/10)));%Nest row,lowest 10% of CVs
138 CVLoLoci=CVNestSort(3,(1:round(L/10)));% locus numbers used in AveNestLoCV
139 AveNestLoCV=mean(NestLoCV);
140 SDNestLoCV=std(NestLoCV);
141
142 % Plot NestBias PestBias vs CV,skew, etc of detNit(1,jndx)
143 subplot(2,2,1); % NestBias vs CVdetNit
144 xlabel('CVdetNit');
145 ylabel('NestBias');
146 pl=plot(CVdetNit,NestBias);
147
148 subplot(2,2,2); % PestBias vs CVdetNit
149 xlabel('CVdetNit');
150 ylabel('PestBias');
151 pl=plot(CVdetNit,PestBias);
152
153 %subplot(2,2,3); % NestBias vs InvCVdetNit
154 %xlabel('InvCVdetNit');
155 %ylabel('NestBias');
156 %pl=plot(InvCVdetNit,NestBias);
157
158 %subplot(2,2,4); % PestBias vs InvCVdetNit
159 %xlabel('InvCVdetNit');
160 %ylabel('PestBias');

```

```

161 %p1=plot(InvCVdetNit,PestBias);
162 %% PLOTTING
163 % SumBIAS2=squeeze(sum(BIAS2,3)); % Sum bias over all loci (3rd dim)
164 % MeanBIAS2=SumBIAS2./L; % Mean bias over all loci, for plotting
165 % Y=Ntry2; X=Ptry2; % Axes for contour plot
166 % contour(X,Y,MeanBIAS2,'showtext','on');
167
168 save(
(filename,'p','L','Nit','Ntry','maxNtry','Ptry','samp','jj','n','j','indit','propdet
Nit','HypExp','BIAS1','Nest','Pest','AveNest','SeNest','CVNest','CVLoLoci','AveNestL
oCV','SDNestLoCV','NestLoCV','CVLoLoci','AveNestLoCV','SDNestLoCV')
169 clearvars -except filename
170
171 %% PASS2: Uses the alternative allele
172 load(filename,'p','L','Nit','Ntry','maxNtry','Ptry','samp','jj','indit')
173 inditTEMP=indit;
174 indit=(inditTEMP-1).*(-1);% swaps zeros and ones
175
176 % Loop for Ntry values, setting adaptive dimensions
177 %NB ensures n<Nit or n<Ntry, whichever is smaller
178 for Nindex=1:(length(Ntry)); % loop trial N's each Fpestlocus, adaptive sampling
179     %Set sample size n,jackknife subsample size j(Nindex)
180     if Nit<Ntry(Nindex);
181         n(Nindex)=round(Nit*samp); j(Nindex)=round(n(Nindex)*jj);
182     else;
183         n(Nindex)=round(samp*Ntry(Nindex)); j(Nindex)=round(n(Nindex)*jj);
184     end; % end adaptive sampling
185     jmax(Nindex)=round(jj*round(maxNtry*samp)); % Common DIM all O & E vectors
186 end; % end Ntry loop for setting adaptive dimensions
187 % Fixed DIMS, zero fill to jmaxALL, for all "l", "Ntry".
188 jmaxALL=max(jmax);
189 HypExp=zeros(length(Ntry),length(Ptry),jmaxALL+1);
190 propdetNit=zeros(L,jmaxALL+1);
191 BIAS1=zeros(length(Ntry),length(Ptry),L);% BIAS for each locus
192 % Loop for Ntry values, calc Hyp Exps for N,n,p
193 for Nindex=1:(length(Ntry)); % loop trial N's each locus,
194     for pindex=1:(length(Ptry)); %loop Hyp Exp in sample of 'j', iTry230831n N
loop
195         parg=Ptry(pindex);Narg=Ntry(Nindex);narg=j(Nindex);%Arguments for HypExp
next
196         [HypExpTEMP1(1:(j(Nindex)+1))]=HypExactExp230607(parg,Narg,narg);
197         HypExp(Nindex,pindex,(1:j(Nindex)+1))=HypExpTEMP1;%fill jmaxALL+1 with0
198     end; % end loop for Hyp Exp
199 end; % end Loop for Ntry values, calc Hyp Exps for N,n,p
200
201 % Main Loop for Ntry values
202 for Nindex=1:(length(Ntry)); % loop trial N's each locus,
203     % Bin edges to tally targets-detected, increment 1; -0.5 to +(n+0.5)
204     counter1=1;
205     for e=-0.5:1:(j(Nindex)+0.5);%"j(Nindex)+1" bins, centres 0...j(Nindex)
206         edges(counter1)=e;
207         counter1=counter1+1;
208     end; % end e loop
209

```

```

210     %% Major Locus loop begins
211 %% Jackknife sample actual population size Nit
212 for l=1:L; %sample loci Nit. Inds random in locus*popsize matrix
213     %ploc(l)=(sum(indit(l,1:n)))/n;%allele prop locus "l", from sample of n
214     for jndx=1:(n(Nindex)-j(Nindex)+1);% overlap jac of j(Nindex),loc'l'
215         inditTEMP(l,(1:j(Nindex)))=indit(l,(jndx:(jndx+j(Nindex)-1)));
216         detNit(l,jndx)=sum(inditTEMP(l,1:j(Nindex)));%TrgtSampd row-loc,col-jack
217     end; % end loop for 'n-j(Nindex)+1' subsamples each sample size j(Nindex)
218 end; % end loop for sampling over loci 'l'
219
220 % Tally results in 'freqdetNit', popsize Nit, & each locus
221 for l=1:L % loop loci for histogram data.
222     HistFreq=histcounts(detNit(l,:),edges);%in n-j(Nindex)+1 Jackknife reps
223     % row vec, bins number detected 0,1,...,j(Nindex)+1
224     PropNit=HistFreq./sum(HistFreq);
225     propdetNit(l,(1:j(Nindex)+1))=PropNit;%filled to jmaxALL+1 with zeros
226 end; % end target-det; dims: locus, bins (0's,1's,...n detected)
227 end; % end sampling fot that Ntry array
228
229 %% BIAS for each locus, & each hypothesised Ntry, Ptry
230 for l=1:L; % locus loop, each needs to examine all Ntrys
231     for Nindex=1:(length(Ntry)); % loop trial N's each locus,
232         for pindex=1:(length(Ptry)); % begin joint estimate p & Nc, loop p values
233             % AbsBias, Absolute value:Obs-HypergeoExps, n-1 inds (count 0,1,...n)
234             HypExpTEMP2=(zeros(1,jmaxALL))';
235             aa=HypExp(Nindex,pindex,(1:jmaxALL+1));
236             HypExpTEMP2=(squeeze(aa))'; % row vec length j(Nindex)
237             PropTEMP=propdetNit;
238             BiasTEMP1=abs(PropTEMP-HypExpTEMP2);%fill to jmax with 0
239             BiasTEMP2=sum(BiasTEMP1(l,:)); % abs bias over all bins, this locus
240             BIAS1(Nindex,pindex,l)=BiasTEMP2;% AbsBias all bins, this locus
241         end; % end pindex loop
242     end; % End loop for simulating and testing Ntry values.
243 end; %end locus loop.
244
245 % Joint Pest-Nest diagnosis each locus, then Nest averaged over all loci
246 %Diagnose p,N this locus, search min BIAS1, index gives Ptry Ntry indices
247 for l=1:L;
248     locBIAS=squeeze(BIAS1(:, :, l)); %bias for this locus (3rd dim)
249     Min=min(locBIAS, [], "all");
250     [row,col]=find(locBIAS==Min); % shows row and col of min value
251     Nest(l)=mean(Ntry(row)); Pest(l)=mean(Ptry(col));% Av if>1 val
252 end; % end locus loop
253 % Over loci, AVE and SE of Nest
254 AveNest=mean(Nest); SeNest=(std(Nest))/sqrt(L);
255
256 % For investigating loci where Pest Biased up so Nest Biased down.
257 NestBias(:)=(Nest(:)-Nit)./Nit;
258 PestBias(:)=(Pest(:)-p(:))./p(:);
259 SDdetNit=(std(detNit,1,2))';
260 AvedetNit=(mean(detNit,2))';
261 CVdetNit=((SDdetNit(:))./(AvedetNit(:)))';
262 % InvCVdetNit=((AvedetNit(:))./(SDdetNit(:)))';
263

```

```

264 %% Now do Nest diagnosis with low CV loci only
265 LocNbr(1,1:L)=[1:L];% serialnumbers for loci
266 CVNest=([CVdetNit(1:L);Nest(1:L);LocNbr(1:L)])';%CVcol1,NestCol2;LocNbrCol3
267 CVNestSort= (sortrows(CVNest,1))';%sorts by CV, all rows (CV,Nest,LocNbr)
268 NestLoCV=CVNestSort(2,(1:round(L/10)));%Nest row,lowest 10% of CVs
269 CVLoLoci=CVNestSort(3,(1:round(L/10)));% locus numbers used in AveNestLoCV
270 AveNestLoCV=mean(NestLoCV);
271 SDNestLoCV=std(NestLoCV);
272
273 %% End PASS2 with allele 2
274
275 %% change names to avoid overwrite of PASS1
276 p2=p;L2=L;Nit2=Nit;Ntry2=Ntry;maxNtry2=maxNtry;Ptry2=Ptry;
277 samp2=samp;jj2=jj;n2=n;j2=j;indit2=indit;
278 propdetNit2=propdetNit;HypExp2=HypExp;BIAS1o2=BIAS1;
279 Nest2=Nest;Pest2=Pest;AveNest2=AveNest;SeNest2=SeNest;CVNest2=CVNest;
280 CVLoLoci2=CVLoLoci;AveNestLoCV2=AveNestLoCV;SDNestLoCV2=SDNestLoCV;
281 NestLoCV2=NestLoCV;
282
283 %% Now make NestBoth by combined av Nest 1 +2
284 NestLoCV2=NestLoCV;
285 NestLoCVTEMP=load(filename,'NestLoCV');%gets back Nest for other allele✓
(overwritten)
286 workaroud=struct2cell(NestLoCVTEMP);NestLoCV1=[workaroud{:}];%workaroud
287 NestBothLoCV=[NestLoCV1,NestLoCV2];% concatenates NestLoCV a1 & a2
288 AveNestBothLoCV=mean(NestBothLoCV);
289 SDNestBothLoCV=std(NestBothLoCV);
290
291 %% Binomial better fit than Hyp? If so, need bigger sample genotyped
292 % for most likely values AveNestLoCV and AvePest(1:L)=(Pest+Pest2)/2
293
294 load(filename,'Pest');% get back Pest from PASS1
295 Pest2reflect=1-Pest2; % Put Pest2 in same direction as Pest2
296 AvePest(:)=(Pest(:)+Pest2reflect(:))./2;% Av p estimate for each locus
297 nNest=round(AveNestBothLoCV*samp);jNest=round(nNest*jj);%appropriate n,j Value
298 BinomExp=zeros(L,jNest+1);%fill to jNest+1 with zeros
299 HypExpFinal=zeros(L,jNest+1);%fill to jNest+1 with zeros
300 HypExpFinalTEMP=zeros(L,jNest+1);%fill to jNest+1 with zeros
301
302 for l=1:L
303     % Binomial Exps for subsample of 'jNest',AvePest(l)
304     bins=[0:1:jNest];% bins for BinomExps
305     parg=AvePest(l);narg=jNest;% Arguments for next line
306     [BinomExpTEMP1(1:jNest+1)]=binopdf(bins,narg,parg);
307     BinomExp(l,(1:jNest+1))=BinomExpTEMP1;%1*jmaxALL, trailing zeros
308     % Hyp Exps for AveNestLoCV,jNest,AvePest(l)
309     parg=AvePest(l);narg=jNest;Narg=round(AveNestLoCV);%Arguments for next
310     [HypExpFinalTEMP(l,(1:(jNest+1)))]=HypExactExp230607(parg,Narg,narg);%1*jNest+1
311 end;% end locus loop
312 % Fix numbers so low they are NaN ("not a number")
313 TF = isnan(HypExpFinalTEMP);% "1" wherever value so low it is NaN, others"0"
314 HypExpFinal=HypExpFinalTEMP;
315 HypExpFinal(TF)=0;
316

```

```

317 % Get final BIAS
318 load(filename,'propdetNit');%propdetNit is L*jmaxALL
319 propdetTRIM=propdetNit(:,1:jNest+1); % trim trailing zeros
320 BIASBinom=abs(propdetTRIM-BinomExp); %DIM row 1:L, Col 1:jNest+1
321 BIASbinVec=reshape(BIASBinom,[1,numel(BIASBinom)]);%for t-test
322 BIASbin=sum(BIASBinom,"all");% NB lower case 'b' sum all bins & all loci
323 BIASHyp=abs(propdetTRIM-HypExpFinal);
324 BIAShypVec=reshape(BIASHyp,[1,numel(BIASHyp)]);%linear for t-test
325 BIAShyp=sum(BIASHyp,"all");% NB lower case 'h' sum all bins & all loci
326
327 % Useless test whether Hypergeom fits jackknife data better than binomial
328 [h,psig] = ttest(BIAShypVec,BIASbinVec);
329 SamSize=BIASHyp-BIASbin; % positive if hypergeom fits better than binomial
330 %if psig<=0.05
331 %     if SamSize>0
332 %         disp('Need bigger sample, SamSize= '); SamSize
333 %     elseif SamSize<0 disp('Hypergeom fits better than Binomial, sample OK');
334 % end; % end if SamSize
335 % end; % end if psig
336
337 save(
(filename,'p2','L2','Nit2','Ntry2','maxNtry2','Ptry2','samp2','jj2','n2','j2','indit
2','propdetNit2','HypExp2','BIASlo2','Nest2','Pest2','AveNest2','SeNest2','CVNest2',
'CVLoLoci2','AveNestLoCV2','SDNestLoCV2','NestLoCV2','CVLoLoci2','AveNestLoCV2','SDN
estLoCV2','NestBothLoCV','AveNestBothLoCV','SDNestBothLoCV','BIASBinom','BIASHyp','p
sig','SamSize','NestBothLoCV','AveNestBothLoCV','SDNestBothLoCV',"--append")
338
339 %% Full Binomial if needed
340 % Binomial Exps for subsample of 'j(Nindex)'
341 %BinomExp=zeros(length(Ntry),length(Ptry),jmaxALL);
342 %for Nindex=1:(length(Ntry)); %loop trial N's
343 %for pindex=1:(length(Ptry)); %loop trial P's
344 %     bins=[0:1:j(Nindex)];% bins for BinomExps
345 %     parg=Ptry(pindex);narg=j(Nindex);% Arguments for next line
346 %     [BinomExpTEMP1(1:(j(Nindex)+1))]=binopdf(bins,narg,parg);
347 %     BinomExp(Nindex,pindex,(1:j(Nindex)+1))=BinomExpTEMP1;%filled to jmax with
zeros
348 %end; % end loop for BinomExp Ptry values
349 %end; % end loop for BinomExp Ntry values

```