Obligate cross-feeding expands the metabolic niche of bacteria

Data & functions

```
niche = Import["/Users/xxx.xlsx"];
```

```
In[•]:= convLis2[x_] :=
         {Flatten[niche[[2]][[x[[1]] ;; x[[1]] + 7, #]] & /@ Range[x[[2]], x[[2]] + 3]],
          Flatten[niche[[2]][[x[[1]] ;; x[[1]] + 7, #]] & /@
            Range[x[[2]] + 14, x[[2]] + 3 + 14]],
          Flatten[niche[[2]][[x[[1]] ;; x[[1]] + 7, #]] & /@
            Range[x[[2]] + 14 + 14, x[[2]] + 3 + 14 + 14]]
         }
```

```
In[•]:= pop5Pos = Partition[Flatten[{{#, 90}, {#, 94}, {#, 98}} & /@
             Accumulate[Prepend[ConstantArray[11, 48], 4]]], {2}];
```

```
In[•]:= pop5 = convLis2 /@ pop5Pos;
```

```
(*Monocultures with AA 5*)
In[•]:= ABR5 = pop5[[18]];
In[•]:= ABH5 = pop5[[19]];
In[•]:= ABW5 = pop5[[20]];
In[•]:= ABL5 = pop5[[21]];



In[•]:= BSR5 = pop5[[22]];
In[•]:= BSH5 = pop5[[23]];
In[•]:= BSL5 = pop5[[24]];
```

```
In[•]:= ECR5 = pop5[[25]];

In[•]:= ECH5 = pop5[[26]];

In[•]:= ECW5 = pop5[[27]];

In[•]:= ECL5 = pop5[[28]];
```

```
In[•]:= SOR5 = pop5[[29]];

In[•]:= SOH5 = pop5[[30]];

In[•]:= SOW5 = pop5[[31]];

In[•]:= SOL5 = pop5[[32]];
```

```
In[•]:= PFW5 = pop5[[33]];

In[•]:= PFL5 = pop5[[34]];
```

```
In[]:= comType = niche[[2]][[#[[1]]]][[#[[2]]]] & /@ Partition[Flatten[
          {{#, 2}, {#, 6}, {#, 10}} & /@ Accumulate[Prepend[ConstantArray[11, 48], 2]]], {2}]

Out[]= {AB R, AB H, AB W , AB L, BS R, BS H, BS L, EC R, EC H, EC W, EC L, SO R, SO H, SO W,
        SO L, PF W, PF L, AB R, AB H, AB W , AB L, BS R, BS H, BS L, EC R, EC H, EC W,
        EC L, SO R, SO H, SO W, SO L, PF W, PF L, AB, BS, EC, SO , PF, AB R - AB H ,
        AB R - BS H , AB R - EC H, AB R - SO H, BS R - AB H, BS R - BS H, BS R - EC H,
        BS R - SO H, EC R - AB H, EC R - BS H, EC R - EC H , EC R - SO H , SO R - AB H,
        SO R - BS H, SO R - EC H, SO R - SO H, AB H - AB W , AB H - EC W, AB H - SO W,
        AB H - PF W, BS H - AB W, BS H - EC W , BS H - SO W, BS H - PF W, EC H - AB W,
        EC H - EC W , EC H - SO W , EC H - PF W, SO H - AB W, SO H - EC W, SO H - SO W ,
        SO H - PF W, AB W - AB R, AB W - BS R, AB W - EC R, AB W - SO R, EC W - AB R,
        EC W - BS R, EC W - EC R, EC W - SO R, SO W - AB R, SO W - BS R, SO W - EC R,
        SO W - SO R, PF W - AB R, PF W - BS R, PF W - EC R, PF W - SO R, AB W - AB L,
        AB W - BS L, AB W - EC L, AB W - SO L, AB W - PF L , EC W - AB L, EC W - BS L,
        EC W - EC L, EC W - SO L, EC W - PF L, SO W - AB L, SO W - BS L, SO W - EC L,
        SO W - SO L, SO W - PF L , PF W - AB L, PF W - BS L, PF W - EC L, PF W - SO L,
        PF W - PF L, AB L - AB R, AB L - BS R, AB L - EC R, AB L - SO R, BS L - AB R,
        BS L - BS R, BS L - EC R, BS L - SO R, EC L - AB R, EC L - BS R, EC L - EC R,
        EC L - SO R, SO L - AB R, SO L - BS R, SO L - EC R, SO L - SO R, PF L - AB R,
        PF L - BS R, PF L - EC R, PF L - SO R, AB H - AB L , AB H - BS L , AB H - EC L,
        AB H - SO L, AB H - PF L, BS H - AB L, BS H - BS - L, BS H - EC L ,
        BS H - SO L, BS H - PF L, EC H - AB L, EC H - BS L, EC H - EC L , EC H - SO L ,
        EC H - PF L, SO H - AB L, SO H - BS L, SO H - EC L, SO H - SO L , SO H - PF L}

In[]:= th[x_] := If[x ≥ 0.08, 1, 0]

In[]:= bit[x_] := If[x ≥ 1, 1, 0]

In[]:= poSting5 = StringReplace[#, {"-" -> "", " " → ""}] & /@ comType[[40 ;; 147]]

Out[]= {ABRABH, ABRBSH, ABRECH, ABRSOH, BSRABH, BSRBSH, BSRECH, BSRSOH, ECRABH, ECRBSH,
        ECRECH, ECRSOH, SORABH, SORBSH, SORECH, SORSOH, ABHABW, ABHECW, ABHSOW, ABHPFW,
        BSHABW, BSHECW, BSHSOW, BSHPFW, ECHABW, ECHECW, ECHSOW, ECHPFW, SOHABW, SOHECW,
        SOHSOW, SOHPFW, ABWABR, ABWBSR, ABWECR, ABWSOR, ECWABR, ECWBSR, ECWECR, ECWSOR,
        SOWABR, SOWBSR, SOWECR, SOWSOR, PFWABR, PFWBSR, PFWECR, PFWSOR, ABWABL, ABWBSL,
        ABWECL, ABWSOL, ABWPFL, ECWABL, ECWBSL, ECWECL, ECWSOL, ECWPFL, SOWABL, SOWBSL,
        SOWECL, SOWSOL, SOWPFL, PFWABL, PFWBSL, PFWECL, PFWSOL, PFWPFL, ABLABR, ABLBSR,
        ABLECR, ABLSOR, BSLABR, BSLBSR, BSLECR, BSLSOR, ECLABR, ECLBSR, ECLECR, ECLSOR,
        SOLABR, SOLBSR, SOLECR, SOLSOR, PFLABR, PFLBSR, PFLECR, PFLSOR, ABHABL, ABHBSL,
        ABHECL, ABHSOL, ABHPFL, BSHABL, BSHBSL, BSHECL, BSHSOL, BSHPFL, ECHABL,
        ECHBSL, ECHECL, ECHSOL, ECHPFL, SOHABL, SOHBSL, SOHECL, SOHSOL, SOHPFL}
```

```
In[◦]:= poStingHour5 = # <> "5" & /@ poSting5

Out[◦]= {ABRABH5, ABRBSH5, ABRECH5, ABRSOH5, BSRABH5, BSRBSH5, BSRECH5, BSRSOH5, ECRABH5,
         ECRBSH5, ECRECH5, ECRSOH5, SORABH5, SORBSH5, SORECH5, SORSOH5, ABHABW5, ABHECW5,
         ABHSOW5, ABHPFW5, BSHABW5, BSHECW5, BSHSOW5, BSHPFW5, ECHABW5, ECHECW5, ECHSOW5,
         ECHPFW5, SOHABW5, SOHECW5, SOHSOW5, SOHPFW5, ABWABR5, ABWBSR5, ABWECR5, ABWSOR5,
         ECWABR5, ECWBSR5, ECWECR5, ECWSOR5, SOWABR5, SOWBSR5, SOWECR5, SOWSOR5,
         PFWABR5, PFWBSR5, PFWECR5, PFWSOR5, ABWABL5, ABWBSL5, ABWECL5, ABWSOL5,
         ABWPFL5, ECWABL5, ECWBSL5, ECWECL5, ECWSOL5, ECWPFL5, SOWABL5, SOWBSL5,
         SOWECL5, SOWSOL5, SOWPFL5, PFWABL5, PFWBSL5, PFWECL5, PFWSOL5, PFWPFL5,
         ABLABR5, ABLBSR5, ABLECR5, ABLSOR5, BSLABR5, BSLBSR5, BSLECR5, BSLSOR5,
         ECLABR5, ECLBSR5, ECLECR5, ECLSOR5, SOLABR5, SOLBSR5, SOLECR5, SOLSOR5,
         PFLABR5, PFLBSR5, PFLECR5, PFLSOR5, ABHABL5, ABHBSL5, ABHECL5, ABHSOL5,
         ABHPFL5, BSHABL5, BSHBSL5, BSHECL5, BSHSOL5, BSHPFL5, ECHABL5, ECHBSL5,
         ECHECL5, ECHSOL5, ECHPFL5, SOHABL5, SOHBSL5, SOHECL5, SOHSOL5, SOHPFL5}

In[◦]:= co5 = ToExpression /@ poStingHour5

In[◦]:= co5

In[◦]:= {ABRABH5, ABRBSH5, ABRECH5, ABRSOH5, BSRABH5, BSRBSH5, BSRECH5, BSRSOH5, ECRABH5,
           ECRBSH5, ECRECH5, ECRSOH5, SORABH5, SORBSH5, SORECH5, SORSOH5, ABHABW5, ABHECW5,
           ABHSOW5, ABHPFW5, BSHABW5, BSHECW5, BSHSOW5, BSHPFW5, ECHABW5, ECHECW5, ECHSOW5,
           ECHPFW5, SOHABW5, SOHECW5, SOHSOW5, SOHPFW5, ABWABR5, ABWBSR5, ABWECR5, ABWSOR5,
           ECWABR5, ECWBSR5, ECWECR5, ECWSOR5, SOWABR5, SOWBSR5, SOWECR5, SOWSOR5, PFWABR5,
           PFWBSR5, PFWECR5, PFWSOR5, ABWABL5, ABWBSL5, ABWECL5, ABWSOL5, ABWPFL5, ECWABL5,
           ECWBSL5, ECWECL5, ECWSOL5, ECWPFL5, SOWABL5, SOWBSL5, SOWECL5, SOWSOL5,
           SOWPFL5, PFWABL5, PFWBSL5, PFWECL5, PFWSOL5, PFWPFL5, ABLABR5, ABLBSR5,
           ABLECR5, ABLSOR5, BSLABR5, BSLBSR5, BSLECR5, BSLSOR5, ECLABR5, ECLBSR5,
           ECLECR5, ECLSOR5, SOLABR5, SOLBSR5, SOLECR5, SOLSOR5, PFLABR5, PFLBSR5,
           PFLECR5, PFLSOR5, ABHABL5, ABHBSL5, ABHECL5, ABHSOL5, ABHPFL5, BSHABL5,
           BSHBSL5, BSHECL5, BSHSOL5, BSHPFL5, ECHABL5, ECHBSL5, ECHECL5, ECHSOL5,
           ECHPFL5, SOHABL5, SOHBSL5, SOHECL5, SOHSOL5, SOHPFL5} = pop5[[40 ;; 147]];
```

```
In[◦]:= funNe[x_] := If[x > 0, x, 0]

        (*Total P1, Total P2, MonoIntersection,
        TotalCo, MoICoIntersection, contract, expans*)

In[◦]:= Clear[th1, bit1, qf1, v1, qf2, v2, overlapMono, qfcocul,
         vcocul, MoICoIntersection, contractxk, expandxk, contract, expans]
```

In[•]:=

```
nicheOverlap[par1_, par2_, coul_, threashold_] := (
  th1 = If[# ≥ threashold, 1, 0] &;
  bit1 = If[# ≥ 1, 1, 0] &;

  qf1 = Map[th1, par1, {2}];
  v1 = bit1 /@ Total[qf1];

  qf2 = Map[th1, par2, {2}];
  v2 = bit1 /@ Total[qf2];

  overlapMono = v1 v2;

  qfcocul = Map[th1, coul, {2}];
  vcocul = bit1 /@ Total[qfcocul];

  MoICoIntersection = overlapMono vcocul;

  contractxk = Total[overlapMono] - Total[MoICoIntersection];
  expandxk = Total[vcocul] - Total[MoICoIntersection];

  contract = funNe[contractxk];
  expans = funNe[expandxk];

  {Total[v1], Total[v2], Total[overlapMono],
   Total[vcocul], Total[MoICoIntersection], contract, expans}


 )
```

```
In[•]:= ki5 = StringReplace[#, {"-" -> " ", " " → ""}] & /@ comType[[40 ;; 147]]

Out[•]= {ABR ABH, ABR BSH, ABR ECH, ABR SOH, BSR ABH, BSR BSH, BSR ECH, BSR SOH, ECR ABH,
         ECR BSH, ECR ECH, ECR SOH, SOR ABH, SOR BSH, SOR ECH, SOR SOH, ABH ABW, ABH ECW,
         ABH SOW, ABH PFW, BSH ABW, BSH ECW, BSH SOW, BSH PFW, ECH ABW, ECH ECW, ECH SOW,
         ECH PFW, SOH ABW, SOH ECW, SOH SOW, SOH PFW, ABW ABR, ABW BSR, ABW ECR, ABW SOR,
         ECW ABR, ECW BSR, ECW ECR, ECW SOR, SOW ABR, SOW BSR, SOW ECR, SOW SOR,
         PFW ABR, PFW BSR, PFW ECR, PFW SOR, ABW ABL, ABW BSL, ABW ECL, ABW SOL,
         ABW PFL, ECW ABL, ECW BSL, ECW ECL, ECW SOL, ECW PFL, SOW ABL, SOW BSL,
         SOW ECL, SOW SOL, SOW PFL, PFW ABL, PFW BSL, PFW ECL, PFW SOL, PFW PFL,
         ABL ABR, ABL BSR, ABL ECR, ABL SOR, BSL ABR, BSL BSR, BSL ECR, BSL SOR,
         ECL ABR, ECL BSR, ECL ECR, ECL SOR, SOL ABR, SOL BSR, SOL ECR, SOL SOR,
         PFL ABR, PFL BSR, PFL ECR, PFL SOR, ABH ABL, ABH BSL, ABH ECL, ABH SOL,
         ABH PFL, BSH ABL, BSH BS L, BSH ECL, BSH SOL, BSH PFL, ECH ABL, ECH BSL,
         ECH ECL, ECH SOL, ECH PFL, SOH ABL, SOH BSL, SOH ECL, SOH SOL, SOH PFL}

In[•]:= pai5 = StringSplit /@ ki5

Out[•]= {{ABR, ABH}, {ABR, BSH}, {ABR, ECH}, {ABR, SOH}, {BSR, ABH}, {BSR, BSH}, {BSR, ECH},
         {BSR, SOH}, {ECR, ABH}, {ECR, BSH}, {ECR, ECH}, {ECR, SOH}, {SOR, ABH}, {SOR, BSH},
         {SOR, ECH}, {SOR, SOH}, {ABH, ABW}, {ABH, ECW}, {ABH, SOW}, {ABH, PFW}, {BSH, ABW},
         {BSH, ECW}, {BSH, SOW}, {BSH, PFW}, {ECH, ABW}, {ECH, ECW}, {ECH, SOW}, {ECH, PFW},
         {SOH, ABW}, {SOH, ECW}, {SOH, SOW}, {SOH, PFW}, {ABW, ABR}, {ABW, BSR}, {ABW, ECR},
         {ABW, SOR}, {ECW, ABR}, {ECW, BSR}, {ECW, ECR}, {ECW, SOR}, {SOW, ABR}, {SOW, BSR},
         {SOW, ECR}, {SOW, SOR}, {PFW, ABR}, {PFW, BSR}, {PFW, ECR}, {PFW, SOR}, {ABW, ABL},
         {ABW, BSL}, {ABW, ECL}, {ABW, SOL}, {ABW, PFL}, {ECW, ABL}, {ECW, BSL}, {ECW, ECL},
         {ECW, SOL}, {ECW, PFL}, {SOW, ABL}, {SOW, BSL}, {SOW, ECL}, {SOW, SOL}, {SOW, PFL},
         {PFW, ABL}, {PFW, BSL}, {PFW, ECL}, {PFW, SOL}, {PFW, PFL}, {ABL, ABR}, {ABL, BSR},
         {ABL, ECR}, {ABL, SOR}, {BSL, ABR}, {BSL, BSR}, {BSL, ECR}, {BSL, SOR}, {ECL, ABR},
         {ECL, BSR}, {ECL, ECR}, {ECL, SOR}, {SOL, ABR}, {SOL, BSR}, {SOL, ECR}, {SOL, SOR},
         {PFL, ABR}, {PFL, BSR}, {PFL, ECR}, {PFL, SOR}, {ABH, ABL}, {ABH, BSL},
         {ABH, ECL}, {ABH, SOL}, {ABH, PFL}, {BSH, ABL}, {BSH, BS, L}, {BSH, ECL},
         {BSH, SOL}, {BSH, PFL}, {ECH, ABL}, {ECH, BSL}, {ECH, ECL}, {ECH, SOL},
         {ECH, PFL}, {SOH, ABL}, {SOH, BSL}, {SOH, ECL}, {SOH, SOL}, {SOH, PFL}}
```

```
In[•]:= qz5 = MapAt[(# <> "5" &), pai5, {All, All}]

Out[•]= {{ABR5, ABH5}, {ABR5, BSH5}, {ABR5, ECH5}, {ABR5, SOH5}, {BSR5, ABH5}, {BSR5, BSH5},
         {BSR5, ECH5}, {BSR5, SOH5}, {ECR5, ABH5}, {ECR5, BSH5}, {ECR5, ECH5}, {ECR5, SOH5},
         {SOR5, ABH5}, {SOR5, BSH5}, {SOR5, ECH5}, {SOR5, SOH5}, {ABH5, ABW5}, {ABH5, ECW5},
         {ABH5, SOW5}, {ABH5, PFW5}, {BSH5, ABW5}, {BSH5, ECW5}, {BSH5, SOW5}, {BSH5, PFW5},
         {ECH5, ABW5}, {ECH5, ECW5}, {ECH5, SOW5}, {ECH5, PFW5}, {SOH5, ABW5}, {SOH5, ECW5},
         {SOH5, SOW5}, {SOH5, PFW5}, {ABW5, ABR5}, {ABW5, BSR5}, {ABW5, ECR5}, {ABW5, SOR5},
         {ECW5, ABR5}, {ECW5, BSR5}, {ECW5, ECR5}, {ECW5, SOR5}, {SOW5, ABR5}, {SOW5, BSR5},
         {SOW5, ECR5}, {SOW5, SOR5}, {PFW5, ABR5}, {PFW5, BSR5}, {PFW5, ECR5}, {PFW5, SOR5},
         {ABW5, ABL5}, {ABW5, BSL5}, {ABW5, ECL5}, {ABW5, SOL5}, {ABW5, PFL5}, {ECW5, ABL5},
         {ECW5, BSL5}, {ECW5, ECL5}, {ECW5, SOL5}, {ECW5, PFL5}, {SOW5, ABL5}, {SOW5, BSL5},
         {SOW5, ECL5}, {SOW5, SOL5}, {SOW5, PFL5}, {PFW5, ABL5}, {PFW5, BSL5}, {PFW5, ECL5},
         {PFW5, SOL5}, {PFW5, PFL5}, {ABL5, ABR5}, {ABL5, BSR5}, {ABL5, ECR5}, {ABL5, SOR5},
         {BSL5, ABR5}, {BSL5, BSR5}, {BSL5, ECR5}, {BSL5, SOR5}, {ECL5, ABR5}, {ECL5, BSR5},
         {ECL5, ECR5}, {ECL5, SOR5}, {SOL5, ABR5}, {SOL5, BSR5}, {SOL5, ECR5}, {SOL5, SOR5},
         {PFL5, ABR5}, {PFL5, BSR5}, {PFL5, ECR5}, {PFL5, SOR5}, {ABH5, ABL5}, {ABH5, BSL5},
         {ABH5, ECL5}, {ABH5, SOL5}, {ABH5, PFL5}, {BSH5, ABL5}, {BSH5, BS5, L5}, {BSH5, ECL5},
         {BSH5, SOL5}, {BSH5, PFL5}, {ECH5, ABL5}, {ECH5, BSL5}, {ECH5, ECL5}, {ECH5, SOL5},
         {ECH5, PFL5}, {SOH5, ABL5}, {SOH5, BSL5}, {SOH5, ECL5}, {SOH5, SOL5}, {SOH5, PFL5}}
```

```
In[•]:= qzK5 = {{"ABR5", "ABH5"}, {"ABR5", "BSH5"}, {"ABR5", "ECH5"}, {"ABR5", "SOH5"},
           {"BSR5", "ABH5"}, {"BSR5", "BSH5"}, {"BSR5", "ECH5"}, {"BSR5", "SOH5"},
           {"ECR5", "ABH5"}, {"ECR5", "BSH5"}, {"ECR5", "ECH5"}, {"ECR5", "SOH5"},
           {"SOR5", "ABH5"}, {"SOR5", "BSH5"}, {"SOR5", "ECH5"}, {"SOR5", "SOH5"},
           {"ABH5", "ABW5"}, {"ABH5", "ECW5"}, {"ABH5", "SOW5"}, {"ABH5", "PFW5"},
           {"BSH5", "ABW5"}, {"BSH5", "ECW5"}, {"BSH5", "SOW5"}, {"BSH5", "PFW5"},
           {"ECH5", "ABW5"}, {"ECH5", "ECW5"}, {"ECH5", "SOW5"}, {"ECH5", "PFW5"},
           {"SOH5", "ABW5"}, {"SOH5", "ECW5"}, {"SOH5", "SOW5"}, {"SOH5", "PFW5"},
           {"ABW5", "ABR5"}, {"ABW5", "BSR5"}, {"ABW5", "ECR5"}, {"ABW5", "SOR5"},
           {"ECW5", "ABR5"}, {"ECW5", "BSR5"}, {"ECW5", "ECR5"}, {"ECW5", "SOR5"},
           {"SOW5", "ABR5"}, {"SOW5", "BSR5"}, {"SOW5", "ECR5"}, {"SOW5", "SOR5"},
           {"PFW5", "ABR5"}, {"PFW5", "BSR5"}, {"PFW5", "ECR5"}, {"PFW5", "SOR5"},
           {"ABW5", "ABL5"}, {"ABW5", "BSL5"}, {"ABW5", "ECL5"}, {"ABW5", "SOL5"},
           {"ABW5", "PFL5"}, {"ECW5", "ABL5"}, {"ECW5", "BSL5"}, {"ECW5", "ECL5"},
           {"ECW5", "SOL5"}, {"ECW5", "PFL5"}, {"SOW5", "ABL5"}, {"SOW5", "BSL5"},
           {"SOW5", "ECL5"}, {"SOW5", "SOL5"}, {"SOW5", "PFL5"}, {"PFW5", "ABL5"},
           {"PFW5", "BSL5"}, {"PFW5", "ECL5"}, {"PFW5", "SOL5"}, {"PFW5", "PFL5"},
           {"ABL5", "ABR5"}, {"ABL5", "BSR5"}, {"ABL5", "ECR5"}, {"ABL5", "SOR5"},
           {"BSL5", "ABR5"}, {"BSL5", "BSR5"}, {"BSL5", "ECR5"}, {"BSL5", "SOR5"},
           {"ECL5", "ABR5"}, {"ECL5", "BSR5"}, {"ECL5", "ECR5"}, {"ECL5", "SOR5"},
           {"SOL5", "ABR5"}, {"SOL5", "BSR5"}, {"SOL5", "ECR5"}, {"SOL5", "SOR5"},
           {"PFL5", "ABR5"}, {"PFL5", "BSR5"}, {"PFL5", "ECR5"}, {"PFL5", "SOR5"},
           {"ABH5", "ABL5"}, {"ABH5", "BSL5"}, {"ABH5", "ECL5"}, {"ABH5", "SOL5"},
           {"ABH5", "PFL5"}, {"BSH5", "ABL5"}, {"BSH5", "BSL5"}, {"BSH5", "ECL5"},
           {"BSH5", "SOL5"}, {"BSH5", "PFL5"}, {"ECH5", "ABL5"}, {"ECH5", "BSL5"},
           {"ECH5", "ECL5"}, {"ECH5", "SOL5"}, {"ECH5", "PFL5"}, {"SOH5", "ABL5"},
           {"SOH5", "BSL5"}, {"SOH5", "ECL5"}, {"SOH5", "SOL5"}, {"SOH5", "PFL5"}};
```

```
In[•]:= Length /@ qzK5
```

```
Out[•]= {2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
         2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
         2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
         2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2}
```

```
In[•]:= mondef5 = MapAt[ToExpression, qzK5, {All, All}];
```

```
In[•]:= jop5 = Table[Join[pai5[[i]], {poSting5[[i]]}], {i, 1, Length[poSting5]}]

Out[•]= {{ABR, ABH, ABRABH}, {ABR, BSH, ABRBSH}, {ABR, ECH, ABRECH}, {ABR, SOH, ABRSOH},
  {BSR, ABH, BSRABH}, {BSR, BSH, BSRBSH}, {BSR, ECH, BSRECH}, {BSR, SOH, BSRSOH},
  {ECR, ABH, ECRABH}, {ECR, BSH, ECRBSH}, {ECR, ECH, ECRECH}, {ECR, SOH, ECRSOH},
  {SOR, ABH, SORABH}, {SOR, BSH, SORBSH}, {SOR, ECH, SORECH}, {SOR, SOH, SORSOH},
  {ABH, ABW, ABHABW}, {ABH, ECW, ABHECW}, {ABH, SOW, ABHSOW}, {ABH, PFW, ABHPFW},
  {BSH, ABW, BSHABW}, {BSH, ECW, BSHECW}, {BSH, SOW, BSHSOW}, {BSH, PFW, BSHPFW},
  {ECH, ABW, ECHABW}, {ECH, ECW, ECHECW}, {ECH, SOW, ECHSOW}, {ECH, PFW, ECHPFW},
  {SOH, ABW, SOHABW}, {SOH, ECW, SOHECW}, {SOH, SOW, SOHSOW}, {SOH, PFW, SOHPFW},
  {ABW, ABR, ABWABR}, {ABW, BSR, ABWBSR}, {ABW, ECR, ABWECR}, {ABW, SOR, ABWSOR},
  {ECW, ABR, ECWABR}, {ECW, BSR, ECWBSR}, {ECW, ECR, ECWECR}, {ECW, SOR, ECWSOR},
  {SOW, ABR, SOWABR}, {SOW, BSR, SOWBSR}, {SOW, ECR, SOWECR}, {SOW, SOR, SOWSOR},
  {PFW, ABR, PFWABR}, {PFW, BSR, PFWBSR}, {PFW, ECR, PFWECR}, {PFW, SOR, PFWSOR},
  {ABW, ABL, ABWABL}, {ABW, BSL, ABWBSL}, {ABW, ECL, ABWECL}, {ABW, SOL, ABWSOL},
  {ABW, PFL, ABWPFL}, {ECW, ABL, ECWABL}, {ECW, BSL, ECWBSL}, {ECW, ECL, ECWECL},
  {ECW, SOL, ECWSOL}, {ECW, PFL, ECWPFL}, {SOW, ABL, SOWABL}, {SOW, BSL, SOWBSL},
  {SOW, ECL, SOWECL}, {SOW, SOL, SOWSOL}, {SOW, PFL, SOWPFL}, {PFW, ABL, PFWABL},
  {PFW, BSL, PFWBSL}, {PFW, ECL, PFWECL}, {PFW, SOL, PFWSOL}, {PFW, PFL, PFWPFL},
  {ABL, ABR, ABLABR}, {ABL, BSR, ABLBSR}, {ABL, ECR, ABLECR}, {ABL, SOR, ABLSOR},
  {BSL, ABR, BSLABR}, {BSL, BSR, BSLBSR}, {BSL, ECR, BSLECR}, {BSL, SOR, BSLSOR},
  {ECL, ABR, ECLABR}, {ECL, BSR, ECLBSR}, {ECL, ECR, ECLECR}, {ECL, SOR, ECLSOR},
  {SOL, ABR, SOLABR}, {SOL, BSR, SOLBSR}, {SOL, ECR, SOLECR}, {SOL, SOR, SOLSOR},
  {PFL, ABR, PFLABR}, {PFL, BSR, PFLBSR}, {PFL, ECR, PFLECR}, {PFL, SOR, PFLSOR},
  {ABH, ABL, ABHABL}, {ABH, BSL, ABHBSL}, {ABH, ECL, ABHECL}, {ABH, SOL, ABHSOL},
  {ABH, PFL, ABHPFL}, {BSH, ABL, BSHABL}, {BSH, BS, L, BSHBSL}, {BSH, ECL, BSHECL},
  {BSH, SOL, BSHSOL}, {BSH, PFL, BSHPFL}, {ECH, ABL, ECHABL}, {ECH, BSL, ECHBSL},
  {ECH, ECL, ECHECL}, {ECH, SOL, ECHSOL}, {ECH, PFL, ECHPFL}, {SOH, ABL, SOHABL},
  {SOH, BSL, SOHBSL}, {SOH, ECL, SOHECL}, {SOH, SOL, SOHSOL}, {SOH, PFL, SOHPFL}}

In[•]:= Length /@ jop5

Out[•]= {3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
  3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
  3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
  3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3}
```

```
In[•]:= jop5 = {{"ABR", "ABH", "ABRABH"}, {"ABR", "BSH", "ABRBSH"}, {"ABR", "ECH", "ABRECH"},
           {"ABR", "SOH", "ABRSOH"}, {"BSR", "ABH", "BSRABH"}, {"BSR", "BSH", "BSRBSH"},
           {"BSR", "ECH", "BSRECH"}, {"BSR", "SOH", "BSRSOH"}, {"ECR", "ABH", "ECRABH"},
           {"ECR", "BSH", "ECRBSH"}, {"ECR", "ECH", "ECRECH"}, {"ECR", "SOH", "ECRSOH"},
           {"SOR", "ABH", "SORABH"}, {"SOR", "BSH", "SORBSH"}, {"SOR", "ECH", "SORECH"},
           {"SOR", "SOH", "SORSOH"}, {"ABH", "ABW", "ABHABW"}, {"ABH", "ECW", "ABHECW"},
           {"ABH", "SOW", "ABHSOW"}, {"ABH", "PFW", "ABHPFW"}, {"BSH", "ABW", "BSHABW"},
           {"BSH", "ECW", "BSHECW"}, {"BSH", "SOW", "BSHSOW"}, {"BSH", "PFW", "BSHPFW"},
           {"ECH", "ABW", "ECHABW"}, {"ECH", "ECW", "ECHECW"}, {"ECH", "SOW", "ECHSOW"},
           {"ECH", "PFW", "ECHPFW"}, {"SOH", "ABW", "SOHABW"}, {"SOH", "ECW", "SOHECW"},
           {"SOH", "SOW", "SOHSOW"}, {"SOH", "PFW", "SOHPFW"}, {"ABW", "ABR", "ABWABR"},
           {"ABW", "BSR", "ABWBSR"}, {"ABW", "ECR", "ABWECR"}, {"ABW", "SOR", "ABWSOR"},
           {"ECW", "ABR", "ECWABR"}, {"ECW", "BSR", "ECWBSR"}, {"ECW", "ECR", "ECWECR"},
           {"ECW", "SOR", "ECWSOR"}, {"SOW", "ABR", "SOWABR"}, {"SOW", "BSR", "SOWBSR"},
           {"SOW", "ECR", "SOWECR"}, {"SOW", "SOR", "SOWSOR"}, {"PFW", "ABR", "PFWABR"},
           {"PFW", "BSR", "PFWBSR"}, {"PFW", "ECR", "PFWECR"}, {"PFW", "SOR", "PFWSOR"},
           {"ABW", "ABL", "ABWABL"}, {"ABW", "BSL", "ABWBSL"}, {"ABW", "ECL", "ABWECL"},
           {"ABW", "SOL", "ABWSOL"}, {"ABW", "PFL", "ABWPFL"}, {"ECW", "ABL", "ECWABL"},
           {"ECW", "BSL", "ECWBSL"}, {"ECW", "ECL", "ECWECL"}, {"ECW", "SOL", "ECWSOL"},
           {"ECW", "PFL", "ECWPFL"}, {"SOW", "ABL", "SOWABL"}, {"SOW", "BSL", "SOWBSL"},
           {"SOW", "ECL", "SOWECL"}, {"SOW", "SOL", "SOWSOL"}, {"SOW", "PFL", "SOWPFL"},
           {"PFW", "ABL", "PFWABL"}, {"PFW", "BSL", "PFWBSL"}, {"PFW", "ECL", "PFWECL"},
           {"PFW", "SOL", "PFWSOL"}, {"PFW", "PFL", "PFWPFL"}, {"ABL", "ABR", "ABLABR"},
           {"ABL", "BSR", "ABLBSR"}, {"ABL", "ECR", "ABLECR"}, {"ABL", "SOR", "ABLSOR"},
           {"BSL", "ABR", "BSLABR"}, {"BSL", "BSR", "BSLBSR"}, {"BSL", "ECR", "BSLECR"},
           {"BSL", "SOR", "BSLSOR"}, {"ECL", "ABR", "ECLABR"}, {"ECL", "BSR", "ECLBSR"},
           {"ECL", "ECR", "ECLECR"}, {"ECL", "SOR", "ECLSOR"}, {"SOL", "ABR", "SOLABR"},
           {"SOL", "BSR", "SOLBSR"}, {"SOL", "ECR", "SOLECR"}, {"SOL", "SOR", "SOLSOR"},
           {"PFL", "ABR", "PFLABR"}, {"PFL", "BSR", "PFLBSR"}, {"PFL", "ECR", "PFLECR"},
           {"PFL", "SOR", "PFLSOR"}, {"ABH", "ABL", "ABHABL"}, {"ABH", "BSL", "ABHBSL"},
           {"ABH", "ECL", "ABHECL"}, {"ABH", "SOL", "ABHSOL"}, {"ABH", "PFL", "ABHPFL"},
           {"BSH", "ABL", "BSHABL"}, {"BSH", "BSL", "BSHBSL"}, {"BSH", "ECL", "BSHECL"},
           {"BSH", "SOL", "BSHSOL"}, {"BSH", "PFL", "BSHPFL"}, {"ECH", "ABL", "ECHABL"},
           {"ECH", "BSL", "ECHBSL"}, {"ECH", "ECL", "ECHECL"}, {"ECH", "SOL", "ECHSOL"},
           {"ECH", "PFL", "ECHPFL"}, {"SOH", "ABL", "SOHABL"}, {"SOH", "BSL", "SOHBSL"},
           {"SOH", "ECL", "SOHECL"}, {"SOH", "SOL", "SOHSOL"}, {"SOH", "PFL", "SOHPFL"}};
```

```
In[•]:= jopNum5 = Table[Join[mondef5[[i]], {co5[[i]]}], {i, 1, Length[co5]}];

In[•]:= cases5 = Join[#, {0.08}] & /@ jopNum5;
```

```
In[•]:= cases5[[1]]

Out[•]= {{{0., 1.45213, 0.0558333, 0.0275333, 0.00563333, 0., 0.332033, 0., 1.54763, 1.38923,
            0., 1.09573, 1.68953, 0.111533, 1.71103, 0.472933, 1.52293, 1.72973, 0., 0.0663333,
            0., 0., 0., 1.24703, 0., 1.19643, 0., 1.3985, 0., 0.249433, 0., 0.101733},
          {0., 1.3554, 0., 0., 0.0568, 0.2042, 0.1008, 0.2193, 1.0904, 1.2501, 0.,
            0.7716, 1.2029, 0.3542, 1.4165, 1.2416, 1.2154, 1.4971, 0., 0.2854, 0.1213,
            0., 0., 1.465, 0.1043, 1.1002, 0.3987, 1.0308, 0.6452, 1.4225, 0.579, 0.5922},
          {0., 1.4649, 0., 0.1804, 0., 0.1812, 0., 0.0349, 1.4533, 1.7979, 0., 0.9161,
            1.5431, 0.1288, 1.3499, 0.4616, 1.5108, 1.7813, 0., 0.133, 0.0432, 0., 0.,
            1.3831, 0.0128, 1.1756, 0.211, 0.9946, 0.2435, 0.8526, 0.2072, 0.0072}},
         {{0., 1.249, 0., 0.0308, 0., 0., 0., 0., 2.028, 1.7448, 0., 1.2377,
            1.8368, 0., 1.5429, 0.5805, 1.6243, 1.8364, 0., 0., 0., 0.0049, 0.,
            1.6348, 0., 1.2044, 0.0073, 1.0032, 0., 0.2871, 0.0161, 0.0641},
          {0., 2.0203, 0., 0.1249, 0., 0.0045, 0.0121, 0., 1.8791, 1.767, 0., 1.3343,
            1.7034, 0.0165, 1.8711, 1.4107, 1.6443, 2.1555, 0., 0., 0.0231, 0., 0.,
            2.3431, 0., 1.5422, 0.0239, 1.0968, 0.0483, 0.4767, 0.2905, 0.0553},
          {0., 1.8164, 0.0565, 0.0724, 0., 0.0145, 0.0043, 0., 2.064, 1.909, 0.,
            1.405, 1.8286, 0., 1.7039, 0.773, 1.91, 1.9312, 0., 0., 0.0098, 0.,
            0., 1.4047, 0., 1.2185, 0.0006, 0.8932, 0.0087, 0.2027, 0., 0.}},
         {{0.0056, 0.6609, 0.0049, 0., 0., 0.0243, 0., 0.0005, 0.0944, 0.2308, 0.,
            0.13, 1.3621, 0.0334, 0.9223, 0.2752, 0.7982, 1.0814, 0., 0.0132, 0., 0.,
            0., 0.8444, 0.0553, 0.2131, 0.0262, 0.3508, 0.0489, 0.1222, 0.0538, 0.0218},
          {0., 1.199, 0., 0., 0., 0.0227, 0., 0., 0.2352, 0.1025, 0., 0.2445,
            1.3634, 0.0458, 1.2273, 0.96, 0.7909, 0.9286, 0., 0., 0., 0.,
            0., 0.4847, 0., 0.1501, 0., 0.7405, 0., 0.4477, 0.2477, 0.},
          {0., 0.4919, 0., 0.0098, 0.0174, 0.1083, 0.0256, 0.0167, 0.2629, 0.2311, 0., 0.5108,
            1.0792, 0.0358, 0.8067, 0.7046, 0.8935, 0.8587, 0., 1.38778×10^-17, 0.0701, 0., 0.,
            1.1164, 0.0234, 0.2925, 0.0098, 0.5508, 0.0349, 0.2484, 0.0878, 0.0429}}, 0.08}

In[•]:= nicheOverlap[ECR5, SOH5, ECRSOH5, 0.08]

Out[•]= {24, 12, 9, 17, 7, 2, 10}

In[•]:= nicheOverlap[ABR5, ABH5, ABRABH5, 0.08]

Out[•]= {25, 15, 15, 15, 14, 1, 1}
```

```
In[•]:= nicheOverlap[{par1_, par2_, coul_, threashold_}] := (
          th1 = If[# ≥ threashold, 1, 0] &;
          bit1 = If[# ≥ 1, 1, 0] &;

          qf1 = Map[th1, par1, {2}];
          v1 = bit1 /@ Total[qf1];

          qf2 = Map[th1, par2, {2}];
          v2 = bit1 /@ Total[qf2];

          overlapMono = v1 v2;

          qfcocul = Map[th1, coul, {2}];
          vcocul = bit1 /@ Total[qfcocul];

          MoICoIntersection = overlapMono vcocul;

          contractxk = Total[overlapMono] - Total[MoICoIntersection];
          expandxk = Total[vcocul] - Total[MoICoIntersection];

          contract = funNe[contractxk];
          expans = funNe[expandxk];

          {Total[v1], Total[v2], Total[overlapMono],
           Total[vcocul], Total[MoICoIntersection], contract, expans}


         )
```

```
(*Total P1,       Total P2,       MonoIntersection,
TotalCo,     MoICoIntersection,    contract,  expans*)
```

```
(*   1,             2,                      3,
4,                 5,                6,        7*)
```

```
In[•]:= gk5 = Map[nicheOverlap, cases5, {1}]

Out[•]= {{25, 15, 15, 15, 14, 1, 1}, {25, 1, 1, 1, 0, 1, 1}, {25, 23, 21, 19, 18, 3, 1},
         {25, 12, 10, 22, 10, 0, 12}, {1, 15, 1, 18, 1, 0, 17}, {1, 1, 0, 1, 0, 0, 1},
         {1, 23, 1, 17, 1, 0, 16}, {1, 12, 0, 16, 0, 0, 16}, {24, 15, 15, 16, 14, 1, 2},
         {24, 1, 1, 8, 1, 0, 7}, {24, 23, 21, 16, 16, 5, 0}, {24, 12, 9, 17, 7, 2, 10},
         {25, 15, 12, 25, 10, 2, 15}, {25, 1, 1, 2, 0, 1, 2}, {25, 23, 20, 27, 17, 3, 10},
         {25, 12, 12, 5, 5, 7, 0}, {15, 17, 14, 16, 14, 0, 2}, {15, 21, 15, 16, 14, 1, 2},
         {15, 11, 7, 21, 7, 0, 14}, {15, 28, 14, 24, 14, 0, 10}, {1, 17, 1, 6, 0, 1, 6},
         {1, 21, 1, 17, 1, 0, 16}, {1, 11, 0, 6, 0, 0, 6}, {1, 28, 1, 4, 1, 0, 3},
         {23, 17, 17, 15, 15, 2, 0}, {23, 21, 19, 15, 15, 4, 0}, {23, 11, 9, 13, 6, 3, 7},
         {23, 28, 21, 21, 16, 5, 5}, {12, 17, 7, 11, 4, 3, 7}, {12, 21, 9, 13, 5, 4, 8},
         {12, 11, 8, 7, 3, 5, 4}, {12, 28, 11, 22, 9, 2, 13}, {17, 25, 17, 16, 13, 4, 3},
         {17, 1, 1, 7, 1, 0, 6}, {17, 24, 17, 16, 14, 3, 2}, {17, 25, 14, 6, 2, 12, 4},
         {21, 25, 21, 15, 15, 6, 0}, {21, 1, 1, 11, 1, 0, 10}, {21, 24, 21, 12, 12, 9, 0},
         {21, 25, 18, 12, 8, 10, 4}, {11, 25, 9, 16, 7, 2, 9}, {11, 1, 0, 15, 0, 0, 15},
         {11, 24, 9, 16, 8, 1, 8}, {11, 25, 11, 9, 6, 5, 3}, {28, 25, 23, 23, 19, 4, 4},
         {28, 1, 1, 7, 0, 1, 7}, {28, 24, 22, 28, 21, 1, 7}, {28, 25, 24, 8, 8, 16, 0},
         {17, 13, 12, 11, 10, 2, 1}, {17, 4, 4, 10, 0, 4, 10}, {17, 19, 16, 12, 12, 4, 0},
         {17, 21, 13, 11, 8, 5, 3}, {17, 30, 17, 5, 5, 12, 0}, {21, 13, 12, 14, 11, 1, 3},
         {21, 4, 4, 11, 2, 2, 9}, {21, 19, 17, 12, 12, 5, 0}, {21, 21, 15, 18, 10, 5, 8},
         {21, 30, 21, 13, 12, 9, 1}, {11, 13, 6, 15, 5, 1, 10}, {11, 4, 2, 12, 1, 1, 11},
         {11, 19, 8, 11, 8, 0, 3}, {11, 21, 11, 7, 4, 7, 3}, {11, 30, 11, 6, 4, 7, 2},
         {28, 13, 12, 20, 11, 1, 9}, {28, 4, 4, 5, 3, 1, 2}, {28, 19, 18, 22, 15, 3, 7},
         {28, 21, 19, 9, 6, 13, 3}, {28, 30, 28, 16, 15, 13, 1}, {13, 25, 13, 18, 12, 1, 6},
         {13, 1, 1, 13, 1, 0, 12}, {13, 24, 13, 16, 11, 2, 5}, {13, 25, 11, 22, 10, 1, 12},
         {4, 25, 4, 14, 4, 0, 10}, {4, 1, 0, 6, 0, 0, 6}, {4, 24, 4, 10, 3, 1, 7},
         {4, 25, 4, 3, 0, 4, 3}, {19, 25, 18, 16, 14, 4, 2}, {19, 1, 1, 7, 1, 0, 6},
         {19, 24, 18, 12, 12, 6, 0}, {19, 25, 16, 10, 10, 6, 0}, {21, 25, 17, 16, 12, 5, 4},
         {21, 1, 1, 14, 1, 0, 13}, {21, 24, 17, 19, 11, 6, 8}, {21, 25, 18, 5, 5, 13, 0},
         {30, 25, 25, 20, 20, 5, 0}, {30, 1, 1, 8, 1, 0, 7}, {30, 24, 24, 15, 14, 10, 1},
         {30, 25, 25, 10, 8, 17, 2}, {15, 13, 11, 18, 11, 0, 7}, {15, 4, 4, 14, 3, 1, 11},
         {15, 19, 15, 15, 14, 1, 1}, {15, 21, 11, 17, 10, 1, 7}, {15, 30, 15, 18, 14, 1, 4},
         {1, 13, 0, 16, 0, 0, 16}, {1, 4, 1, 3, 0, 1, 3}, {1, 19, 1, 5, 1, 0, 4},
         {1, 21, 0, 5, 0, 0, 5}, {1, 30, 1, 8, 1, 0, 7}, {23, 13, 13, 18, 13, 0, 5},
         {23, 4, 4, 16, 3, 1, 13}, {23, 19, 19, 17, 16, 3, 1}, {23, 21, 18, 16, 13, 5, 3},
         {23, 30, 23, 15, 14, 9, 1}, {12, 13, 6, 19, 6, 0, 13}, {12, 4, 2, 1, 0, 2, 1},
         {12, 19, 8, 8, 5, 3, 3}, {12, 21, 10, 10, 5, 5, 5}, {12, 30, 12, 5, 5, 7, 0}}
```

In[∘]:=

```
nicheOverlapDistances[{par1_, par2_, coul_, threashold_}] := (
  th1 = If[# ≥ threashold, 1, 0] &;
  bit1 = If[# ≥ 1, 1, 0] &;
  UnionFunTwoToOne = If[# == 2, 1, #] &;  (*NewNewNewNewNewNewNewNewNewNew*)

  qf1 = Map[th1, par1, {2}];
  v1 = bit1 /@ Total[qf1];

  qf2 = Map[th1, par2, {2}];
  v2 = bit1 /@ Total[qf2];

  overlapMono = v1 v2;

  UnionP1P2 = Map[UnionFunTwoToOne , (v1 + v2), {1}];
  (*NewNewNewNewNewNewNewNewNewNew*)

  qfcocul = Map[th1, coul, {2}];
  vcocul = bit1 /@ Total[qfcocul];

  MoICoIntersection = overlapMono vcocul;

  contractxk = Total[overlapMono] - Total[MoICoIntersection];
  expandxk = Total[vcocul] - Total[MoICoIntersection];

  contract = funNe[contractxk];
  expans = funNe[expandxk];

  (*
   SuperNEvector= vcocul + UnionP1P2 ; (*NewNewNewNewNewNewNewNewNewNew*)
  SuperNEnumber=Count[SuperNEvector,1]; (*NewNewNewNewNewNewNewNewNewNew*)
  *)

  SuperNEnumber = Count[Partition[Riffle[vcocul, UnionP1P2], {2}], {1, 0}];


  {Total[v1], Total[v2], Total[overlapMono], Total[vcocul],
   Total[MoICoIntersection], contract, expans, SuperNEnumber}


 )
```

```
(*Total P1,       Total P2,       MonoIntersection,
TotalCo,     MoICoIntersection,    contract,  expans*)
```

```
(*   1,              2,                       3,
4,                  5,                 6,        7*)

In[•]:= gk5KK = Map[nicheOverlapDistances, cases5, {1}]

Out[•]= {{25, 15, 15, 15, 14, 1, 1, 0}, {25, 1, 1, 1, 0, 1, 1, 0}, {25, 23, 21, 19, 18, 3, 1, 0},
  {25, 12, 10, 22, 10, 0, 12, 0}, {1, 15, 1, 18, 1, 0, 17, 5}, {1, 1, 0, 1, 0, 0, 1, 1},
  {1, 23, 1, 17, 1, 0, 16, 1}, {1, 12, 0, 16, 0, 0, 16, 8}, {24, 15, 15, 16, 14, 1, 2, 0},
  {24, 1, 1, 8, 1, 0, 7, 1}, {24, 23, 21, 16, 16, 5, 0, 0}, {24, 12, 9, 17, 7, 2, 10, 0},
  {25, 15, 12, 25, 10, 2, 15, 1}, {25, 1, 1, 2, 0, 1, 2, 0}, {25, 23, 20, 27, 17, 3, 10, 2},
  {25, 12, 12, 5, 5, 7, 0, 0}, {15, 17, 14, 16, 14, 0, 2, 1}, {15, 21, 15, 16, 14, 1, 2, 0},
  {15, 11, 7, 21, 7, 0, 14, 5}, {15, 28, 14, 24, 14, 0, 10, 1}, {1, 17, 1, 6, 0, 1, 6, 1},
  {1, 21, 1, 17, 1, 0, 16, 3}, {1, 11, 0, 6, 0, 0, 6, 2}, {1, 28, 1, 4, 1, 0, 3, 0},
  {23, 17, 17, 15, 15, 2, 0, 0}, {23, 21, 19, 15, 15, 4, 0, 0}, {23, 11, 9, 13, 6, 3, 7, 0},
  {23, 28, 21, 21, 16, 5, 5, 0}, {12, 17, 7, 11, 4, 3, 7, 1}, {12, 21, 9, 13, 5, 4, 8, 1},
  {12, 11, 8, 7, 3, 5, 4, 2}, {12, 28, 11, 22, 9, 2, 13, 0}, {17, 25, 17, 16, 13, 4, 3, 0},
  {17, 1, 1, 7, 1, 0, 6, 0}, {17, 24, 17, 16, 14, 3, 2, 0}, {17, 25, 14, 6, 2, 12, 4, 0},
  {21, 25, 21, 15, 15, 6, 0, 0}, {21, 1, 1, 11, 1, 0, 10, 1}, {21, 24, 21, 12, 12, 9, 0, 0},
  {21, 25, 18, 12, 8, 10, 4, 0}, {11, 25, 9, 16, 7, 2, 9, 0}, {11, 1, 0, 15, 0, 0, 15, 7},
  {11, 24, 9, 16, 8, 1, 8, 1}, {11, 25, 11, 9, 6, 5, 3, 0}, {28, 25, 23, 23, 19, 4, 4, 0},
  {28, 1, 1, 7, 0, 1, 7, 0}, {28, 24, 22, 28, 21, 1, 7, 0}, {28, 25, 24, 8, 8, 16, 0, 0},
  {17, 13, 12, 11, 10, 2, 1, 0}, {17, 4, 4, 10, 0, 4, 10, 3}, {17, 19, 16, 12, 12, 4, 0, 0},
  {17, 21, 13, 11, 8, 5, 3, 0}, {17, 30, 17, 5, 5, 12, 0, 0}, {21, 13, 12, 14, 11, 1, 3, 0},
  {21, 4, 4, 11, 2, 2, 9, 0}, {21, 19, 17, 12, 12, 5, 0, 0}, {21, 21, 15, 18, 10, 5, 8, 1},
  {21, 30, 21, 13, 12, 9, 1, 0}, {11, 13, 6, 15, 5, 1, 10, 3}, {11, 4, 2, 12, 1, 1, 11, 3},
  {11, 19, 8, 11, 8, 0, 3, 1}, {11, 21, 11, 7, 4, 7, 3, 1}, {11, 30, 11, 6, 4, 7, 2, 0},
  {28, 13, 12, 20, 11, 1, 9, 1}, {28, 4, 4, 5, 3, 1, 2, 0}, {28, 19, 18, 22, 15, 3, 7, 1},
  {28, 21, 19, 9, 6, 13, 3, 0}, {28, 30, 28, 16, 15, 13, 1, 0},
  {13, 25, 13, 18, 12, 1, 6, 1}, {13, 1, 1, 13, 1, 0, 12, 3}, {13, 24, 13, 16, 11, 2, 5, 1},
  {13, 25, 11, 22, 10, 1, 12, 1}, {4, 25, 4, 14, 4, 0, 10, 1}, {4, 1, 0, 6, 0, 0, 6, 5},
  {4, 24, 4, 10, 3, 1, 7, 1}, {4, 25, 4, 3, 0, 4, 3, 0}, {19, 25, 18, 16, 14, 4, 2, 0},
  {19, 1, 1, 7, 1, 0, 6, 1}, {19, 24, 18, 12, 12, 6, 0, 0}, {19, 25, 16, 10, 10, 6, 0, 0},
  {21, 25, 17, 16, 12, 5, 4, 0}, {21, 1, 1, 14, 1, 0, 13, 2}, {21, 24, 17, 19, 11, 6, 8, 1},
  {21, 25, 18, 5, 5, 13, 0, 0}, {30, 25, 25, 20, 20, 5, 0, 0}, {30, 1, 1, 8, 1, 0, 7, 0},
  {30, 24, 24, 15, 14, 10, 1, 0}, {30, 25, 25, 10, 8, 17, 2, 0},
  {15, 13, 11, 18, 11, 0, 7, 3}, {15, 4, 4, 14, 3, 1, 11, 2}, {15, 19, 15, 15, 14, 1, 1, 1},
  {15, 21, 11, 17, 10, 1, 7, 2}, {15, 30, 15, 18, 14, 1, 4, 0},
  {1, 13, 0, 16, 0, 0, 16, 2}, {1, 4, 1, 3, 0, 1, 3, 2}, {1, 19, 1, 5, 1, 0, 4, 1},
  {1, 21, 0, 5, 0, 0, 5, 1}, {1, 30, 1, 8, 1, 0, 7, 0}, {23, 13, 13, 18, 13, 0, 5, 2},
  {23, 4, 4, 16, 3, 1, 13, 0}, {23, 19, 19, 17, 16, 3, 1, 0}, {23, 21, 18, 16, 13, 5, 3, 1},
  {23, 30, 23, 15, 14, 9, 1, 0}, {12, 13, 6, 19, 6, 0, 13, 3}, {12, 4, 2, 1, 0, 2, 1, 0},
  {12, 19, 8, 8, 5, 3, 3, 0}, {12, 21, 10, 10, 5, 5, 5, 1}, {12, 30, 12, 5, 5, 7, 0, 0}}
```

Quantifying Niche Expansion and Contraction

Expansion vs. Contraction

In[•]:=

```
fuk[x_] :=
 {Table[{x[[1]][[i]], Style[x[[1]][[i]], FontFamily → x[[3]], Black, x[[4]],
     x[[5]]], {0, x[[6]]}, ■}, {i, 1, Length[x[[1]]]}],

  Table[{x[[2]][[i]], Style[x[[2]][[i]], FontFamily → x[[3]],
     Black, x[[4]], x[[5]]], {0, x[[6]]}, ■}, {i, 1, Length[x[[2]]]}]

 }
```

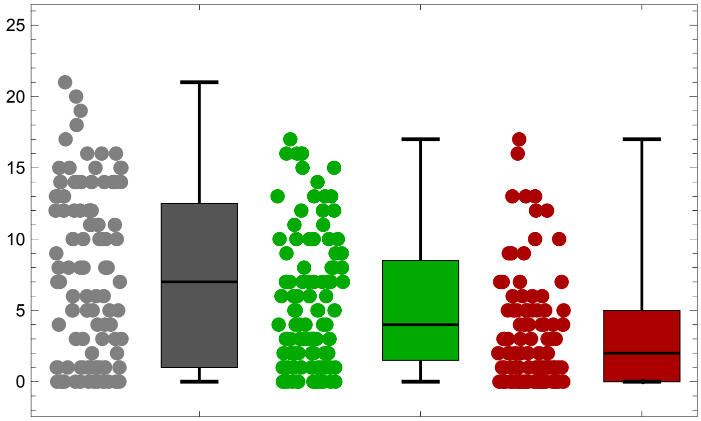
In[•]:=

```
{cf = Graphics[{EdgeForm[{Black}], FaceForm[White], Disk[]}],
  c0 = Graphics[{EdgeForm[{Black}], FaceForm[Gray], Disk[]}],
  tp = Graphics[{FaceForm[White],
     EdgeForm[Directive[Black]], Triangle[{{0, 0}, {1, 2}, {2, 0}}]}],
  tp0 = Graphics[{FaceForm[Gray], EdgeForm[Directive[Black]],
     Triangle[{{0, 0}, {1, 2}, {2, 0}}]}]};
```

In[•]:=

```
a1 = BoxWhiskerChart[{#[[5]] & /@ gk5, #[[7]] & /@ gk5, #[[6]] & /@ gk5},
   {{"MedianMarker", 1, Thickness[0.004]}, {"Whiskers", Thickness[0.004]},
    {"Fences", Thick}}, ChartBaseStyle → EdgeForm[Dashing[0.99]],
   ChartStyle → {{Darker[Gray], Darker[Green], Darker[Red]}},
   Frame → True, FrameTicks → {None, {5, 10, 20}, None, None},
   BarSpacing → 1.9, PlotRange → {{0.39, 3.1}, {-1, 25}}];

wq1 := RandomReal[{-0.15, 0.15}]
jj1 = Table[0.5 + wq1, Length[#[[5]] & /@ gk5]];
jj2 = Table[1.5 + wq1, Length[#[[7]] & /@ gk5]];
jj3 = Table[2.5 + wq1, Length[#[[6]] & /@ gk5]];

a2 = ListPlot[Partition[Riffle[jj1, #[[5]] & /@ gk5], {2}], PlotMarkers →
    {Graphics[{EdgeForm[{Gray}], FaceForm[Gray], Disk[]}], Scaled[0.035]},
   PlotStyle → Black, PlotRange → {{0, 4}, {0, 25}}];
a3 = ListPlot[Partition[Riffle[jj2, #[[7]] & /@ gk5], {2}], PlotMarkers →
    {Graphics[{EdgeForm[{Darker[Green]}], FaceForm[Darker[Green]], Disk[]}],
     Scaled[0.035]}, PlotStyle → Black, PlotRange → {{0, 4}, {0, 25}}];
a4 = ListPlot[Partition[Riffle[jj3, #[[6]] & /@ gk5], {2}], PlotMarkers →
    {Graphics[{EdgeForm[{Darker[Red]}], FaceForm[Darker[Red]], Disk[]}],
     Scaled[0.035]}, PlotStyle → Black, PlotRange → {{0, 4}, {0, 25}}];
Show[
 a1,
 a2,
 a3,
 a4]
```

Out[•]=



Correlation Niche Expansion and Contraction

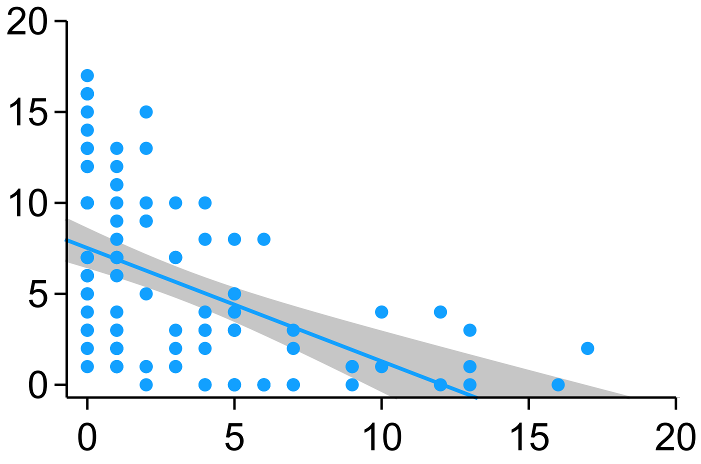
```
In[•]:= predictted = #[[5]] & /@ gk5;
        expanssion = #[[7]] & /@ gk5;
        contracction = #[[6]] & /@ gk5;
```

```
In[•]:= SpearmanRankTest[contracction, expanssion, "TestDataTable"]
```

```
Out[•]=                | Statistic   P-Value
        Spearman Rank  | -0.578852   5.31132×10^-11
```

```
In[•]:= ECPairs5 = Partition[Riffle[contracction, expanssion], {2}];
        lmEC5 = LinearModelFit[ECPairs5, x, x];
        bands90EC5[x_] = lmEC5["MeanPredictionBands", ConfidenceLevel → .95];

        ExpContr1 = Plot[{lmEC5[x], bands90EC5[x]}, {x, -0.7, 25},
           PlotStyle → {Directive[RGBColor[18 / (255), 160 / (255), 255 / (255)], Thick],
             Lighter[Lighter[Gray]]}, Filling → {2 → {{1}, Lighter[Lighter[Gray]]}},
           PlotRange → {{-0.7, 20}, {-0.7, 20}},
           Ticks → fuk[{{0, 5, 10, 15, 20}, {0, 5, 10, 15, 20}, "Arial", Plain, 20, 0.02}],
           TicksStyle → Thickness[0.004],
           AxesStyle → Thickness[0.004], AxesOrigin → {-0.7, -0.7}];

        blN = Graphics[{EdgeForm[{RGBColor[18 / (255), 160 / (255), 255 / (255)]}],
            FaceForm[RGBColor[18 / (255), 160 / (255), 255 / (255)]], Disk[]}];
        blG = Graphics[{EdgeForm[{Gray}], FaceForm[
             RGBColor[18 / (255), 160 / (255), 255 / (255)]], Disk[]}];
        blB = Graphics[{EdgeForm[{Black}], FaceForm[
             RGBColor[18 / (255), 160 / (255), 255 / (255)]], Disk[]}];

        ExpContr2 = ListPlot[ECPairs5, PlotMarkers → {blN, Scaled[0.035]}];
        Show[ExpContr1, ExpContr2]
```

Out[•]=



Niche Expansion and contraction as a function of niche (dis)similarity

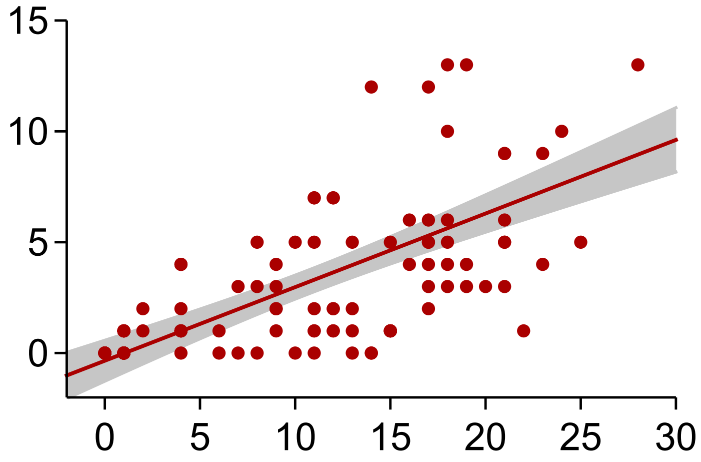
Ecological proxies of Niche Expansion and Contraction: Niche Intersection-Union

```
(*Total P1,       Total P2,       MonoIntersection,
TotalCo,     MoICoIntersection,    contract,  expans*)

(*   1,              2,                      3,
4,                 5,                 6,         7*)
```

In[∘]:=

```
ECIntersContraction5 = Partition[Riffle[#[[3]] & /@ gk5, #[[6]] & /@ gk5], {2}];

lmECIntersContraction5 = LinearModelFit[ECIntersContraction5, x, x];

bands90EClmECIntersContraction5[x_] =
  lmECIntersContraction5["MeanPredictionBands", ConfidenceLevel → .95];



NC1 = Plot[
   {lmECIntersContraction5[x], bands90EClmECIntersContraction5[x]}, {x, -2, 30},
   PlotStyle → {Directive[Darker[Red], Thick], Lighter[Lighter[Gray]]}, Filling →
    {2 → {{1}, Lighter[Lighter[Gray]]}}, PlotRange → {{-2, 30}, {-2, 15}},
   Ticks → fuk[{{0, 5, 10, 15, 20, 25, 30}, {0, 5, 10, 15, 20, 25, 30},
      "Arial", Plain, 20, 0.02}], TicksStyle → Thickness[0.004],
   AxesStyle → Thickness[0.004], AxesOrigin → {-2, -2}];

rdN = Graphics[{EdgeForm[{Darker[Red]}], FaceForm[Darker[Red]], Disk[]}];
rdG = Graphics[{EdgeForm[{Gray}], FaceForm[Darker[Red]], Disk[]}];
rdB = Graphics[{EdgeForm[{Black}], FaceForm[Darker[Red]], Disk[]}];

NC2 = ListPlot[ECIntersContraction5, PlotMarkers → {rdN, Scaled[0.035]}];
Show[NC1, NC2]
```

Out[∘]=



```
(* Union minus Intersection *)
```

In[∘]:=

```
umi5 = ((#[[1]] & /@ gk5) + (#[[2]] & /@ gk5)) - 2 (#[[3]] & /@ gk5)
```

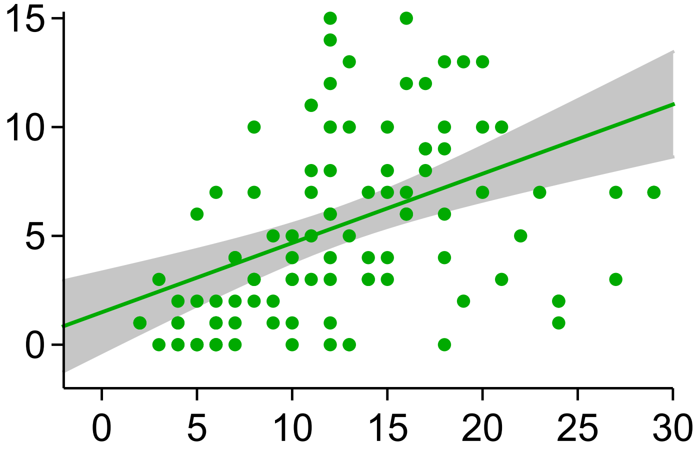
Out[∘]=

```
{10, 24, 6, 17, 14, 2, 22, 13, 9, 23, 5, 18, 16, 24, 8, 13, 4, 6, 12, 15, 16,
 20, 12, 27, 6, 6, 16, 9, 15, 15, 7, 18, 8, 16, 7, 14, 4, 20, 3, 10, 18, 12, 17,
 14, 7, 27, 8, 5, 6, 13, 4, 12, 13, 10, 17, 6, 12, 9, 12, 11, 14, 10, 19, 17, 24,
 11, 11, 2, 12, 12, 11, 16, 21, 5, 20, 21, 8, 18, 7, 12, 12, 20, 11, 10, 5, 29,
 6, 5, 6, 11, 4, 14, 15, 14, 3, 18, 22, 29, 10, 19, 4, 8, 7, 13, 12, 15, 13, 18}
```

In[∘]:=

```
ECUIExpansion5 = Partition[Riffle[umi5, #[[7]] & /@ gk5], {2}];

lmECUIExpansion5 = LinearModelFit[ECUIExpansion5, x, x];

bands90EClmECUIExpansion5[x_] =
  lmECUIExpansion5["MeanPredictionBands", ConfidenceLevel → .95];

NE1 = Plot[{lmECUIExpansion5[x], bands90EClmECUIExpansion5[x]}, {x, -2, 30},
   PlotStyle →  {Directive[Darker[Green], Thick], Lighter[Lighter[Gray]]},
   Filling → {2 → {{1}, Lighter[Lighter[Gray]]}},
   PlotRange →  {{-2, 30}, {-2, 15.3}},
   Ticks → fuk[{{0, 5, 10, 15, 20, 25, 30}, {0, 5, 10, 15, 20, 25, 30},
      "Arial", Plain, 20, 0.02}], TicksStyle → Thickness[0.004],
   AxesStyle → Thickness[0.004], AxesOrigin → {-2, -2}];

grN = Graphics[{EdgeForm[{Darker[Green]}], FaceForm[Darker[Green]], Disk[]}];
grG = Graphics[{EdgeForm[{Gray}], FaceForm[Darker[Green]], Disk[]}];
grB = Graphics[{EdgeForm[{Black}], FaceForm[Darker[Green]], Disk[]}];

NE2 = ListPlot[ECUIExpansion5, PlotMarkers → {grN, Scaled[0.035]}];
Show[NE1, NE2]
```

Out[∘]=



Jaccard Distance

```
In[•]:= nicheOverlapDistancesJac[{par1_, par2_, coul_, threashold_}] := (
          th1 = If[# ≥ threashold, 1, 0] &;
          bit1 = If[# ≥ 1, 1, 0] &;
          UnionFunTwoToOne = If[# == 2, 1, #] &;  (*NewNewNewNewNewNewNewNewNewNew*)

          qf1 = Map[th1, par1, {2}];
          v1 = bit1 /@ Total[qf1];

          qf2 = Map[th1, par2, {2}];
          v2 = bit1 /@ Total[qf2];

          overlapMono = v1 v2;

          UnionP1P2  = Map[UnionFunTwoToOne , (v1 + v2), {1}];
          (*NewNewNewNewNewNewNewNewNewNew*)

          qfcocul = Map[th1, coul, {2}];
          vcocul = bit1 /@ Total[qfcocul];

          MoICoIntersection = overlapMono vcocul;

          contractxk = Total[overlapMono] - Total[MoICoIntersection];
          expandxk = Total[vcocul] - Total[MoICoIntersection];

          contract = funNe[contractxk];
          expans = funNe[expandxk];

          (*
           SuperNEvector= vcocul + UnionP1P2 ; (*NewNewNewNewNewNewNewNewNewNew*)
          SuperNEnumber=Count[SuperNEvector,1];  (*NewNewNewNewNewNewNewNewNewNew*)
          *)

          SuperNEnumber = Count[Partition[Riffle[vcocul, UnionP1P2], {2}], {1, 0}];

          Jac = N[Total[overlapMono] / Total[UnionP1P2]];
          InvJac = N[(Total[UnionP1P2] - Total[overlapMono]) / Total[UnionP1P2]];


          {Total[v1], Total[v2], Total[overlapMono], Total[vcocul],
           Total[MoICoIntersection], contract, expans, SuperNEnumber, Jac, InvJac}
```

```
 )
```

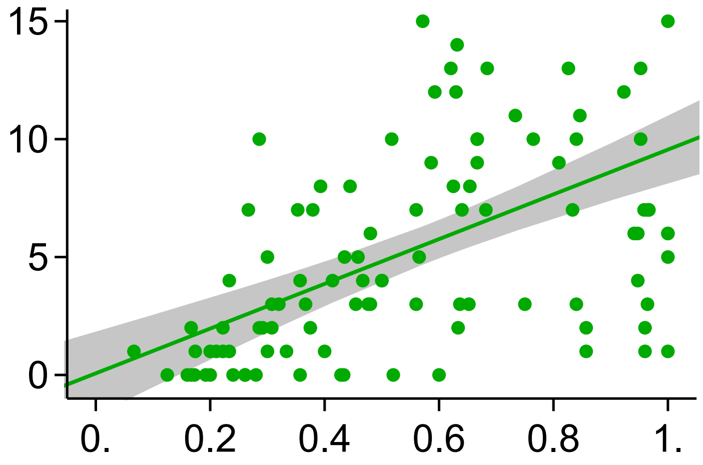
```
(*Total P1,       Total P2,       MonoIntersection,    TotalCo,
MoICoIntersection,    contract,  expans,  SuperExpans,  Jacc,  InvJacc*)

(*   1,              2,                         3,               4,
5,                 6,         7,         8,           9,      10*)
```

```
In[•]:= gk5KJaccK = Map[nicheOverlapDistancesJac, cases5, {1}]

Out[•]= {{25, 15, 15, 15, 14, 1, 1, 0, 0.6, 0.4}, {25, 1, 1, 1, 0, 1, 1, 0, 0.04, 0.96},
  {25, 23, 21, 19, 18, 3, 1, 0, 0.777778, 0.222222},
  {25, 12, 10, 22, 10, 0, 12, 0, 0.37037, 0.62963},
  {1, 15, 1, 18, 1, 0, 17, 5, 0.0666667, 0.933333},
  {1, 1, 0, 1, 0, 0, 1, 1, 0., 1.}, {1, 23, 1, 17, 1, 0, 16, 1, 0.0434783, 0.956522},
  {1, 12, 0, 16, 0, 0, 16, 8, 0., 1.}, {24, 15, 15, 16, 14, 1, 2, 0, 0.625, 0.375},
  {24, 1, 1, 8, 1, 0, 7, 1, 0.0416667, 0.958333},
  {24, 23, 21, 16, 16, 5, 0, 0, 0.807692, 0.192308},
  {24, 12, 9, 17, 7, 2, 10, 0, 0.333333, 0.666667},
  {25, 15, 12, 25, 10, 2, 15, 1, 0.428571, 0.571429},
  {25, 1, 1, 2, 0, 1, 2, 0, 0.04, 0.96}, {25, 23, 20, 27, 17, 3, 10, 2, 0.714286, 0.285714},
  {25, 12, 12, 5, 5, 7, 0, 0, 0.48, 0.52},
  {15, 17, 14, 16, 14, 0, 2, 1, 0.777778, 0.222222},
  {15, 21, 15, 16, 14, 1, 2, 0, 0.714286, 0.285714},
  {15, 11, 7, 21, 7, 0, 14, 5, 0.368421, 0.631579},
  {15, 28, 14, 24, 14, 0, 10, 1, 0.482759, 0.517241},
  {1, 17, 1, 6, 0, 1, 6, 1, 0.0588235, 0.941176},
  {1, 21, 1, 17, 1, 0, 16, 3, 0.047619, 0.952381},
  {1, 11, 0, 6, 0, 0, 6, 2, 0., 1.}, {1, 28, 1, 4, 1, 0, 3, 0, 0.0357143, 0.964286},
  {23, 17, 17, 15, 15, 2, 0, 0, 0.73913, 0.26087},
  {23, 21, 19, 15, 15, 4, 0, 0, 0.76, 0.24}, {23, 11, 9, 13, 6, 3, 7, 0, 0.36, 0.64},
  {23, 28, 21, 21, 16, 5, 5, 0, 0.7, 0.3}, {12, 17, 7, 11, 4, 3, 7, 1, 0.318182, 0.681818},
  {12, 21, 9, 13, 5, 4, 8, 1, 0.375, 0.625}, {12, 11, 8, 7, 3, 5, 4, 2, 0.533333, 0.466667},
  {12, 28, 11, 22, 9, 2, 13, 0, 0.37931, 0.62069},
  {17, 25, 17, 16, 13, 4, 3, 0, 0.68, 0.32}, {17, 1, 1, 7, 1, 0, 6, 0, 0.0588235, 0.941176},
  {17, 24, 17, 16, 14, 3, 2, 0, 0.708333, 0.291667},
  {17, 25, 14, 6, 2, 12, 4, 0, 0.5, 0.5}, {21, 25, 21, 15, 15, 6, 0, 0, 0.84, 0.16},
  {21, 1, 1, 11, 1, 0, 10, 1, 0.047619, 0.952381},
  {21, 24, 21, 12, 12, 9, 0, 0, 0.875, 0.125},
  {21, 25, 18, 12, 8, 10, 4, 0, 0.642857, 0.357143},
  {11, 25, 9, 16, 7, 2, 9, 0, 0.333333, 0.666667}, {11, 1, 0, 15, 0, 0, 15, 7, 0., 1.},
  {11, 24, 9, 16, 8, 1, 8, 1, 0.346154, 0.653846}, {11, 25, 11, 9, 6, 5, 3, 0, 0.44, 0.56},
  {28, 25, 23, 23, 19, 4, 4, 0, 0.766667, 0.233333},
  {28, 1, 1, 7, 0, 1, 7, 0, 0.0357143, 0.964286},
  {28, 24, 22, 28, 21, 1, 7, 0, 0.733333, 0.266667},
{28, 25, 24, 8, 8, 16, 0, 0, 0.827586, 0.172414},
{17, 13, 12, 11, 10, 2, 1, 0, 0.666667, 0.333333},
{17, 4, 4, 10, 0, 4, 10, 3, 0.235294, 0.764706},
{17, 19, 16, 12, 12, 4, 0, 0, 0.8, 0.2}, {17, 21, 13, 11, 8, 5, 3, 0, 0.52, 0.48},
{17, 30, 17, 5, 5, 12, 0, 0, 0.566667, 0.433333},
{21, 13, 12, 14, 11, 1, 3, 0, 0.545455, 0.454545},
{21, 4, 4, 11, 2, 2, 9, 0, 0.190476, 0.809524},
{21, 19, 17, 12, 12, 5, 0, 0, 0.73913, 0.26087},
{21, 21, 15, 18, 10, 5, 8, 1, 0.555556, 0.444444},
{21, 30, 21, 13, 12, 9, 1, 0, 0.7, 0.3}, {11, 13, 6, 15, 5, 1, 10, 3, 0.333333, 0.666667},
{11, 4, 2, 12, 1, 1, 11, 3, 0.153846, 0.846154},
{11, 19, 8, 11, 8, 0, 3, 1, 0.363636, 0.636364},
{11, 21, 11, 7, 4, 7, 3, 1, 0.52381, 0.47619},
{11, 30, 11, 6, 4, 7, 2, 0, 0.366667, 0.633333},
{28, 13, 12, 20, 11, 1, 9, 1, 0.413793, 0.586207},
{28, 4, 4, 5, 3, 1, 2, 0, 0.142857, 0.857143},
{28, 19, 18, 22, 15, 3, 7, 1, 0.62069, 0.37931},
{28, 21, 19, 9, 6, 13, 3, 0, 0.633333, 0.366667},
{28, 30, 28, 16, 15, 13, 1, 0, 0.933333, 0.0666667},
{13, 25, 13, 18, 12, 1, 6, 1, 0.52, 0.48},
{13, 1, 1, 13, 1, 0, 12, 3, 0.0769231, 0.923077},
{13, 24, 13, 16, 11, 2, 5, 1, 0.541667, 0.458333},
{13, 25, 11, 22, 10, 1, 12, 1, 0.407407, 0.592593},
{4, 25, 4, 14, 4, 0, 10, 1, 0.16, 0.84}, {4, 1, 0, 6, 0, 0, 6, 5, 0., 1.},
{4, 24, 4, 10, 3, 1, 7, 1, 0.166667, 0.833333}, {4, 25, 4, 3, 0, 4, 3, 0, 0.16, 0.84},
{19, 25, 18, 16, 14, 4, 2, 0, 0.692308, 0.307692},
{19, 1, 1, 7, 1, 0, 6, 1, 0.0526316, 0.947368}, {19, 24, 18, 12, 12, 6, 0, 0, 0.72, 0.28},
{19, 25, 16, 10, 10, 6, 0, 0, 0.571429, 0.428571},
{21, 25, 17, 16, 12, 5, 4, 0, 0.586207, 0.413793},
{21, 1, 1, 14, 1, 0, 13, 2, 0.047619, 0.952381},
{21, 24, 17, 19, 11, 6, 8, 1, 0.607143, 0.392857},
{21, 25, 18, 5, 5, 13, 0, 0, 0.642857, 0.357143},
{30, 25, 25, 20, 20, 5, 0, 0, 0.833333, 0.166667},
{30, 1, 1, 8, 1, 0, 7, 0, 0.0333333, 0.966667}, {30, 24, 24, 15, 14, 10, 1, 0, 0.8, 0.2},
{30, 25, 25, 10, 8, 17, 2, 0, 0.833333, 0.166667},
{15, 13, 11, 18, 11, 0, 7, 3, 0.647059, 0.352941},
{15, 4, 4, 14, 3, 1, 11, 2, 0.266667, 0.733333},
{15, 19, 15, 15, 14, 1, 1, 1, 0.789474, 0.210526},
{15, 21, 11, 17, 10, 1, 7, 2, 0.44, 0.56}, {15, 30, 15, 18, 14, 1, 4, 0, 0.5, 0.5},
{1, 13, 0, 16, 0, 0, 16, 2, 0., 1.}, {1, 4, 1, 3, 0, 1, 3, 2, 0.25, 0.75},
{1, 19, 1, 5, 1, 0, 4, 1, 0.0526316, 0.947368},
{1, 21, 0, 5, 0, 0, 5, 1, 0., 1.}, {1, 30, 1, 8, 1, 0, 7, 0, 0.0333333, 0.966667},
{23, 13, 13, 18, 13, 0, 5, 2, 0.565217, 0.434783},
{23, 4, 4, 16, 3, 1, 13, 0, 0.173913, 0.826087},
{23, 19, 19, 17, 16, 3, 1, 0, 0.826087, 0.173913},
{23, 21, 18, 16, 13, 5, 3, 1, 0.692308, 0.307692},
{23, 30, 23, 15, 14, 9, 1, 0, 0.766667, 0.233333},
{12, 13, 6, 19, 6, 0, 13, 3, 0.315789, 0.684211},
{12, 4, 2, 1, 0, 2, 1, 0, 0.142857, 0.857143},
{12, 19, 8, 8, 5, 3, 3, 0, 0.347826, 0.652174},
{12, 21, 10, 10, 5, 5, 5, 1, 0.434783, 0.565217}, {12, 30, 12, 5, 5, 7, 0, 0, 0.4, 0.6}}
```

In[•]:=

```
ECUIExpansion5Jacc =
  Partition[Riffle[#[[10]] & /@ gk5KJaccK, #[[7]] & /@ gk5KJaccK], {2}];

lmECUIExpansion5Jacc = LinearModelFit[ECUIExpansion5Jacc, x, x];

bands90EClmECUIExpansion5Jacc[x_] =
  lmECUIExpansion5Jacc["MeanPredictionBands", ConfidenceLevel → .95];



Exp1 =
  Plot[{lmECUIExpansion5Jacc[x], bands90EClmECUIExpansion5Jacc[x]}, {x, -2, 30},
   PlotStyle → {Directive[Darker[Green], Thick], Lighter[Lighter[Gray]]},
   Filling → {2 → {{1}, Lighter[Lighter[Gray]]}},
   PlotRange → {{-0.05, 1.05}, {-1, 15.5}},
   Ticks → fuk[{{0.0, 0.2, 0.4, 0.6, 0.8, 1.0}, {0, 5, 10, 15, 20, 25, 30},
      "Arial", Plain, 20, 0.02}], TicksStyle → Thickness[0.004],
   AxesStyle → Thickness[0.004], AxesOrigin → {-0.05, -1}];

grN = Graphics[{EdgeForm[{Darker[Green]}], FaceForm[Darker[Green]], Disk[]}];
grG = Graphics[{EdgeForm[{Gray}], FaceForm[Darker[Green]], Disk[]}];
grB = Graphics[{EdgeForm[{Black}], FaceForm[Darker[Green]], Disk[]}];

Exp2 = ListPlot[ECUIExpansion5Jacc, PlotMarkers → {grN, Scaled[0.035]}];
Show[Exp1, Exp2]
```

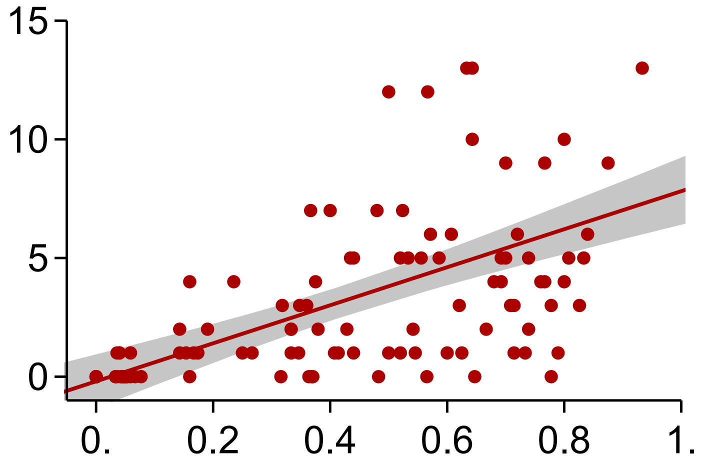
Out[●]=



In[●]:=

```
ECIntersContraction5Jacc =
  Partition[Riffle[#[[9]] & /@ gk5KJaccK, #[[6]] & /@ gk5KJaccK], {2}];

lmECIntersContraction5Jacc = LinearModelFit[ECIntersContraction5Jacc, x, x];

bands90EClmECIntersContraction5Jacc[x_] =
  lmECIntersContraction5Jacc["MeanPredictionBands", ConfidenceLevel → .95];

Contr1 = Plot[{lmECIntersContraction5Jacc[x],
    bands90EClmECIntersContraction5Jacc[x]}, {x, -2, 30},
   PlotStyle →  {Directive[Darker[Red], Thick], Lighter[Lighter[Gray]]},
   Filling → {2 → {{1}, Lighter[Lighter[Gray]]}},
   PlotRange →  {{-0.05, 1.0}, {-1, 15}},
   Ticks → fuk[{{0.0, 0.2,  0.4,  0.6,  0.8, 1.0}, {0, 5, 10, 15, 20, 25, 30},
      "Arial", Plain, 20, 0.02}], TicksStyle → Thickness[0.004],
   AxesStyle → Thickness[0.004], AxesOrigin → {-0.05, -1}];

rdN = Graphics[{EdgeForm[{Darker[Red]}], FaceForm[Darker[Red]], Disk[]}];
rdG = Graphics[{EdgeForm[{Gray}], FaceForm[Darker[Red]], Disk[]}];
rdB = Graphics[{EdgeForm[{Black}], FaceForm[Darker[Red]], Disk[]}];

Contr2 = ListPlot[ECIntersContraction5Jacc, PlotMarkers → {rdN, Scaled[0.035]}];
Show[Contr1, Contr2]
```

Out[●]=



Niche Expansion and contraction in specialists vs. generalists

Specialists - Generalists

```
In[•]:= genspeDat5 = {{#[[1]], #[[2]]}, {#[[7]]}} & /@ gk5
Out[•]= {{{25, 15}, {1}}, {{25, 1}, {1}}, {{25, 23}, {1}}, {{25, 12}, {12}}, {{1, 15}, {17}},
  {{1, 1}, {1}}, {{1, 23}, {16}}, {{1, 12}, {16}}, {{24, 15}, {2}}, {{24, 1}, {7}},
  {{24, 23}, {0}}, {{24, 12}, {10}}, {{25, 15}, {15}}, {{25, 1}, {2}}, {{25, 23}, {10}},
  {{25, 12}, {0}}, {{15, 17}, {2}}, {{15, 21}, {2}}, {{15, 11}, {14}},
  {{15, 28}, {10}}, {{1, 17}, {6}}, {{1, 21}, {16}}, {{1, 11}, {6}}, {{1, 28}, {3}},
  {{23, 17}, {0}}, {{23, 21}, {0}}, {{23, 11}, {7}}, {{23, 28}, {5}}, {{12, 17}, {7}},
  {{12, 21}, {8}}, {{12, 11}, {4}}, {{12, 28}, {13}}, {{17, 25}, {3}}, {{17, 1}, {6}},
  {{17, 24}, {2}}, {{17, 25}, {4}}, {{21, 25}, {0}}, {{21, 1}, {10}}, {{21, 24}, {0}},
  {{21, 25}, {4}}, {{11, 25}, {9}}, {{11, 1}, {15}}, {{11, 24}, {8}}, {{11, 25}, {3}},
  {{28, 25}, {4}}, {{28, 1}, {7}}, {{28, 24}, {7}}, {{28, 25}, {0}}, {{17, 13}, {1}},
  {{17, 4}, {10}}, {{17, 19}, {0}}, {{17, 21}, {3}}, {{17, 30}, {0}}, {{21, 13}, {3}},
  {{21, 4}, {9}}, {{21, 19}, {0}}, {{21, 21}, {8}}, {{21, 30}, {1}}, {{11, 13}, {10}},
  {{11, 4}, {11}}, {{11, 19}, {3}}, {{11, 21}, {3}}, {{11, 30}, {2}}, {{28, 13}, {9}},
  {{28, 4}, {2}}, {{28, 19}, {7}}, {{28, 21}, {3}}, {{28, 30}, {1}}, {{13, 25}, {6}},
  {{13, 1}, {12}}, {{13, 24}, {5}}, {{13, 25}, {12}}, {{4, 25}, {10}},
  {{4, 1}, {6}}, {{4, 24}, {7}}, {{4, 25}, {3}}, {{19, 25}, {2}}, {{19, 1}, {6}},
  {{19, 24}, {0}}, {{19, 25}, {0}}, {{21, 25}, {4}}, {{21, 1}, {13}}, {{21, 24}, {8}},
  {{21, 25}, {0}}, {{30, 25}, {0}}, {{30, 1}, {7}}, {{30, 24}, {1}}, {{30, 25}, {2}},
  {{15, 13}, {7}}, {{15, 4}, {11}}, {{15, 19}, {1}}, {{15, 21}, {7}}, {{15, 30}, {4}},
  {{1, 13}, {16}}, {{1, 4}, {3}}, {{1, 19}, {4}}, {{1, 21}, {5}}, {{1, 30}, {7}},
  {{23, 13}, {5}}, {{23, 4}, {13}}, {{23, 19}, {1}}, {{23, 21}, {3}}, {{23, 30}, {1}},
  {{12, 13}, {13}}, {{12, 4}, {1}}, {{12, 19}, {3}}, {{12, 21}, {5}}, {{12, 30}, {0}}}
```

```
In[•]:= genspeDatSimple5 = Partition[Flatten[genspeDat5], {3}];
```

```
In[•]:= Length[genspeDatSimple5]
Out[•]= 108
```

Niche Expansion and Contraction Nonlinear model Fit

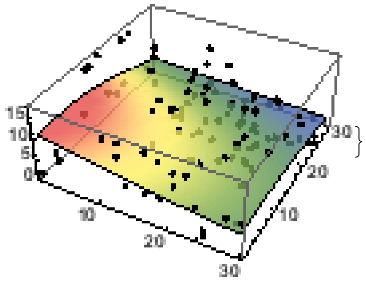
```
(*Niche Expansion*)
```

```
In[•]:= genspeDat25 = Partition[Flatten[genspeDat5], {3}]

Out[•]= {{25, 15, 1}, {25, 1, 1}, {25, 23, 1}, {25, 12, 12}, {1, 15, 17}, {1, 1, 1},
         {1, 23, 16}, {1, 12, 16}, {24, 15, 2}, {24, 1, 7}, {24, 23, 0}, {24, 12, 10},
         {25, 15, 15}, {25, 1, 2}, {25, 23, 10}, {25, 12, 0}, {15, 17, 2}, {15, 21, 2},
         {15, 11, 14}, {15, 28, 10}, {1, 17, 6}, {1, 21, 16}, {1, 11, 6}, {1, 28, 3},
         {23, 17, 0}, {23, 21, 0}, {23, 11, 7}, {23, 28, 5}, {12, 17, 7}, {12, 21, 8},
         {12, 11, 4}, {12, 28, 13}, {17, 25, 3}, {17, 1, 6}, {17, 24, 2}, {17, 25, 4},
         {21, 25, 0}, {21, 1, 10}, {21, 24, 0}, {21, 25, 4}, {11, 25, 9}, {11, 1, 15},
         {11, 24, 8}, {11, 25, 3}, {28, 25, 4}, {28, 1, 7}, {28, 24, 7}, {28, 25, 0},
         {17, 13, 1}, {17, 4, 10}, {17, 19, 0}, {17, 21, 3}, {17, 30, 0}, {21, 13, 3},
         {21, 4, 9}, {21, 19, 0}, {21, 21, 8}, {21, 30, 1}, {11, 13, 10}, {11, 4, 11},
         {11, 19, 3}, {11, 21, 3}, {11, 30, 2}, {28, 13, 9}, {28, 4, 2}, {28, 19, 7},
         {28, 21, 3}, {28, 30, 1}, {13, 25, 6}, {13, 1, 12}, {13, 24, 5}, {13, 25, 12},
         {4, 25, 10}, {4, 1, 6}, {4, 24, 7}, {4, 25, 3}, {19, 25, 2}, {19, 1, 6},
         {19, 24, 0}, {19, 25, 0}, {21, 25, 4}, {21, 1, 13}, {21, 24, 8}, {21, 25, 0},
         {30, 25, 0}, {30, 1, 7}, {30, 24, 1}, {30, 25, 2}, {15, 13, 7}, {15, 4, 11},
         {15, 19, 1}, {15, 21, 7}, {15, 30, 4}, {1, 13, 16}, {1, 4, 3}, {1, 19, 4},
         {1, 21, 5}, {1, 30, 7}, {23, 13, 5}, {23, 4, 13}, {23, 19, 1}, {23, 21, 3},
         {23, 30, 1}, {12, 13, 13}, {12, 4, 1}, {12, 19, 3}, {12, 21, 5}, {12, 30, 0}}
```

```
In[•]:= nlmExpansion5 = NonlinearModelFit[genspeDat25, A Exp[-((x - xm) ^2/(2 vx ^2) + (y - ym) ^2/(2 vy ^2))],
          {A, xm, ym, vx, vy}, {x, y}, MaxIterations → ∞(*,WorkingPrecision→5*)]

Out[•]= FittedModel[ 22.7068 e^(-0.00017715 (63.1307+x)^2-0.00166552 (-≪17≫+y)^2) ]
```

```
In[•]:= {fitExpansion5 = nlmExpansion5["BestFit"],
         Show[Plot3D[fitExpansion5, {x, 1, 30}, {y, 1, 30}, ColorFunction → "DarkRainbow",
           PlotStyle → Directive[Opacity[0.8], Red], Mesh → None, PlotRange → All],
          Graphics3D[{PointSize[0.025], Point[genspeDat25]}]]}
```

Out[•]= {22.7068 $e^{-0.00017715\ (63.1307+x)^2-0.00166552\ (-6.63495+y)^2}$,



}

```
In[•]:= nlmExpansion5["ANOVATable"]

                    | DF   SS       MS
        ------------+-----------------------
        Model       | 5    3762.15  752.43
Out[•]= Error       | 103  1893.85  18.3869
        Uncorrected Total | 108  5656.
        Corrected Total   | 107  2366.96
```

```
In[•]:= anovaE = nlmExpansion5["ANOVATableEntries"]
Out[•]= {{5, 3762.15, 752.43}, {103, 1893.85, 18.3869}, {108, 5656.}, {107, 2366.96}}
```

```
In[•]:= fRatioE = anovaE[[1, 3]] / anovaE[[2, 3]]
Out[•]= 40.9221
```

```
In[•]:= pValueE = 1 - CDF[FRatioDistribution[anovaE[[1, 1]], anovaE[[2, 1]]], fRatioE]
Out[•]= 0.
```

```
(*Niche Contraction*)
```

```
In[•]:= genContrDat5 = {{#[[1]], #[[2]]}, {#[[6]]}} & /@ gk5
Out[•]= {{{25, 15}, {1}}, {{25, 1}, {1}}, {{25, 23}, {3}}, {{25, 12}, {0}}, {{1, 15}, {0}},
  {{1, 1}, {0}}, {{1, 23}, {0}}, {{1, 12}, {0}}, {{24, 15}, {1}}, {{24, 1}, {0}},
  {{24, 23}, {5}}, {{24, 12}, {2}}, {{25, 15}, {2}}, {{25, 1}, {1}}, {{25, 23}, {3}},
  {{25, 12}, {7}}, {{15, 17}, {0}}, {{15, 21}, {1}}, {{15, 11}, {0}}, {{15, 28}, {0}},
  {{1, 17}, {1}}, {{1, 21}, {0}}, {{1, 11}, {0}}, {{1, 28}, {0}}, {{23, 17}, {2}},
  {{23, 21}, {4}}, {{23, 11}, {3}}, {{23, 28}, {5}}, {{12, 17}, {3}}, {{12, 21}, {4}},
  {{12, 11}, {5}}, {{12, 28}, {2}}, {{17, 25}, {4}}, {{17, 1}, {0}}, {{17, 24}, {3}},
  {{17, 25}, {12}}, {{21, 25}, {6}}, {{21, 1}, {0}}, {{21, 24}, {9}}, {{21, 25}, {10}},
  {{11, 25}, {2}}, {{11, 1}, {0}}, {{11, 24}, {1}}, {{11, 25}, {5}}, {{28, 25}, {4}},
  {{28, 1}, {1}}, {{28, 24}, {1}}, {{28, 25}, {16}}, {{17, 13}, {2}}, {{17, 4}, {4}},
  {{17, 19}, {4}}, {{17, 21}, {5}}, {{17, 30}, {12}}, {{21, 13}, {1}}, {{21, 4}, {2}},
  {{21, 19}, {5}}, {{21, 21}, {5}}, {{21, 30}, {9}}, {{11, 13}, {1}}, {{11, 4}, {1}},
  {{11, 19}, {0}}, {{11, 21}, {7}}, {{11, 30}, {7}}, {{28, 13}, {1}}, {{28, 4}, {1}},
  {{28, 19}, {3}}, {{28, 21}, {13}}, {{28, 30}, {13}}, {{13, 25}, {1}},
  {{13, 1}, {0}}, {{13, 24}, {2}}, {{13, 25}, {1}}, {{4, 25}, {0}}, {{4, 1}, {0}},
  {{4, 24}, {1}}, {{4, 25}, {4}}, {{19, 25}, {4}}, {{19, 1}, {0}}, {{19, 24}, {6}},
  {{19, 25}, {6}}, {{21, 25}, {5}}, {{21, 1}, {0}}, {{21, 24}, {6}}, {{21, 25}, {13}},
  {{30, 25}, {5}}, {{30, 1}, {0}}, {{30, 24}, {10}}, {{30, 25}, {17}},
  {{15, 13}, {0}}, {{15, 4}, {1}}, {{15, 19}, {1}}, {{15, 21}, {1}}, {{15, 30}, {1}},
  {{1, 13}, {0}}, {{1, 4}, {1}}, {{1, 19}, {0}}, {{1, 21}, {0}}, {{1, 30}, {0}},
  {{23, 13}, {0}}, {{23, 4}, {1}}, {{23, 19}, {3}}, {{23, 21}, {5}}, {{23, 30}, {9}},
  {{12, 13}, {0}}, {{12, 4}, {2}}, {{12, 19}, {3}}, {{12, 21}, {5}}, {{12, 30}, {7}}}
```

In[∘]:=
```
genContrDatSimple5 = Partition[Flatten[genContrDat5], {3}]
```

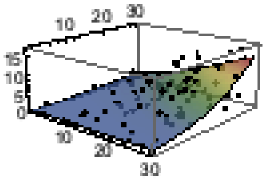
Out[∘]=
```
{{25, 15, 1}, {25, 1, 1}, {25, 23, 3}, {25, 12, 0}, {1, 15, 0}, {1, 1, 0}, {1, 23, 0},
 {1, 12, 0}, {24, 15, 1}, {24, 1, 0}, {24, 23, 5}, {24, 12, 2}, {25, 15, 2}, {25, 1, 1},
 {25, 23, 3}, {25, 12, 7}, {15, 17, 0}, {15, 21, 1}, {15, 11, 0}, {15, 28, 0},
 {1, 17, 1}, {1, 21, 0}, {1, 11, 0}, {1, 28, 0}, {23, 17, 2}, {23, 21, 4}, {23, 11, 3},
 {23, 28, 5}, {12, 17, 3}, {12, 21, 4}, {12, 11, 5}, {12, 28, 2}, {17, 25, 4},
 {17, 1, 0}, {17, 24, 3}, {17, 25, 12}, {21, 25, 6}, {21, 1, 0}, {21, 24, 9},
 {21, 25, 10}, {11, 25, 2}, {11, 1, 0}, {11, 24, 1}, {11, 25, 5}, {28, 25, 4},
 {28, 1, 1}, {28, 24, 1}, {28, 25, 16}, {17, 13, 2}, {17, 4, 4}, {17, 19, 4},
 {17, 21, 5}, {17, 30, 12}, {21, 13, 1}, {21, 4, 2}, {21, 19, 5}, {21, 21, 5},
 {21, 30, 9}, {11, 13, 1}, {11, 4, 1}, {11, 19, 0}, {11, 21, 7}, {11, 30, 7},
 {28, 13, 1}, {28, 4, 1}, {28, 19, 3}, {28, 21, 13}, {28, 30, 13}, {13, 25, 1},
 {13, 1, 0}, {13, 24, 2}, {13, 25, 1}, {4, 25, 0}, {4, 1, 0}, {4, 24, 1}, {4, 25, 4},
 {19, 25, 4}, {19, 1, 0}, {19, 24, 6}, {19, 25, 6}, {21, 25, 5}, {21, 1, 0},
 {21, 24, 6}, {21, 25, 13}, {30, 25, 5}, {30, 1, 0}, {30, 24, 10}, {30, 25, 17},
 {15, 13, 0}, {15, 4, 1}, {15, 19, 1}, {15, 21, 1}, {15, 30, 1}, {1, 13, 0}, {1, 4, 1},
 {1, 19, 0}, {1, 21, 0}, {1, 30, 0}, {23, 13, 0}, {23, 4, 1}, {23, 19, 3}, {23, 21, 5},
 {23, 30, 9}, {12, 13, 0}, {12, 4, 2}, {12, 19, 3}, {12, 21, 5}, {12, 30, 7}}
```

In[∘]:=
```
nlmContraction5 =
  NonlinearModelFit[genContrDatSimple5, A Exp[-((x - xm) ^2/(2 vx ^2) + (y - ym) ^2/(2 vy ^2))],
   {A, xm, ym, vx, vy}, {x, y}(*,MaxIterations→∞,WorkingPrecision→5*)]
```

Out[∘]= FittedModel[$32.1539\, e^{-0.00148502\,(-41.528+x)^2-0.00161335\,(-\ll 18 \gg+y)^2}$]

In[∘]:=
```
{fitContraction5 = nlmContraction5["BestFit"],
 Show[Plot3D[fitContraction5, {x, 1, 30}, {y, 1, 30}, ColorFunction → "DarkRainbow",
   PlotStyle → Directive[Opacity[0.8], Red], Mesh → None, PlotRange → All],
  Graphics3D[{PointSize[0.028], Point[genContrDatSimple5]}]]}
```

Out[∘]= { $32.1539\, e^{-0.00148502\,(-41.528+x)^2-0.00161335\,(-50.0887+y)^2}$,



}

In[∘]:=
```
nlmContraction5["ANOVATable"]
```

Out[∘]=
```
                  | DF   SS        MS
Model             | 5    1944.79   388.959
Error             | 103  712.205   6.91461
Uncorrected Total | 108  2657.
Corrected Total   | 107  1542.1
```

```
In[•]:= anovaC = nlmContraction5["ANOVATableEntries"]

Out[•]= {{5, 1944.79, 388.959}, {103, 712.205, 6.91461}, {108, 2657.}, {107, 1542.1}}

In[•]:= fRatioC = anovaC[[1, 3]] / anovaC[[2, 3]]

Out[•]= 56.2517

In[•]:= pValueC = 1 - CDF[FRatioDistribution[anovaC[[1, 1]], anovaC[[2, 1]]], fRatioC]

Out[•]= 0.
```

Niche Expansion and contraction vs. phylogenetic distance between partners

Phylogeny

```
In[•]:= ki5 = StringReplace[#, {"-" -> " ", " " → ""}] & /@ comType[[40 ;; 147]]

Out[•]= {ABR ABH, ABR BSH, ABR ECH, ABR SOH, BSR ABH, BSR BSH, BSR ECH, BSR SOH, ECR ABH,
         ECR BSH, ECR ECH, ECR SOH, SOR ABH, SOR BSH, SOR ECH, SOR SOH, ABH ABW, ABH ECW,
         ABH SOW, ABH PFW, BSH ABW, BSH ECW, BSH SOW, BSH PFW, ECH ABW, ECH ECW, ECH SOW,
         ECH PFW, SOH ABW, SOH ECW, SOH SOW, SOH PFW, ABW ABR, ABW BSR, ABW ECR, ABW SOR,
         ECW ABR, ECW BSR, ECW ECR, ECW SOR, SOW ABR, SOW BSR, SOW ECR, SOW SOR,
         PFW ABR, PFW BSR, PFW ECR, PFW SOR, ABW ABL, ABW BSL, ABW ECL, ABW SOL,
         ABW PFL, ECW ABL, ECW BSL, ECW ECL, ECW SOL, ECW PFL, SOW ABL, SOW BSL,
         SOW ECL, SOW SOL, SOW PFL, PFW ABL, PFW BSL, PFW ECL, PFW SOL, PFW PFL,
         ABL ABR, ABL BSR, ABL ECR, ABL SOR, BSL ABR, BSL BSR, BSL ECR, BSL SOR,
         ECL ABR, ECL BSR, ECL ECR, ECL SOR, SOL ABR, SOL BSR, SOL ECR, SOL SOR,
         PFL ABR, PFL BSR, PFL ECR, PFL SOR, ABH ABL, ABH BSL, ABH ECL, ABH SOL,
         ABH PFL, BSH ABL, BSH BS L, BSH ECL, BSH SOL, BSH PFL, ECH ABL, ECH BSL,
         ECH ECL, ECH SOL, ECH PFL, SOH ABL, SOH BSL, SOH ECL, SOH SOL, SOH PFL}
```

```
In[•]:= pai5 = StringSplit /@ ki5

Out[•]= {{ABR, ABH}, {ABR, BSH}, {ABR, ECH}, {ABR, SOH}, {BSR, ABH}, {BSR, BSH}, {BSR, ECH},
         {BSR, SOH}, {ECR, ABH}, {ECR, BSH}, {ECR, ECH}, {ECR, SOH}, {SOR, ABH}, {SOR, BSH},
         {SOR, ECH}, {SOR, SOH}, {ABH, ABW}, {ABH, ECW}, {ABH, SOW}, {ABH, PFW}, {BSH, ABW},
         {BSH, ECW}, {BSH, SOW}, {BSH, PFW}, {ECH, ABW}, {ECH, ECW}, {ECH, SOW}, {ECH, PFW},
         {SOH, ABW}, {SOH, ECW}, {SOH, SOW}, {SOH, PFW}, {ABW, ABR}, {ABW, BSR}, {ABW, ECR},
         {ABW, SOR}, {ECW, ABR}, {ECW, BSR}, {ECW, ECR}, {ECW, SOR}, {SOW, ABR}, {SOW, BSR},
         {SOW, ECR}, {SOW, SOR}, {PFW, ABR}, {PFW, BSR}, {PFW, ECR}, {PFW, SOR}, {ABW, ABL},
         {ABW, BSL}, {ABW, ECL}, {ABW, SOL}, {ABW, PFL}, {ECW, ABL}, {ECW, BSL}, {ECW, ECL},
         {ECW, SOL}, {ECW, PFL}, {SOW, ABL}, {SOW, BSL}, {SOW, ECL}, {SOW, SOL}, {SOW, PFL},
         {PFW, ABL}, {PFW, BSL}, {PFW, ECL}, {PFW, SOL}, {PFW, PFL}, {ABL, ABR}, {ABL, BSR},
         {ABL, ECR}, {ABL, SOR}, {BSL, ABR}, {BSL, BSR}, {BSL, ECR}, {BSL, SOR}, {ECL, ABR},
         {ECL, BSR}, {ECL, ECR}, {ECL, SOR}, {SOL, ABR}, {SOL, BSR}, {SOL, ECR}, {SOL, SOR},
         {PFL, ABR}, {PFL, BSR}, {PFL, ECR}, {PFL, SOR}, {ABH, ABL}, {ABH, BSL},
         {ABH, ECL}, {ABH, SOL}, {ABH, PFL}, {BSH, ABL}, {BSH, BS, L}, {BSH, ECL},
         {BSH, SOL}, {BSH, PFL}, {ECH, ABL}, {ECH, BSL}, {ECH, ECL}, {ECH, SOL},
         {ECH, PFL}, {SOH, ABL}, {SOH, BSL}, {SOH, ECL}, {SOH, SOL}, {SOH, PFL}}

In[•]:= qz5 = MapAt[(# <> "5" &), pai5, {All, All}]

Out[•]= {{ABR5, ABH5}, {ABR5, BSH5}, {ABR5, ECH5}, {ABR5, SOH5}, {BSR5, ABH5}, {BSR5, BSH5},
         {BSR5, ECH5}, {BSR5, SOH5}, {ECR5, ABH5}, {ECR5, BSH5}, {ECR5, ECH5}, {ECR5, SOH5},
         {SOR5, ABH5}, {SOR5, BSH5}, {SOR5, ECH5}, {SOR5, SOH5}, {ABH5, ABW5}, {ABH5, ECW5},
         {ABH5, SOW5}, {ABH5, PFW5}, {BSH5, ABW5}, {BSH5, ECW5}, {BSH5, SOW5}, {BSH5, PFW5},
         {ECH5, ABW5}, {ECH5, ECW5}, {ECH5, SOW5}, {ECH5, PFW5}, {SOH5, ABW5}, {SOH5, ECW5},
         {SOH5, SOW5}, {SOH5, PFW5}, {ABW5, ABR5}, {ABW5, BSR5}, {ABW5, ECR5}, {ABW5, SOR5},
         {ECW5, ABR5}, {ECW5, BSR5}, {ECW5, ECR5}, {ECW5, SOR5}, {SOW5, ABR5}, {SOW5, BSR5},
         {SOW5, ECR5}, {SOW5, SOR5}, {PFW5, ABR5}, {PFW5, BSR5}, {PFW5, ECR5}, {PFW5, SOR5},
         {ABW5, ABL5}, {ABW5, BSL5}, {ABW5, ECL5}, {ABW5, SOL5}, {ABW5, PFL5}, {ECW5, ABL5},
         {ECW5, BSL5}, {ECW5, ECL5}, {ECW5, SOL5}, {ECW5, PFL5}, {SOW5, ABL5}, {SOW5, BSL5},
         {SOW5, ECL5}, {SOW5, SOL5}, {SOW5, PFL5}, {PFW5, ABL5}, {PFW5, BSL5}, {PFW5, ECL5},
         {PFW5, SOL5}, {PFW5, PFL5}, {ABL5, ABR5}, {ABL5, BSR5}, {ABL5, ECR5}, {ABL5, SOR5},
         {BSL5, ABR5}, {BSL5, BSR5}, {BSL5, ECR5}, {BSL5, SOR5}, {ECL5, ABR5}, {ECL5, BSR5},
         {ECL5, ECR5}, {ECL5, SOR5}, {SOL5, ABR5}, {SOL5, BSR5}, {SOL5, ECR5}, {SOL5, SOR5},
         {PFL5, ABR5}, {PFL5, BSR5}, {PFL5, ECR5}, {PFL5, SOR5}, {ABH5, ABL5}, {ABH5, BSL5},
         {ABH5, ECL5}, {ABH5, SOL5}, {ABH5, PFL5}, {BSH5, ABL5}, {BSH5, BS5, L5}, {BSH5, ECL5},
         {BSH5, SOL5}, {BSH5, PFL5}, {ECH5, ABL5}, {ECH5, BSL5}, {ECH5, ECL5}, {ECH5, SOL5},
         {ECH5, PFL5}, {SOH5, ABL5}, {SOH5, BSL5}, {SOH5, ECL5}, {SOH5, SOL5}, {SOH5, PFL5}}
```

```
In[•]:= qzK5 = {{"ABR5", "ABH5"}, {"ABR5", "BSH5"}, {"ABR5", "ECH5"}, {"ABR5", "SOH5"},
           {"BSR5", "ABH5"}, {"BSR5", "BSH5"}, {"BSR5", "ECH5"}, {"BSR5", "SOH5"},
           {"ECR5", "ABH5"}, {"ECR5", "BSH5"}, {"ECR5", "ECH5"}, {"ECR5", "SOH5"},
           {"SOR5", "ABH5"}, {"SOR5", "BSH5"}, {"SOR5", "ECH5"}, {"SOR5", "SOH5"},
           {"ABH5", "ABW5"}, {"ABH5", "ECW5"}, {"ABH5", "SOW5"}, {"ABH5", "PFW5"},
           {"BSH5", "ABW5"}, {"BSH5", "ECW5"}, {"BSH5", "SOW5"}, {"BSH5", "PFW5"},
           {"ECH5", "ABW5"}, {"ECH5", "ECW5"}, {"ECH5", "SOW5"}, {"ECH5", "PFW5"},
           {"SOH5", "ABW5"}, {"SOH5", "ECW5"}, {"SOH5", "SOW5"}, {"SOH5", "PFW5"},
           {"ABW5", "ABR5"}, {"ABW5", "BSR5"}, {"ABW5", "ECR5"}, {"ABW5", "SOR5"},
           {"ECW5", "ABR5"}, {"ECW5", "BSR5"}, {"ECW5", "ECR5"}, {"ECW5", "SOR5"},
           {"SOW5", "ABR5"}, {"SOW5", "BSR5"}, {"SOW5", "ECR5"}, {"SOW5", "SOR5"},
           {"PFW5", "ABR5"}, {"PFW5", "BSR5"}, {"PFW5", "ECR5"}, {"PFW5", "SOR5"},
           {"ABW5", "ABL5"}, {"ABW5", "BSL5"}, {"ABW5", "ECL5"}, {"ABW5", "SOL5"},
           {"ABW5", "PFL5"}, {"ECW5", "ABL5"}, {"ECW5", "BSL5"}, {"ECW5", "ECL5"},
           {"ECW5", "SOL5"}, {"ECW5", "PFL5"}, {"SOW5", "ABL5"}, {"SOW5", "BSL5"},
           {"SOW5", "ECL5"}, {"SOW5", "SOL5"}, {"SOW5", "PFL5"}, {"PFW5", "ABL5"},
           {"PFW5", "BSL5"}, {"PFW5", "ECL5"}, {"PFW5", "SOL5"}, {"PFW5", "PFL5"},
           {"ABL5", "ABR5"}, {"ABL5", "BSR5"}, {"ABL5", "ECR5"}, {"ABL5", "SOR5"},
           {"BSL5", "ABR5"}, {"BSL5", "BSR5"}, {"BSL5", "ECR5"}, {"BSL5", "SOR5"},
           {"ECL5", "ABR5"}, {"ECL5", "BSR5"}, {"ECL5", "ECR5"}, {"ECL5", "SOR5"},
           {"SOL5", "ABR5"}, {"SOL5", "BSR5"}, {"SOL5", "ECR5"}, {"SOL5", "SOR5"},
           {"PFL5", "ABR5"}, {"PFL5", "BSR5"}, {"PFL5", "ECR5"}, {"PFL5", "SOR5"},
           {"ABH5", "ABL5"}, {"ABH5", "BSL5"}, {"ABH5", "ECL5"}, {"ABH5", "SOL5"},
           {"ABH5", "PFL5"}, {"BSH5", "ABL5"}, {"BSH5", "BSL5"}, {"BSH5", "ECL5"},
           {"BSH5", "SOL5"}, {"BSH5", "PFL5"}, {"ECH5", "ABL5"}, {"ECH5", "BSL5"},
           {"ECH5", "ECL5"}, {"ECH5", "SOL5"}, {"ECH5", "PFL5"}, {"SOH5", "ABL5"},
           {"SOH5", "BSL5"}, {"SOH5", "ECL5"}, {"SOH5", "SOL5"}, {"SOH5", "PFL5"}};

In[•]:= mondef5 = MapAt[ToExpression, qzK5, {All, All}];
```

```
In[•]:= jop5 = Table[Join[qzK5[[i]], {poStingHour5[[i]]}], {i, 1, Length[poStingHour5]}]

Out[•]= {{ABR5, ABH5, ABRABH5}, {ABR5, BSH5, ABRBSH5}, {ABR5, ECH5, ABRECH5},
         {ABR5, SOH5, ABRSOH5}, {BSR5, ABH5, BSRABH5}, {BSR5, BSH5, BSRBSH5},
         {BSR5, ECH5, BSRECH5}, {BSR5, SOH5, BSRSOH5}, {ECR5, ABH5, ECRABH5},
         {ECR5, BSH5, ECRBSH5}, {ECR5, ECH5, ECRECH5}, {ECR5, SOH5, ECRSOH5},
         {SOR5, ABH5, SORABH5}, {SOR5, BSH5, SORBSH5}, {SOR5, ECH5, SORECH5},
         {SOR5, SOH5, SORSOH5}, {ABH5, ABW5, ABHABW5}, {ABH5, ECW5, ABHECW5},
         {ABH5, SOW5, ABHSOW5}, {ABH5, PFW5, ABHPFW5}, {BSH5, ABW5, BSHABW5},
         {BSH5, ECW5, BSHECW5}, {BSH5, SOW5, BSHSOW5}, {BSH5, PFW5, BSHPFW5},
         {ECH5, ABW5, ECHABW5}, {ECH5, ECW5, ECHECW5}, {ECH5, SOW5, ECHSOW5},
         {ECH5, PFW5, ECHPFW5}, {SOH5, ABW5, SOHABW5}, {SOH5, ECW5, SOHECW5},
         {SOH5, SOW5, SOHSOW5}, {SOH5, PFW5, SOHPFW5}, {ABW5, ABR5, ABWABR5},
         {ABW5, BSR5, ABWBSR5}, {ABW5, ECR5, ABWECR5}, {ABW5, SOR5, ABWSOR5},
         {ECW5, ABR5, ECWABR5}, {ECW5, BSR5, ECWBSR5}, {ECW5, ECR5, ECWECR5},
         {ECW5, SOR5, ECWSOR5}, {SOW5, ABR5, SOWABR5}, {SOW5, BSR5, SOWBSR5},
         {SOW5, ECR5, SOWECR5}, {SOW5, SOR5, SOWSOR5}, {PFW5, ABR5, PFWABR5},
         {PFW5, BSR5, PFWBSR5}, {PFW5, ECR5, PFWECR5}, {PFW5, SOR5, PFWSOR5},
         {ABW5, ABL5, ABWABL5}, {ABW5, BSL5, ABWBSL5}, {ABW5, ECL5, ABWECL5},
         {ABW5, SOL5, ABWSOL5}, {ABW5, PFL5, ABWPFL5}, {ECW5, ABL5, ECWABL5},
         {ECW5, BSL5, ECWBSL5}, {ECW5, ECL5, ECWECL5}, {ECW5, SOL5, ECWSOL5},
         {ECW5, PFL5, ECWPFL5}, {SOW5, ABL5, SOWABL5}, {SOW5, BSL5, SOWBSL5},
         {SOW5, ECL5, SOWECL5}, {SOW5, SOL5, SOWSOL5}, {SOW5, PFL5, SOWPFL5},
         {PFW5, ABL5, PFWABL5}, {PFW5, BSL5, PFWBSL5}, {PFW5, ECL5, PFWECL5},
         {PFW5, SOL5, PFWSOL5}, {PFW5, PFL5, PFWPFL5}, {ABL5, ABR5, ABLABR5},
         {ABL5, BSR5, ABLBSR5}, {ABL5, ECR5, ABLECR5}, {ABL5, SOR5, ABLSOR5},
         {BSL5, ABR5, BSLABR5}, {BSL5, BSR5, BSLBSR5}, {BSL5, ECR5, BSLECR5},
         {BSL5, SOR5, BSLSOR5}, {ECL5, ABR5, ECLABR5}, {ECL5, BSR5, ECLBSR5},
         {ECL5, ECR5, ECLECR5}, {ECL5, SOR5, ECLSOR5}, {SOL5, ABR5, SOLABR5},
         {SOL5, BSR5, SOLBSR5}, {SOL5, ECR5, SOLECR5}, {SOL5, SOR5, SOLSOR5},
         {PFL5, ABR5, PFLABR5}, {PFL5, BSR5, PFLBSR5}, {PFL5, ECR5, PFLECR5},
         {PFL5, SOR5, PFLSOR5}, {ABH5, ABL5, ABHABL5}, {ABH5, BSL5, ABHBSL5},
         {ABH5, ECL5, ABHECL5}, {ABH5, SOL5, ABHSOL5}, {ABH5, PFL5, ABHPFL5},
         {BSH5, ABL5, BSHABL5}, {BSH5, BSL5, BSHBSL5}, {BSH5, ECL5, BSHECL5},
         {BSH5, SOL5, BSHSOL5}, {BSH5, PFL5, BSHPFL5}, {ECH5, ABL5, ECHABL5},
         {ECH5, BSL5, ECHBSL5}, {ECH5, ECL5, ECHECL5}, {ECH5, SOL5, ECHSOL5},
         {ECH5, PFL5, ECHPFL5}, {SOH5, ABL5, SOHABL5}, {SOH5, BSL5, SOHBSL5},
         {SOH5, ECL5, SOHECL5}, {SOH5, SOL5, SOHSOL5}, {SOH5, PFL5, SOHPFL5}}
```

```
In[•]:= gjk5 = Drop[#, -1] & /@ jop5

Out[•]= {{ABR5, ABH5}, {ABR5, BSH5}, {ABR5, ECH5}, {ABR5, SOH5}, {BSR5, ABH5}, {BSR5, BSH5},
         {BSR5, ECH5}, {BSR5, SOH5}, {ECR5, ABH5}, {ECR5, BSH5}, {ECR5, ECH5}, {ECR5, SOH5},
         {SOR5, ABH5}, {SOR5, BSH5}, {SOR5, ECH5}, {SOR5, SOH5}, {ABH5, ABW5}, {ABH5, ECW5},
         {ABH5, SOW5}, {ABH5, PFW5}, {BSH5, ABW5}, {BSH5, ECW5}, {BSH5, SOW5}, {BSH5, PFW5},
         {ECH5, ABW5}, {ECH5, ECW5}, {ECH5, SOW5}, {ECH5, PFW5}, {SOH5, ABW5}, {SOH5, ECW5},
         {SOH5, SOW5}, {SOH5, PFW5}, {ABW5, ABR5}, {ABW5, BSR5}, {ABW5, ECR5}, {ABW5, SOR5},
         {ECW5, ABR5}, {ECW5, BSR5}, {ECW5, ECR5}, {ECW5, SOR5}, {SOW5, ABR5}, {SOW5, BSR5},
         {SOW5, ECR5}, {SOW5, SOR5}, {PFW5, ABR5}, {PFW5, BSR5}, {PFW5, ECR5}, {PFW5, SOR5},
         {ABW5, ABL5}, {ABW5, BSL5}, {ABW5, ECL5}, {ABW5, SOL5}, {ABW5, PFL5}, {ECW5, ABL5},
         {ECW5, BSL5}, {ECW5, ECL5}, {ECW5, SOL5}, {ECW5, PFL5}, {SOW5, ABL5}, {SOW5, BSL5},
         {SOW5, ECL5}, {SOW5, SOL5}, {SOW5, PFL5}, {PFW5, ABL5}, {PFW5, BSL5}, {PFW5, ECL5},
         {PFW5, SOL5}, {PFW5, PFL5}, {ABL5, ABR5}, {ABL5, BSR5}, {ABL5, ECR5}, {ABL5, SOR5},
         {BSL5, ABR5}, {BSL5, BSR5}, {BSL5, ECR5}, {BSL5, SOR5}, {ECL5, ABR5}, {ECL5, BSR5},
         {ECL5, ECR5}, {ECL5, SOR5}, {SOL5, ABR5}, {SOL5, BSR5}, {SOL5, ECR5}, {SOL5, SOR5},
         {PFL5, ABR5}, {PFL5, BSR5}, {PFL5, ECR5}, {PFL5, SOR5}, {ABH5, ABL5}, {ABH5, BSL5},
         {ABH5, ECL5}, {ABH5, SOL5}, {ABH5, PFL5}, {BSH5, ABL5}, {BSH5, BSL5}, {BSH5, ECL5},
         {BSH5, SOL5}, {BSH5, PFL5}, {ECH5, ABL5}, {ECH5, BSL5}, {ECH5, ECL5}, {ECH5, SOL5},
         {ECH5, PFL5}, {SOH5, ABL5}, {SOH5, BSL5}, {SOH5, ECL5}, {SOH5, SOL5}, {SOH5, PFL5}}
```

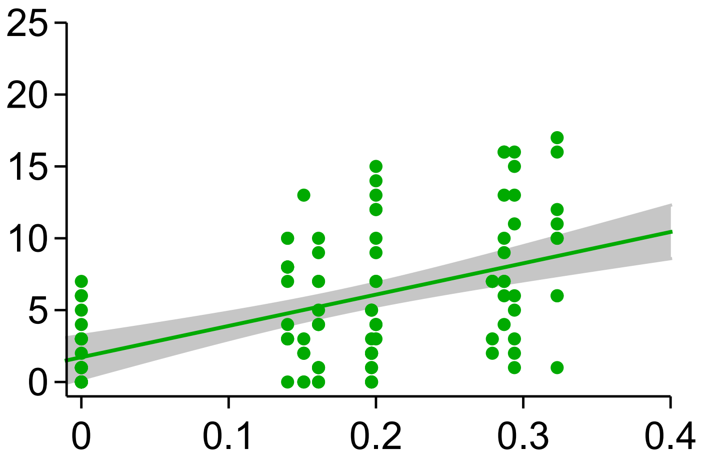
```
In[•]:= stk5 = StringTake[#, 1] &;
```

```
In[•]:= hio5 = {{"A", "A"}, {"A", "B"}, {"A", "E"}, {"A", "S"}, {"B", "A"}, {"B", "B"},
           {"B", "E"}, {"B", "S"}, {"E", "A"}, {"E", "B"}, {"E", "E"}, {"E", "S"},
           {"S", "A"}, {"S", "B"}, {"S", "E"}, {"S", "S"}, {"A", "A"}, {"A", "E"},
           {"A", "S"}, {"A", "P"}, {"B", "A"}, {"B", "E"}, {"B", "S"}, {"B", "P"},
           {"E", "A"}, {"E", "E"}, {"E", "S"}, {"E", "P"}, {"S", "A"}, {"S", "E"},
           {"S", "S"}, {"S", "P"}, {"A", "A"}, {"A", "B"}, {"A", "E"}, {"A", "S"},
           {"E", "A"}, {"E", "B"}, {"E", "E"}, {"E", "S"}, {"S", "A"}, {"S", "B"},
           {"S", "E"}, {"S", "S"}, {"P", "A"}, {"P", "B"}, {"P", "E"}, {"P", "S"},
           {"A", "A"}, {"A", "B"}, {"A", "E"}, {"A", "S"}, {"A", "P"}, {"E", "A"},
           {"E", "B"}, {"E", "E"}, {"E", "S"}, {"E", "P"}, {"S", "A"}, {"S", "B"},
           {"S", "E"}, {"S", "S"}, {"S", "P"}, {"P", "A"}, {"P", "B"}, {"P", "E"},
           {"P", "S"}, {"P", "P"}, {"A", "A"}, {"A", "B"}, {"A", "E"}, {"A", "S"},
           {"B", "A"}, {"B", "B"}, {"B", "E"}, {"B", "S"}, {"E", "A"}, {"E", "B"},
           {"E", "E"}, {"E", "S"}, {"S", "A"}, {"S", "B"}, {"S", "E"}, {"S", "S"},
           {"P", "A"}, {"P", "B"}, {"P", "E"}, {"P", "S"}, {"A", "A"}, {"A", "B"},
           {"A", "E"}, {"A", "S"}, {"A", "P"}, {"B", "A"}, {"B", "B"}, {"B", "E"},
           {"B", "S"}, {"B", "P"}, {"E", "A"}, {"E", "B"}, {"E", "E"}, {"E", "S"},
           {"E", "P"}, {"S", "A"}, {"S", "B"}, {"S", "E"}, {"S", "S"}, {"S", "P"}};
```

```
In[∘]:= PhyDiCorr5 =
          hio5 /. {{"A", "A"} → 0, {"A", "B"} → 0.323, {"A", "E"} → 0.197, {"A", "S"} → 0.2,
            {"B", "A"} → 0.323, {"B", "B"} → 0, {"B", "E"} → 0.287, {"B", "S"} → 0.294,
            {"E", "A"} → 0.197, {"E", "B"} → 0.287, {"E", "E"} → 0, {"E", "S"} → 0.14,
            {"S", "A"} → 0.2, {"S", "B"} → 0.294, {"S", "E"} → 0.14, {"S", "S"} → 0,
            {"A", "A"} → 0, {"A", "E"} → 0.197, {"A", "S"} → 0.2, {"A", "P"} → 0.161,
            {"B", "A"} → 0.323, {"B", "E"} → 0.287, {"B", "S"} → 0.294, {"B", "P"} → 0.279,
            {"E", "A"} → 0.197, {"E", "E"} → 0, {"E", "S"} → 0.14, {"E", "P"} → 0.161,
            {"S", "A"} → 0.2, {"S", "E"} → 0.14, {"S", "S"} → 0, {"S", "P"} → 0.151,
            {"A", "A"} → 0, {"A", "B"} → 0.323, {"A", "E"} → 0.197, {"A", "S"} → 0.2,
            {"E", "A"} → 0.197, {"E", "B"} → 0.287, {"E", "E"} → 0, {"E", "S"} → 0.14,
            {"S", "A"} → 0.2, {"S", "B"} → 0.294, {"S", "E"} → 0.14, {"S", "S"} → 0,
            {"P", "A"} → 0.161, {"P", "B"} → 0.279, {"P", "E"} → 0.161, {"P", "S"} → 0.151,
            {"A", "A"} → 0, {"A", "B"} → 0.323, {"A", "E"} → 0.197, {"A", "S"} → 0.2,
            {"A", "P"} → 0.161, {"E", "A"} → 0.197, {"E", "B"} → 0.287, {"E", "E"} → 0,
            {"E", "S"} → 0.14, {"E", "P"} → 0.161, {"S", "A"} → 0.2, {"S", "B"} → 0.294,
            {"S", "E"} → 0.14, {"S", "S"} → 0, {"S", "P"} → 0.151, {"P", "A"} → 0.161,
            {"P", "B"} → 0.279, {"P", "E"} → 0.161, {"P", "S"} → 0.151, {"P", "P"} → 0,
            {"A", "A"} → 0, {"A", "B"} → 0.323, {"A", "E"} → 0.197, {"A", "S"} → 0.2,
            {"B", "A"} → 0.323, {"B", "B"} → 0, {"B", "E"} → 0.287, {"B", "S"} → 0.294,
            {"E", "A"} → 0.197, {"E", "B"} → 0.287, {"E", "E"} → 0, {"E", "S"} → 0.14,
            {"S", "A"} → 0.2, {"S", "B"} → 0.294, {"S", "E"} → 0.14, {"S", "S"} → 0,
            {"P", "A"} → 0.161, {"P", "B"} → 0.279, {"P", "E"} → 0.161, {"P", "S"} → 0.151,
            {"A", "A"} → 0, {"A", "B"} → 0.323, {"A", "E"} → 0.197, {"A", "S"} → 0.2,
            {"A", "P"} → 0.161, {"B", "A"} → 0.323, {"B", "B"} → 0, {"B", "E"} → 0.287,
            {"B", "S"} → 0.294, {"B", "P"} → 0.279, {"E", "A"} → 0.197, {"E", "B"} → 0.287,
            {"E", "E"} → 0, {"E", "S"} → 0.14, {"E", "P"} → 0.161, {"S", "A"} → 0.2,
            {"S", "B"} → 0.294, {"S", "E"} → 0.14, {"S", "S"} → 0, {"S", "P"} → 0.151};

In[∘]:= expPo5 = #[[7]] & /@ gk5

Out[∘]= {1, 1, 1, 12, 17, 1, 16, 16, 2, 7, 0, 10, 15, 2, 10, 0, 2, 2, 14, 10, 6, 16, 6, 3, 0, 0, 7,
         5, 7, 8, 4, 13, 3, 6, 2, 4, 0, 10, 0, 4, 9, 15, 8, 3, 4, 7, 7, 0, 1, 10, 0, 3, 0, 3,
         9, 0, 8, 1, 10, 11, 3, 3, 2, 9, 2, 7, 3, 1, 6, 12, 5, 12, 10, 6, 7, 3, 2, 6, 0, 0, 4,
         13, 8, 0, 0, 7, 1, 2, 7, 11, 1, 7, 4, 16, 3, 4, 5, 7, 5, 13, 1, 3, 1, 13, 1, 3, 5, 0}

In[∘]:= contrPo5 = #[[6]] & /@ gk5

Out[∘]= {1, 1, 3, 0, 0, 0, 0, 0, 1, 0, 5, 2, 2, 1, 3, 7, 0, 1, 0, 0, 1, 0, 0, 0, 2, 4, 3, 5,
         3, 4, 5, 2, 4, 0, 3, 12, 6, 0, 9, 10, 2, 0, 1, 5, 4, 1, 1, 16, 2, 4, 4, 5, 12, 1,
         2, 5, 5, 9, 1, 1, 0, 7, 7, 1, 1, 3, 13, 13, 1, 0, 2, 1, 0, 0, 1, 4, 4, 0, 6, 6, 5,
         0, 6, 13, 5, 0, 10, 17, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 3, 5, 9, 0, 2, 3, 5, 7}

In[∘]:= PhyloExp5Days = Partition[Riffle[PhyDiCorr5, expPo5], {2}];
```

```
In[•]:= lmPhyloExp5Days = LinearModelFit[PhyloExp5Days, x, x]

Out[•]= FittedModel[ 1.72316 + 21.788 x ]

In[•]:= bands90PhyloExp5Days[x_] =
          lmPhyloExp5Days["MeanPredictionBands", ConfidenceLevel → .95];

In[•]:= PhyloExp1 = Plot[{lmPhyloExp5Days[x], bands90PhyloExp5Days[x]}, {x, -0.01, 0.4},
           PlotStyle → {Directive[Darker[Green], Thick], Lighter[Lighter[Gray]]},
           Filling → {2 → {{1}, Lighter[Lighter[Gray]]}},
           PlotRange → {{-0.01, 0.4}, {-1, 25}}, Ticks →
             fuk[{{0, 0.1, 0.2, 0.3, 0.4}, {0, 5, 10, 15, 20, 25}, "Arial", Plain, 20, 0.02}],
           TicksStyle → Thickness[0.004], AxesStyle → Thickness[0.004],
           AxesOrigin → {-0.01, -1}];

        grN = Graphics[{EdgeForm[{Darker[Green]}], FaceForm[Darker[Green]], Disk[]}];
        grG = Graphics[{EdgeForm[{Gray}], FaceForm[Darker[Green]], Disk[]}];
        grB = Graphics[{EdgeForm[{Black}], FaceForm[Darker[Green]], Disk[]}];

        PhyloExp2 = ListPlot[PhyloExp5Days, PlotMarkers → {grN, Scaled[0.035]}];
        Show[PhyloExp1, PhyloExp2]
```

Out[•]=



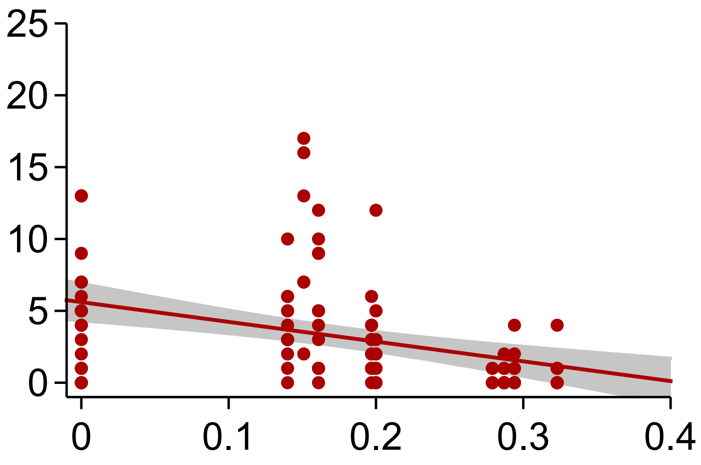
```
In[•]:= SpearmanRankTest[PhyDiCorr5, expPo5, "TestDataTable"]
```

Out[•]=

	Statistic	P-Value
Spearman Rank	0.49334	5.8047×10^{-8}

```
In[•]:= PhyloContr5Days = Partition[Riffle[PhyDiCorr5, contrPo5], {2}];
```

```
In[•]:= lmPhyloContr5Days = LinearModelFit[PhyloContr5Days, x, x]

Out[•]= FittedModel[ 5.60544 – 13.7345 x ]

In[•]:= bands90PhyloContr5Days[x_] =
          lmPhyloContr5Days["MeanPredictionBands", ConfidenceLevel → .95];

In[•]:= ConPhyl1 =
          Plot[{lmPhyloContr5Days[x], bands90PhyloContr5Days[x]}, {x, -0.01, 0.4},
           PlotStyle → {Directive[Darker[Red], Thick], Lighter[Lighter[Gray]]},
           Filling → {2 → {{1}, Lighter[Lighter[Gray]]}},
           PlotRange → {{-0.01, 0.4}, {-1, 25}}, Ticks →
            fuk[{{0, 0.1, 0.2, 0.3, 0.4}, {0, 5, 10, 15, 20, 25}, "Arial", Plain, 20, 0.02}],
           TicksStyle → Thickness[0.004], AxesStyle → Thickness[0.004],
           AxesOrigin → {-0.01, -1}];

        rdN = Graphics[{EdgeForm[{Darker[Red]}], FaceForm[Darker[Red]], Disk[]}];
        rdG = Graphics[{EdgeForm[{Gray}], FaceForm[Darker[Red]], Disk[]}];
        rdB = Graphics[{EdgeForm[{Black}], FaceForm[Darker[Red]], Disk[]}];

        ConPhyl2 = ListPlot[PhyloContr5Days, PlotMarkers → {rdN, Scaled[0.035]}];
        Show[ConPhyl1, ConPhyl2]

Out[•]=
```



```
In[•]:= SpearmanRankTest[PhyDiCorr5, contrPo5, "TestDataTable"]

Out[•]=               | Statistic   P-Value
        Spearman Rank | –0.513351   1.3381×10^-8
```

```
(*Phylogeny and Intersection between Niches -
 Niche Overlap*)
```

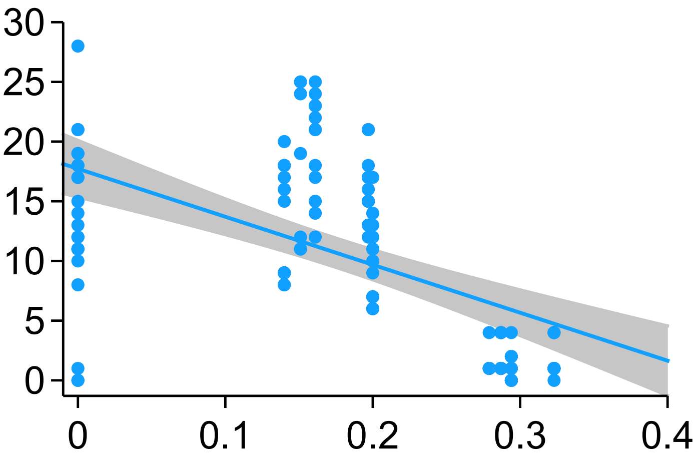
```
In[•]:= Intersect5 = #[[3]] & /@ gk5

Out[•]= {15, 1, 21, 10, 1, 0, 1, 0, 15, 1, 21, 9, 12, 1, 20, 12, 14, 15, 7, 14, 1, 1,
         0, 1, 17, 19, 9, 21, 7, 9, 8, 11, 17, 1, 17, 14, 21, 1, 21, 18, 9, 0, 9, 11,
         23, 1, 22, 24, 12, 4, 16, 13, 17, 12, 4, 17, 15, 21, 6, 2, 8, 11, 11, 12, 4,
         18, 19, 28, 13, 1, 13, 11, 4, 0, 4, 4, 18, 1, 18, 16, 17, 1, 17, 18, 25, 1,
         24, 25, 11, 4, 15, 11, 15, 0, 1, 1, 0, 1, 13, 4, 19, 18, 23, 6, 2, 8, 10, 12}
```

```
In[•]:= PhyloInter5Days = Partition[Riffle[PhyDiCorr5, Intersect5], {2}];

In[•]:= lmPhyloInter5Days = LinearModelFit[PhyloInter5Days, x, x]

Out[•]= FittedModel[ 17.7231 - 40.1901 x ]

In[•]:= bands90PhyloInter5Days[x_] =
          lmPhyloInter5Days["MeanPredictionBands", ConfidenceLevel → .95];
```

In[•]:=

```
InterPhylo1 =
  Plot[{lmPhyloInter5Days[x], bands90PhyloInter5Days[x]}, {x, -0.01, 0.4},
   PlotStyle → {Directive[RGBColor[18 / (255), 160 / (255), 255 / (255)], Thick],
     Lighter[Lighter[Gray]]}, Filling → {2 → {{1}, Lighter[Lighter[Gray]]}},
   PlotRange → {{-0.01, 0.4}, {-1.3, 30}}, Ticks → fuk[
     {{0, 0.1, 0.2, 0.3, 0.4}, {0, 5, 10, 15, 20, 25, 30}, "Arial", Plain, 20, 0.02}],
   TicksStyle → Thickness[0.004], AxesStyle → Thickness[0.004],
   AxesOrigin → {-0.01, -1.3}];

blN = Graphics[{EdgeForm[{RGBColor[18 / (255), 160 / (255), 255 / (255)]}],
    FaceForm[RGBColor[18 / (255), 160 / (255), 255 / (255)]], Disk[]}];
blG = Graphics[{EdgeForm[{Gray}], FaceForm[
     RGBColor[18 / (255), 160 / (255), 255 / (255)]], Disk[]}];
blB = Graphics[{EdgeForm[{Black}], FaceForm[
     RGBColor[18 / (255), 160 / (255), 255 / (255)]], Disk[]}];



InterPhylo2 = ListPlot[PhyloInter5Days, PlotMarkers → {blN, Scaled[0.035]}];
Show[InterPhylo1, InterPhylo2]
```

Out[•]=

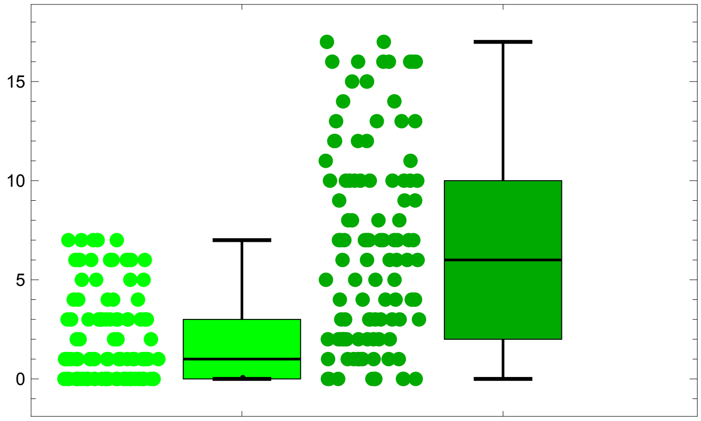


```
In[•]:= wi5 =
          {1, 6, 11, 16, 17, 26, 31, 33, 39, 44, 49, 56, 62, 68, 69, 74, 79, 84, 89, 95, 101, 107};

In[•]:= be5 = Complement[Range[108], wi5];

In[•]:= expPo5 = #[[7]] & /@ gk5

Out[•]= {1, 1, 1, 12, 17, 1, 16, 16, 2, 7, 0, 10, 15, 2, 10, 0, 2, 2, 14, 10, 6, 16, 6, 3, 0, 0, 7,
         5, 7, 8, 4, 13, 3, 6, 2, 4, 0, 10, 0, 4, 9, 15, 8, 3, 4, 7, 7, 0, 1, 10, 0, 3, 0, 3,
         9, 0, 8, 1, 10, 11, 3, 3, 2, 9, 2, 7, 3, 1, 6, 12, 5, 12, 10, 6, 7, 3, 2, 6, 0, 0, 4,
         13, 8, 0, 0, 7, 1, 2, 7, 11, 1, 7, 4, 16, 3, 4, 5, 7, 5, 13, 1, 3, 1, 13, 1, 3, 5, 0}

In[•]:= contrPo5 = #[[6]] & /@ gk5

Out[•]= {1, 1, 3, 0, 0, 0, 0, 0, 1, 0, 5, 2, 2, 1, 3, 7, 0, 1, 0, 0, 1, 0, 0, 0, 2, 4, 3, 5,
         3, 4, 5, 2, 4, 0, 3, 12, 6, 0, 9, 10, 2, 0, 1, 5, 4, 1, 1, 16, 2, 4, 4, 5, 12, 1,
         2, 5, 5, 9, 1, 1, 0, 7, 7, 1, 1, 3, 13, 13, 1, 0, 2, 1, 0, 0, 1, 4, 4, 0, 6, 6, 5,
         0, 6, 13, 5, 0, 10, 17, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 3, 5, 9, 0, 2, 3, 5, 7}


In[•]:= ExpansionWithin5 = expPo5[[#]] & /@ wi5

Out[•]= {1, 1, 0, 0, 2, 0, 4, 3, 0, 3, 1, 0, 3, 1, 6, 6, 0, 0, 7, 3, 1, 5}

In[•]:= ExpansionBetween5 = expPo5[[#]] & /@ be5

Out[•]= {1, 1, 12, 17, 16, 16, 2, 7, 10, 15, 2, 10, 2, 14, 10, 6, 16, 6,
         3, 0, 7, 5, 7, 8, 13, 6, 2, 4, 0, 10, 4, 9, 15, 8, 4, 7, 7, 0, 10, 0, 3,
         0, 3, 9, 8, 1, 10, 11, 3, 2, 9, 2, 7, 3, 12, 5, 12, 10, 7, 3, 2, 6, 0,
         4, 13, 8, 0, 7, 1, 2, 11, 1, 7, 4, 16, 4, 5, 7, 5, 13, 3, 1, 13, 1, 3, 0}

In[•]:= ℋE = MannWhitneyTest[
           {ExpansionWithin5, ExpansionBetween5}, 0, "HypothesisTestData"];
        ℋE["TestDataTable"]

Out[•]=               | Statistic  P-Value
        Mann-Whitney  | 421.5      0.0000598989
```

In[•]:=

```
A1 = BoxWhiskerChart[{ExpansionWithin5, ExpansionBetween5},
   {{"MedianMarker", 1, Thickness[0.004]}, {"Whiskers", Thickness[0.004]},
    {"Fences", Thick}}, ChartBaseStyle → EdgeForm[Black],
   ChartStyle → {Thick, {Green, Darker[Green]}}, Frame → True,
   FrameTicks → {None, {5, 10, 20}, None, None}];

wq2 := RandomReal[{-0.18, 0.18}]
jjA1 = Table[0.5 + wq2, Length[#[[5]] & /@ gk5]];
jjA2 = Table[1.5 + wq2, Length[#[[7]] & /@ gk5]];

grNBri = Graphics[{EdgeForm[{Green}], FaceForm[Green], Disk[]}];

A2 = ListPlot[Partition[Riffle[jjA1, ExpansionWithin5], {2}], PlotMarkers →
    {grNBri, Scaled[0.035]}, PlotStyle → Black, PlotRange → {{0, 4}, {0, 25}}];
A3 = ListPlot[Partition[Riffle[jjA2, ExpansionBetween5], {2}], PlotMarkers →
    {grN, Scaled[0.035]}, PlotStyle → Black, PlotRange → {{0, 4}, {0, 25}}];

Show[A1, A2, A3]
```

Out[•]=



In[•]:= `expPo5 = #[[7]] & /@ gk5`

Out[•]=

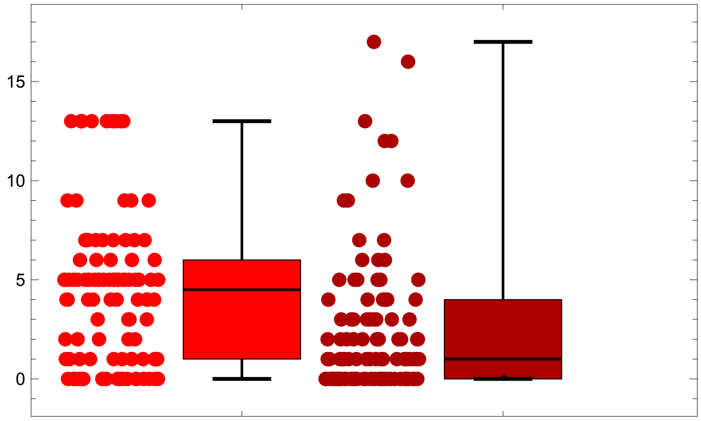
```
{1, 1, 1, 12, 17, 1, 16, 16, 2, 7, 0, 10, 15, 2, 10, 0, 2, 2, 14, 10, 6, 16, 6, 3, 0, 0, 7,
 5, 7, 8, 4, 13, 3, 6, 2, 4, 0, 10, 0, 4, 9, 15, 8, 3, 4, 7, 7, 0, 1, 10, 0, 3, 0, 3,
 9, 0, 8, 1, 10, 11, 3, 3, 2, 9, 2, 7, 3, 1, 6, 12, 5, 12, 10, 6, 7, 3, 2, 6, 0, 0, 4,
 13, 8, 0, 0, 7, 1, 2, 7, 11, 1, 7, 4, 16, 3, 4, 5, 7, 5, 13, 1, 3, 1, 13, 1, 3, 5, 0}
```

```
In[•]:= contrPo5 = #[[6]] & /@ gk5

Out[•]= {1, 1, 3, 0, 0, 0, 0, 0, 1, 0, 5, 2, 2, 1, 3, 7, 0, 1, 0, 0, 1, 0, 0, 0, 2, 4, 3, 5,
         3, 4, 5, 2, 4, 0, 3, 12, 6, 0, 9, 10, 2, 0, 1, 5, 4, 1, 1, 16, 2, 4, 4, 5, 12, 1,
         2, 5, 5, 9, 1, 1, 0, 7, 7, 1, 1, 3, 13, 13, 1, 0, 2, 1, 0, 0, 1, 4, 4, 0, 6, 6, 5,
         0, 6, 13, 5, 0, 10, 17, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 3, 5, 9, 0, 2, 3, 5, 7}

In[•]:= ContractionWithin5 = contrPo5[[#]] & /@ wi5

Out[•]= {1, 0, 5, 7, 0, 4, 5, 4, 9, 5, 2, 5, 7, 13, 1, 0, 6, 13, 0, 1, 3, 5}

In[•]:= ContractionBetween5 = contrPo5[[#]] & /@ be5

Out[•]= {1, 3, 0, 0, 0, 0, 1, 0, 2, 2, 1, 3, 1, 0, 0, 1, 0, 0, 0, 2, 3, 5, 3, 4, 2, 0, 3, 12, 6, 0,
         10, 2, 0, 1, 4, 1, 1, 16, 4, 4, 5, 12, 1, 2, 5, 9, 1, 1, 0, 7, 1, 1, 3, 13, 0, 2, 1, 0,
         1, 4, 4, 0, 6, 5, 0, 6, 5, 0, 10, 17, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 5, 9, 0, 2, 3, 7}

In[•]:= 𝓗C = MannWhitneyTest[
           {ContractionWithin5, ContractionBetween5}, 0, "HypothesisTestData"];
        𝓗C["TestDataTable"]

Out[•]=               | Statistic  P-Value
        Mann-Whitney  | 1202.      0.0467876
```

In[∘]:=

```
B1 = BoxWhiskerChart[{ContractionWithin5, ContractionBetween5},
   {{"MedianMarker", 1, Thickness[0.004]}, {"Whiskers", Thickness[0.004]},
    {"Fences", Thick}}, ChartBaseStyle → EdgeForm[Black],
   ChartStyle → {Thick, {Red, Darker[Red]}}, Frame → True,
   FrameTicks → {None, {5, 10, 20}, None, None}];

wq2 := RandomReal[{-0.18, 0.18}]
jjB1 = Table[0.5 + wq2, Length[#[[5]] & /@ gk5]];
jjB2 = Table[1.5 + wq2, Length[#[[7]] & /@ gk5]];

rdNBri = Graphics[{EdgeForm[{Red}], FaceForm[Red], Disk[]}];

B2 = ListPlot[Partition[Riffle[jjB1, ContractionWithin5], {2}], PlotMarkers →
    {rdNBri, Scaled[0.035]}, PlotStyle → Black, PlotRange → {{0, 4}, {0, 25}}];
B3 = ListPlot[Partition[Riffle[jjB2, ContractionBetween5], {2}], PlotMarkers →
    {rdN, Scaled[0.035]}, PlotStyle → Black, PlotRange → {{0, 4}, {0, 25}}];

Show[B1, B2, B3]
```

Out[∘]=



The role of Species and Auxotrophy type on Niche Expansion and Contraction

Species

(*Expansion*)

```
(*Expansion*)

In[•]:= expan = #[[7]] & /@ gk5KJaccK

Out[•]= {1, 1, 1, 12, 17, 1, 16, 16, 2, 7, 0, 10, 15, 2, 10, 0, 2, 2, 14, 10, 6, 16, 6, 3, 0, 0, 7,
         5, 7, 8, 4, 13, 3, 6, 2, 4, 0, 10, 0, 4, 9, 15, 8, 3, 4, 7, 7, 0, 1, 10, 0, 3, 0, 3,
         9, 0, 8, 1, 10, 11, 3, 3, 2, 9, 2, 7, 3, 1, 6, 12, 5, 12, 10, 6, 7, 3, 2, 6, 0, 0, 4,
         13, 8, 0, 0, 7, 1, 2, 7, 11, 1, 7, 4, 16, 3, 4, 5, 7, 5, 13, 1, 3, 1, 13, 1, 3, 5, 0}

(*Contraction*)

In[•]:= contrac = #[[6]] & /@ gk5KJaccK

Out[•]= {1, 1, 3, 0, 0, 0, 0, 0, 1, 0, 5, 2, 2, 1, 3, 7, 0, 1, 0, 0, 1, 0, 0, 0, 2, 4, 3, 5,
         3, 4, 5, 2, 4, 0, 3, 12, 6, 0, 9, 10, 2, 0, 1, 5, 4, 1, 1, 16, 2, 4, 4, 5, 12, 1,
         2, 5, 5, 9, 1, 1, 0, 7, 7, 1, 1, 3, 13, 13, 1, 0, 2, 1, 0, 0, 1, 4, 4, 0, 6, 6, 5,
         0, 6, 13, 5, 0, 10, 17, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 3, 5, 9, 0, 2, 3, 5, 7}

(*Super-Expansion*)

In[•]:= supexp = #[[8]] & /@ gk5KJaccK

Out[•]= {0, 0, 0, 0, 5, 1, 1, 8, 0, 1, 0, 0, 1, 0, 2, 0, 1, 0, 5, 1, 1, 3, 2, 0, 0, 0, 0,
         0, 1, 1, 2, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 7, 1, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0,
         0, 0, 1, 0, 3, 3, 1, 1, 0, 1, 0, 1, 0, 0, 1, 3, 1, 1, 1, 5, 1, 0, 0, 1, 0, 0, 0,
         2, 1, 0, 0, 0, 0, 0, 3, 2, 1, 2, 0, 2, 2, 1, 1, 0, 2, 0, 0, 1, 0, 3, 0, 0, 1, 0}
```

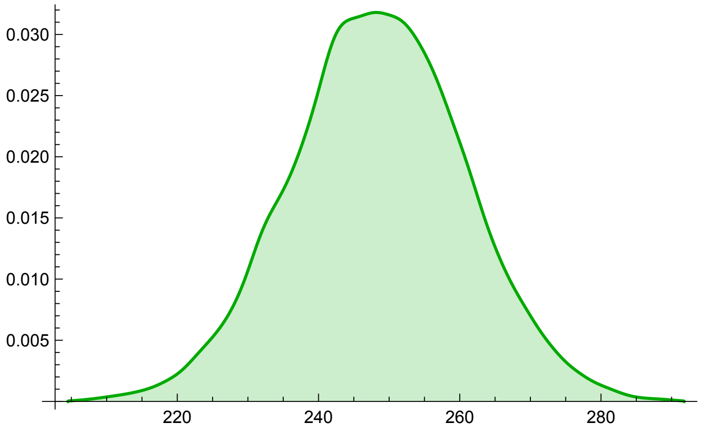
```
In[•]:= jko = {#[[1]], #[[2]]} & /@ jop5

Out[•]= {{ABR5, ABH5}, {ABR5, BSH5}, {ABR5, ECH5}, {ABR5, SOH5}, {BSR5, ABH5}, {BSR5, BSH5},
         {BSR5, ECH5}, {BSR5, SOH5}, {ECR5, ABH5}, {ECR5, BSH5}, {ECR5, ECH5}, {ECR5, SOH5},
         {SOR5, ABH5}, {SOR5, BSH5}, {SOR5, ECH5}, {SOR5, SOH5}, {ABH5, ABW5}, {ABH5, ECW5},
         {ABH5, SOW5}, {ABH5, PFW5}, {BSH5, ABW5}, {BSH5, ECW5}, {BSH5, SOW5}, {BSH5, PFW5},
         {ECH5, ABW5}, {ECH5, ECW5}, {ECH5, SOW5}, {ECH5, PFW5}, {SOH5, ABW5}, {SOH5, ECW5},
         {SOH5, SOW5}, {SOH5, PFW5}, {ABW5, ABR5}, {ABW5, BSR5}, {ABW5, ECR5}, {ABW5, SOR5},
         {ECW5, ABR5}, {ECW5, BSR5}, {ECW5, ECR5}, {ECW5, SOR5}, {SOW5, ABR5}, {SOW5, BSR5},
         {SOW5, ECR5}, {SOW5, SOR5}, {PFW5, ABR5}, {PFW5, BSR5}, {PFW5, ECR5}, {PFW5, SOR5},
         {ABW5, ABL5}, {ABW5, BSL5}, {ABW5, ECL5}, {ABW5, SOL5}, {ABW5, PFL5}, {ECW5, ABL5},
         {ECW5, BSL5}, {ECW5, ECL5}, {ECW5, SOL5}, {ECW5, PFL5}, {SOW5, ABL5}, {SOW5, BSL5},
         {SOW5, ECL5}, {SOW5, SOL5}, {SOW5, PFL5}, {PFW5, ABL5}, {PFW5, BSL5}, {PFW5, ECL5},
         {PFW5, SOL5}, {PFW5, PFL5}, {ABL5, ABR5}, {ABL5, BSR5}, {ABL5, ECR5}, {ABL5, SOR5},
         {BSL5, ABR5}, {BSL5, BSR5}, {BSL5, ECR5}, {BSL5, SOR5}, {ECL5, ABR5}, {ECL5, BSR5},
         {ECL5, ECR5}, {ECL5, SOR5}, {SOL5, ABR5}, {SOL5, BSR5}, {SOL5, ECR5}, {SOL5, SOR5},
         {PFL5, ABR5}, {PFL5, BSR5}, {PFL5, ECR5}, {PFL5, SOR5}, {ABH5, ABL5}, {ABH5, BSL5},
         {ABH5, ECL5}, {ABH5, SOL5}, {ABH5, PFL5}, {BSH5, ABL5}, {BSH5, BSL5}, {BSH5, ECL5},
         {BSH5, SOL5}, {BSH5, PFL5}, {ECH5, ABL5}, {ECH5, BSL5}, {ECH5, ECL5}, {ECH5, SOL5},
         {ECH5, PFL5}, {SOH5, ABL5}, {SOH5, BSL5}, {SOH5, ECL5}, {SOH5, SOL5}, {SOH5, PFL5}}

In[•]:= hkk =
         {StringTake[jko[[#]][[1]], 1], StringTake[jko[[#]][[2]], 1]} & /@ Range[Length[jko]]

Out[•]= {{A, A}, {A, B}, {A, E}, {A, S}, {B, A}, {B, B}, {B, E}, {B, S}, {E, A},
         {E, B}, {E, E}, {E, S}, {S, A}, {S, B}, {S, E}, {S, S}, {A, A}, {A, E},
         {A, S}, {A, P}, {B, A}, {B, E}, {B, S}, {B, P}, {E, A}, {E, E}, {E, S}, {E, P},
         {S, A}, {S, E}, {S, S}, {S, P}, {A, A}, {A, B}, {A, E}, {A, S}, {E, A}, {E, B},
         {E, E}, {E, S}, {S, A}, {S, B}, {S, E}, {S, S}, {P, A}, {P, B}, {P, E}, {P, S},
         {A, A}, {A, B}, {A, E}, {A, S}, {A, P}, {E, A}, {E, B}, {E, E}, {E, S}, {E, P},
         {S, A}, {S, B}, {S, E}, {S, S}, {S, P}, {P, A}, {P, B}, {P, E}, {P, S}, {P, P},
         {A, A}, {A, B}, {A, E}, {A, S}, {B, A}, {B, B}, {B, E}, {B, S}, {E, A}, {E, B},
         {E, E}, {E, S}, {S, A}, {S, B}, {S, E}, {S, S}, {P, A}, {P, B}, {P, E}, {P, S},
         {A, A}, {A, B}, {A, E}, {A, S}, {A, P}, {B, A}, {B, B}, {B, E}, {B, S}, {B, P},
         {E, A}, {E, B}, {E, E}, {E, S}, {E, P}, {S, A}, {S, B}, {S, E}, {S, S}, {S, P}}
```

```
In[•]:= PosA = MemberQ[hkk[[#]], "A"] & /@ Range[Length[hkk]]

Out[•]= {True, True, True, True, True, False, False, False, True, False, False, False, True,
         False, False, False, True, True, True, True, True, False, False, False, True,
         False, False, False, True, False, False, False, True, True, True, True, True,
         False, False, False, True, False, False, False, True, False, False, False, True,
         True, True, True, True, True, False, False, False, False, True, False, False,
         False, False, True, False, False, False, False, True, True, True, True, True,
         False, False, False, True, False, False, False, True, False, False, False, True,
         False, False, False, True, True, True, True, True, True, False, False, False,
         False, True, False, False, False, False, True, False, False, False, False}

In[•]:= PosB = MemberQ[hkk[[#]], "B"] & /@ Range[Length[hkk]];

In[•]:= PosE = MemberQ[hkk[[#]], "E"] & /@ Range[Length[hkk]];

In[•]:= PosS = MemberQ[hkk[[#]], "S"] & /@ Range[Length[hkk]];

In[•]:= PosP = MemberQ[hkk[[#]], "P"] & /@ Range[Length[hkk]];

        (***)


        (***)

In[•]:= Count[PosA, True] / Length[PosA] // N

Out[•]= 0.416667

In[•]:= Count[PosB, True] / Length[PosA] // N

Out[•]= 0.324074

In[•]:= Count[PosE, True] / Length[PosA] // N

Out[•]= 0.416667

In[•]:= Count[PosS, True] / Length[PosA] // N

Out[•]= 0.416667

In[•]:= Count[PosP, True] / Length[PosA] // N

Out[•]= 0.222222


In[•]:= jooA = PosA /. {True → 1, False → 0}

Out[•]= {1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0,
         0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1,
         0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1,
         0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0}
```

```
In[•]:= jooB = PosB /. {True → 1, False → 0};
        jooE = PosE /. {True → 1, False → 0};
        jooS = PosS /. {True → 1, False → 0};
        jooP = PosP /. {True → 1, False → 0};


        (*Observed*)

In[•]:= Total[expan jooA]

Out[•]= 269

In[•]:= Total[expan jooB]

Out[•]= 285

In[•]:= Total[expan jooE]

Out[•]= 206

In[•]:= Total[expan jooS]

Out[•]= 289

In[•]:= Total[expan jooP]

Out[•]= 96

In[•]:= N[{Total[expan jooA], Total[expan jooB], Total[expan jooE],
           Total[expan jooS], Total[expan jooP]} / (Total[expan jooA] +
             Total[expan jooB] + Total[expan jooE] + Total[expan jooS] + Total[expan jooP])]

Out[•]= {0.234934, 0.248908, 0.179913, 0.252402, 0.0838428}


        (** A: 269  *)
```

```
In[•]:= PermuExpanA = Table[(
            hoz = Tally[Sort[RandomInteger[{1, Length[expan]}, Total[expan]]]];
            guq = ConstantArray[0, Length[expan]];
            hoz1 = #[[1]] → #[[2]] & /@ hoz;
            expanSimul = ReplacePart[guq, hoz1];
            Total[expanSimul jooA]), {10 000}];

         SmoothHistogram[PermuExpanA, PlotStyle → Darker[Green], Filling → Axis]
```

Out[•]=

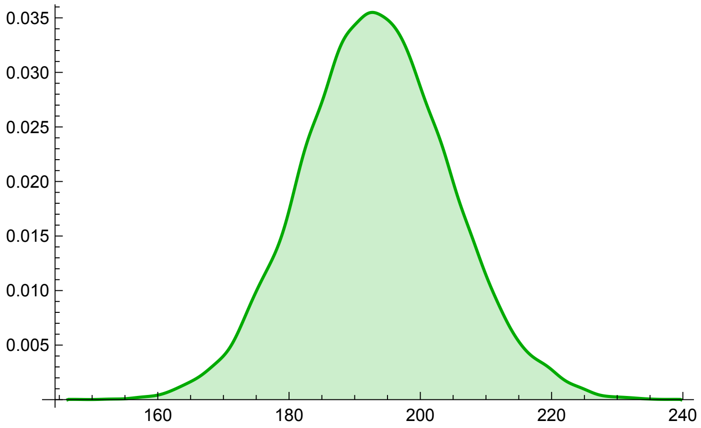


```
In[•]:= (# ≥ 269) & /@ PermuExpanA
        Count[(# ≥ 269) & /@ PermuExpanA, True]

Out[•]= 6

In[•]:= N[Count[(# ≥ 269) & /@ PermuExpanA, True] / Length[PermuExpanA]]

Out[•]= 0.0491
```

```
(** B: 285  *)
```

```
In[•]:=  PermuExpanB = Table[(
             hoz = Tally[Sort[RandomInteger[{1, Length[expan]}, Total[expan]]]];
             guq = ConstantArray[0, Length[expan]];
             hoz1 = #[[1]] → #[[2]] & /@ hoz;
             expanSimul = ReplacePart[guq, hoz1];
             Total[expanSimul jooB]), {10 000}];

         SmoothHistogram[PermuExpanB, PlotStyle → Darker[Green], Filling → Axis]
```

Out[•]=

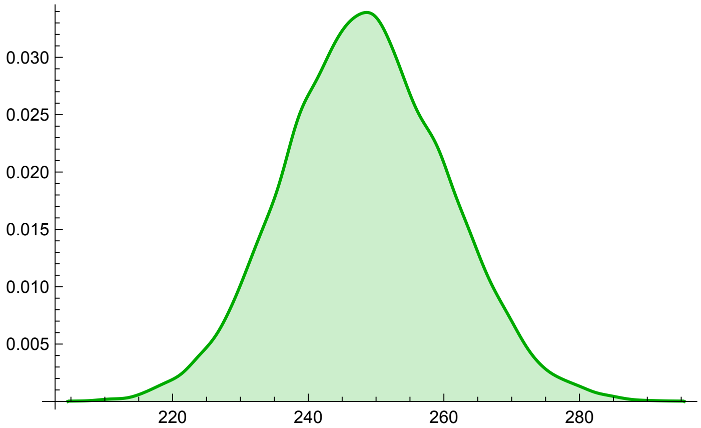


```
In[•]:=  N[Count[(# ≥  285) & /@ PermuExpanB, True] / Length[PermuExpanB]]
Out[•]=  0.

In[•]:=  Mean[PermuExpanB] // N
Out[•]=  193.048
```

```
(** E: 206  *)
```

```
In[•]:= PermuExpanE = Table[(
             hoz = Tally[Sort[RandomInteger[{1, Length[expan]}, Total[expan]]]];
             guq = ConstantArray[0, Length[expan]];
             hoz1 = #[[1]] → #[[2]] & /@ hoz;
             expanSimul = ReplacePart[guq, hoz1];
             Total[expanSimul jooE]), {10 000}];

         SmoothHistogram[PermuExpanE, PlotStyle → Darker[Green], Filling → Axis]
```

Out[•]=

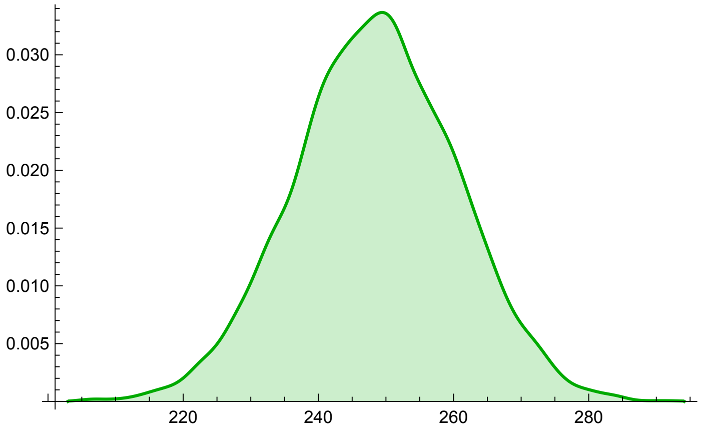


```
In[•]:= N[Count[(# ≤  206) & /@ PermuExpanE, True] / Length[PermuExpanE]]

Out[•]= 0.0003

In[•]:= Mean[PermuExpanE] // N

Out[•]= 248.118
```

```
(** S: 289  *)
```

```
In[•]:= PermuExpanS = Table[(
            hoz = Tally[Sort[RandomInteger[{1, Length[expan]}, Total[expan]]]];
            guq = ConstantArray[0, Length[expan]];
            hoz1 = #[[1]] → #[[2]] & /@ hoz;
            expanSimul = ReplacePart[guq, hoz1];
            Total[expanSimul jooS]), {10 000}];

        SmoothHistogram[PermuExpanS, PlotStyle → Darker[Green], Filling → Axis]
```

Out[•]=

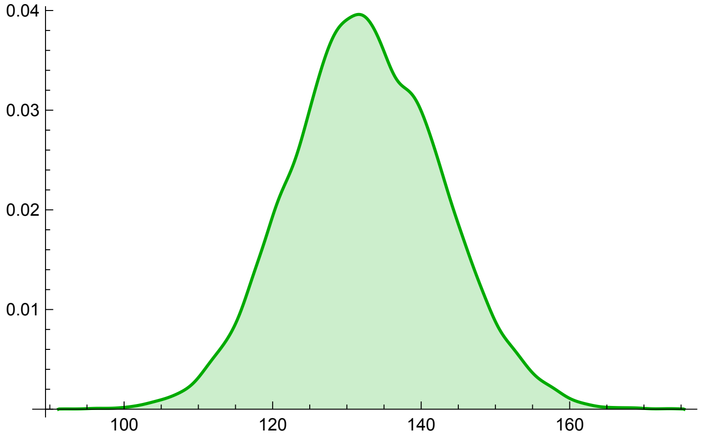


```
In[•]:= N[Count[(# ≥ 289) & /@ PermuExpanS, True] / Length[PermuExpanS]]

Out[•]= 0.0007
```

```
(** P: 96  *)
```

In[•]:=
```
PermuExpanP = Table[(
    hoz = Tally[Sort[RandomInteger[{1, Length[expan]}, Total[expan]]]];
    guq = ConstantArray[0, Length[expan]];
    hoz1 = #[[1]] → #[[2]] & /@ hoz;
    expanSimul = ReplacePart[guq, hoz1];
    Total[expanSimul jooP]), {10 000}];

SmoothHistogram[PermuExpanP, PlotStyle → Darker[Green], Filling → Axis]
```

Out[•]=



In[•]:=
```
N[Count[(# ≤  96) & /@ PermuExpanP, True] / Length[PermuExpanP]]
```

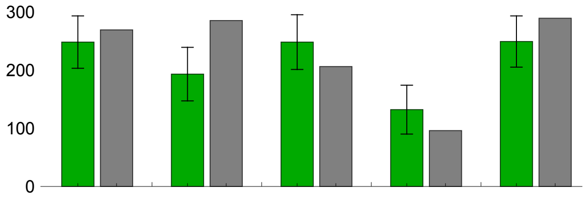
Out[•]=
```
0.0001
```

```
(****)
```

In[•]:=
```
Spec3 = {{Around[248, 293 - 248], 200},
   {Around[193, 239 - 193], 200}, {Around[248, 295 - 248], 200},
   {Around[132, 174 - 132], 200}, {Around[249, 293 - 249], 200}};
```

In[•]:=
```
BarChart[Spec3, ChartStyle → {Darker[Green], Gray}, PlotTheme → "Web",
 BarSpacing → {0.2, 1.2}, ImageSize → 300, AspectRatio → 1 / 3]
```

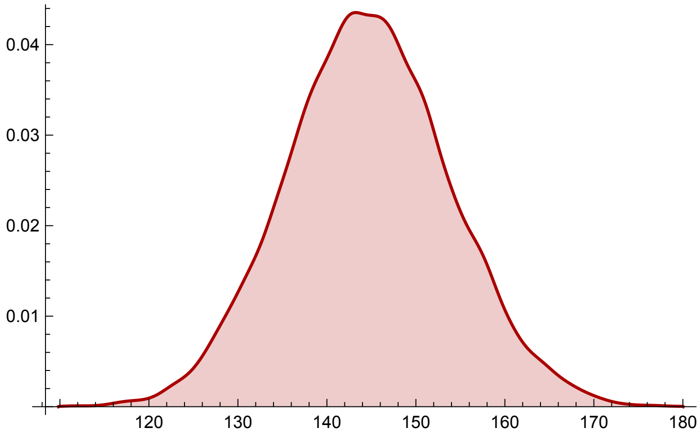
Out[•]=



```
(*Contraction*)

(** A: 98  *)
```

```
In[•]:= PermuContracA = Table[(
            hoz = Tally[Sort[RandomInteger[{1, Length[contrac]}, Total[contrac]]]];
            guq = ConstantArray[0, Length[contrac]];
            hoz1 = #[[1]] → #[[2]] & /@ hoz;
            contracSimul = ReplacePart[guq, hoz1];
            Total[contracSimul jooA]), {10 000}];

        SmoothHistogram[PermuContracA, PlotStyle → Darker[Red], Filling → Axis]
```

Out[•]=

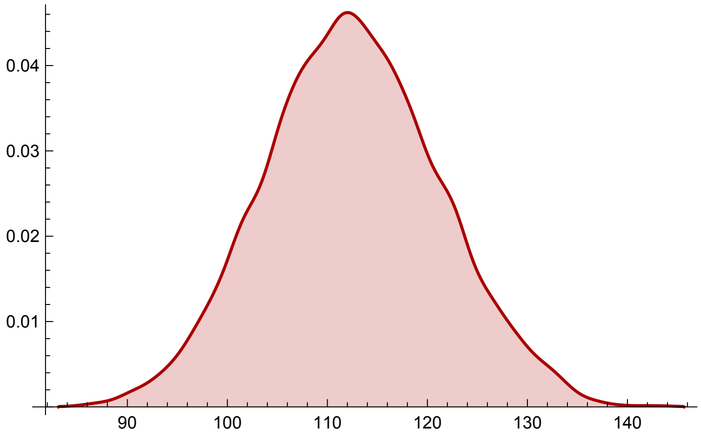


```
In[•]:= (# ≤  98) & /@ PermuContracA

In[•]:= Count[(# ≤  98) & /@ PermuContracA, True]

Out[•]= 0
```

```
(** B: 22  *)
```

```
In[•]:= PermuContracB = Table[(
            hoz = Tally[Sort[RandomInteger[{1, Length[contrac]}, Total[contrac]]]];
            guq = ConstantArray[0, Length[contrac]];
            hoz1 = #[[1]] → #[[2]] & /@ hoz;
            contracSimul = ReplacePart[guq, hoz1];
            Total[contracSimul jooB]), {10 000}];

        SmoothHistogram[PermuContracB, PlotStyle → Darker[Red], Filling → Axis]
```

Out[•]=



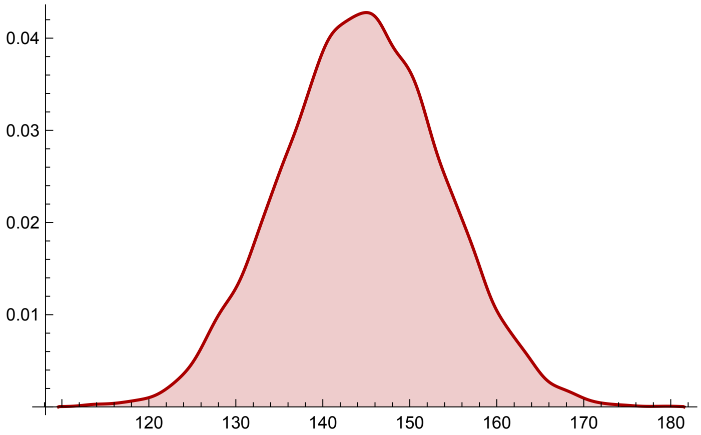
```
In[•]:= N[Count[(# ≤  22) & /@ PermuContracB, True] / Length[PermuContracB]]

Out[•]= 0.
```

```
(** E: 149  *)
```

In[∘]:=

```
PermuContracE = Table[(
    hoz = Tally[Sort[RandomInteger[{1, Length[contrac]}, Total[contrac]]]];
    guq = ConstantArray[0, Length[contrac]];
    hoz1 = #[[1]] → #[[2]] & /@ hoz;
    contracSimul = ReplacePart[guq, hoz1];
    Total[contracSimul jooE]), {10 000}];

SmoothHistogram[PermuContracE, PlotStyle → Darker[Red], Filling → Axis]
```

Out[∘]=



In[∘]:=

```
N[Count[(# ≤  149) & /@ PermuContracE, True] / Length[PermuContracE]]
N[Count[(# ≥  149) & /@ PermuContracE, True] / Length[PermuContracE]]
```

Out[∘]=

```
0.7111
```

Out[∘]=

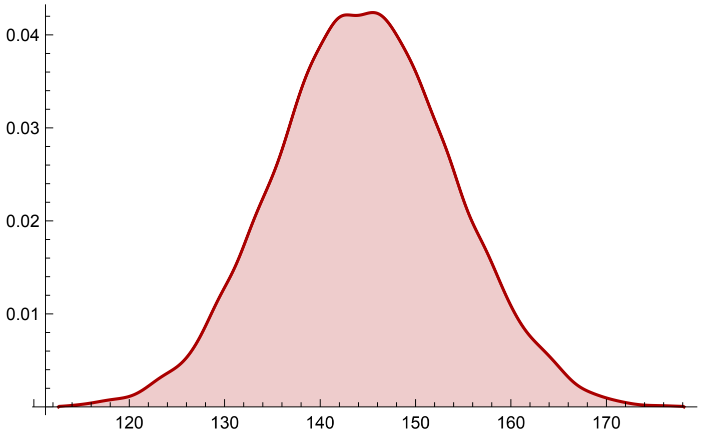
```
0.3266
```

```
(** S: 192  *)
```

In[•]:=

```
PermuContracS = Table[(
    hoz = Tally[Sort[RandomInteger[{1, Length[contrac]}, Total[contrac]]]];
    guq = ConstantArray[0, Length[contrac]];
    hoz1 = #[[1]] → #[[2]] & /@ hoz;
    contracSimul = ReplacePart[guq, hoz1];
    Total[contracSimul jooS]), {10 000}];

SmoothHistogram[PermuContracS, PlotStyle → Darker[Red], Filling → Axis]
```

Out[•]=



In[•]:=

```
N[Count[(# ≥  192) & /@ PermuContracS, True] / Length[PermuContracS]]
```

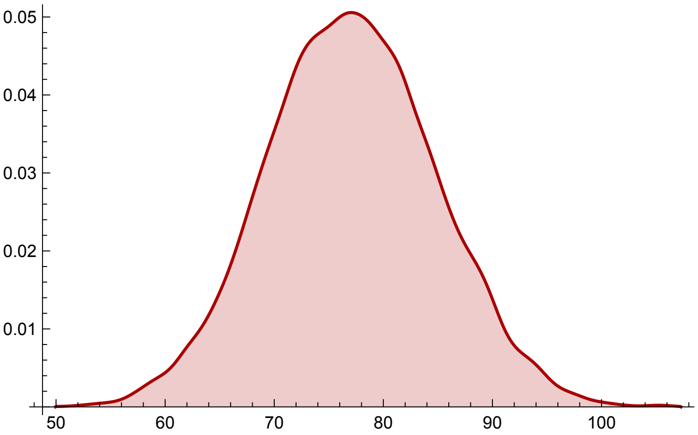
Out[•]=

```
0.
```

```
(** P: 137  *)
```

In[◦]:=
```
PermuContracP = Table[(
    hoz = Tally[Sort[RandomInteger[{1, Length[contrac]}, Total[contrac]]]];
    guq = ConstantArray[0, Length[contrac]];
    hoz1 = #[[1]] → #[[2]] & /@ hoz;
    contracSimul = ReplacePart[guq, hoz1];
    Total[contracSimul jooP]), {10 000}];

SmoothHistogram[PermuContracP, PlotStyle → Darker[Red], Filling → Axis]
```

Out[◦]=



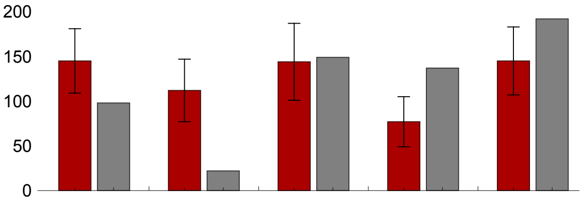
In[◦]:=
```
N[Count[(# ≥   137) & /@ PermuContracP, True] / Length[PermuContracP]]
```

Out[◦]=
```
0.
```

In[◦]:=
```
Spec3Con = {{Around[145, 181 - 145], 98},
   {Around[112, 147 - 112], 22}, {Around[144, 187 - 144], 149},
   {Around[77, 105 - 77], 137}, {Around[145, 183 - 145], 192}};
```

In[◦]:=
```
BarChart[Spec3Con, ChartStyle → {Darker[Red], Gray}, PlotTheme → "Web",
 BarSpacing → {0.2, 1.2}, ImageSize → 300, AspectRatio → 1/3]
```

Out[◦]=

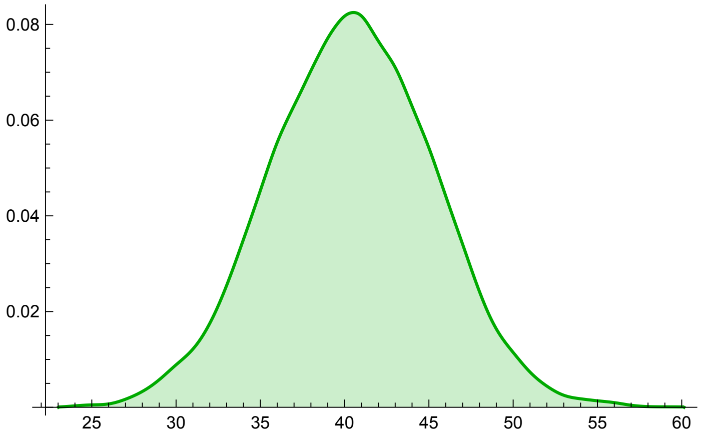


```
(*Double Expansion*)
```

```
(** A: 44  *)
```

```
In[•]:= PermuSuperExpanA = Table[(
            hoz = Tally[Sort[RandomInteger[{1, Length[supexp]}, Total[supexp]]]];
            guq = ConstantArray[0, Length[supexp]];
            hoz1 = #[[1]] → #[[2]] & /@ hoz;
            SuperexpanSimul = ReplacePart[guq, hoz1];
            Total[SuperexpanSimul jooA]), {10 000}];

        SmoothHistogram[PermuSuperExpanA, PlotStyle → Darker[Green], Filling → Axis]
```

Out[•]=

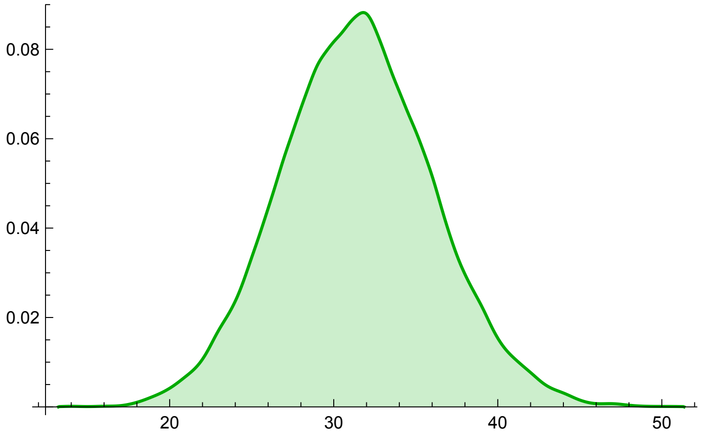


```
(# ≥ 44) & /@ PermuSuperExpanA

Count[(# ≥ 44) & /@ PermuSuperExpanA, True]

Out[•]= 6

In[•]:= N[Count[(# ≥ 44) & /@ PermuSuperExpanA, True] / Length[PermuSuperExpanA]]
        N[Count[(# ≤  44) & /@ PermuSuperExpanA, True] / Length[PermuSuperExpanA]]

Out[•]= 0.259

Out[•]= 0.8029
```

```
(** B: 57  *)
```

```
In[•]:= PermuSuperExpanB = Table[(
            hoz = Tally[Sort[RandomInteger[{1, Length[supexp]}, Total[supexp]]]];
            guq = ConstantArray[0, Length[supexp]];
            hoz1 = #[[1]] → #[[2]] & /@ hoz;
            SuperexpanSimul = ReplacePart[guq, hoz1];
            Total[SuperexpanSimul jooB]), {10 000}];

        SmoothHistogram[PermuSuperExpanB, PlotStyle → Darker[Green], Filling → Axis]
```

Out[•]=



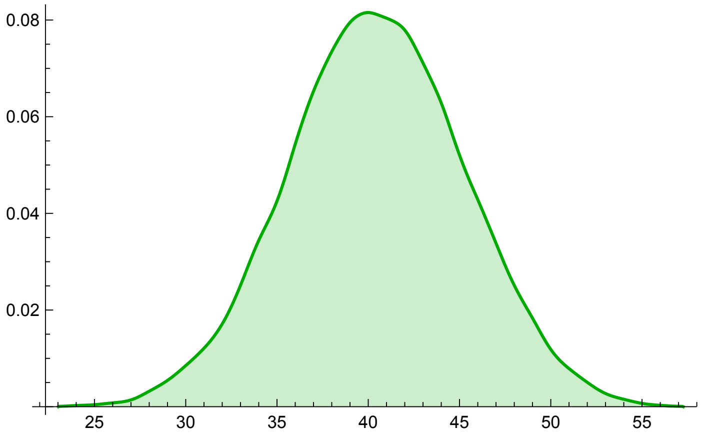
```
In[•]:= N[Count[(# ≥  57) & /@ PermuSuperExpanB, True] / Length[PermuSuperExpanB]]

Out[•]= 0.

In[•]:= Mean[PermuSuperExpanB] // N

Out[•]= 31.481
```

```
(** E: 22  *)
```

In[•]:=

```
PermuSuperExpanE = Table[(
    hoz = Tally[Sort[RandomInteger[{1, Length[supexp]}, Total[supexp]]]];
    guq = ConstantArray[0, Length[supexp]];
    hoz1 = #[[1]] → #[[2]] & /@ hoz;
    SuperexpanSimul = ReplacePart[guq, hoz1];
    Total[SuperexpanSimul jooE]), {10 000}];

SmoothHistogram[PermuSuperExpanE, PlotStyle → Darker[Green], Filling → Axis]
```

Out[•]=



In[•]:=

```
N[Count[(# ≤  22) & /@ PermuSuperExpanE, True] / Length[PermuSuperExpanE]]
```

Out[•]=

```
0.
```

In[•]:=

```
Mean[PermuSuperExpanE] // N
```

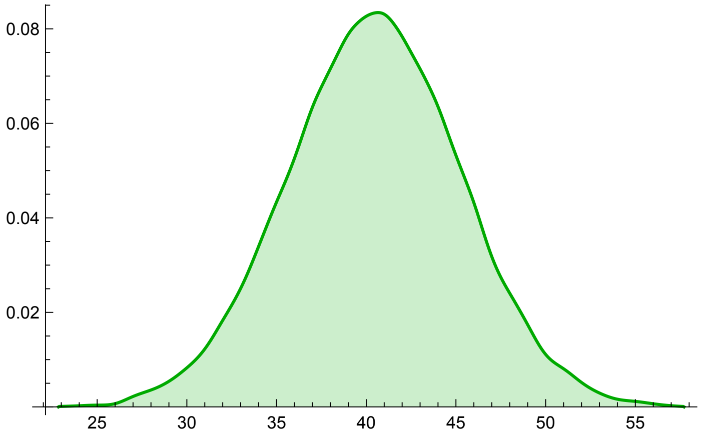
Out[•]=

```
40.2534
```

```
(** S: 51  *)
```

```
In[•]:= PermuSuperExpanS = Table[(
            hoz = Tally[Sort[RandomInteger[{1, Length[supexp]}, Total[supexp]]]];
            guq = ConstantArray[0, Length[supexp]];
            hoz1 = #[[1]] → #[[2]] & /@ hoz;
            SuperexpanSimul = ReplacePart[guq, hoz1];
            Total[SuperexpanSimul jooS]), {10 000}];

        SmoothHistogram[PermuSuperExpanS, PlotStyle → Darker[Green], Filling → Axis]
```

Out[•]=

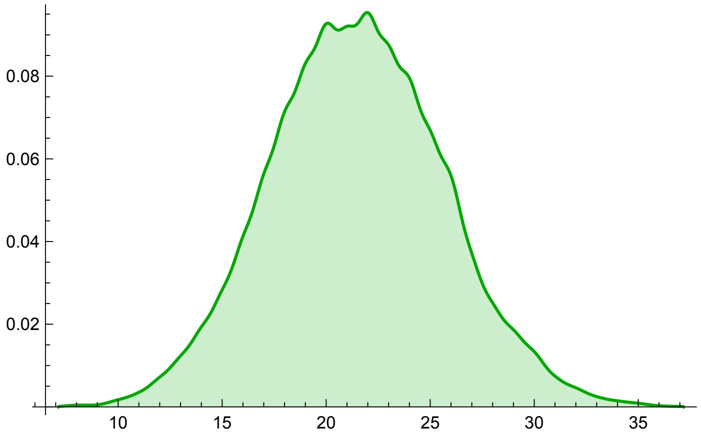


```
In[•]:= N[Count[(# ≥  51) & /@ PermuSuperExpanS, True] / Length[PermuSuperExpanS]]
Out[•]= 0.0206

In[•]:= Mean[PermuSuperExpanS] // N
Out[•]= 40.3703
```

```
(** P: 3  *)
```

```
In[•]:= PermuSuperExpanP = Table[(
            hoz = Tally[Sort[RandomInteger[{1, Length[supexp]}, Total[supexp]]]];
            guq = ConstantArray[0, Length[supexp]];
            hoz1 = #[[1]] → #[[2]] & /@ hoz;
            SuperexpanSimul = ReplacePart[guq, hoz1];
            Total[SuperexpanSimul jooP]), {10 000}];

        SmoothHistogram[PermuSuperExpanP, PlotStyle → Darker[Green], Filling → Axis]
```

Out[•]=

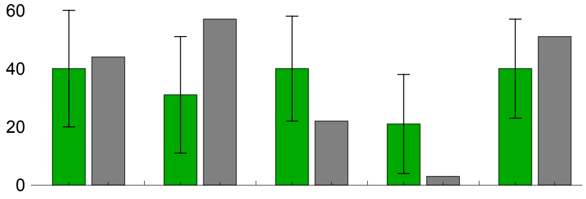


```
In[•]:= N[Count[(# ≤   3) & /@ PermuSuperExpanP, True] / Length[PermuSuperExpanP]]
Out[•]= 0.

In[•]:= N[Mean[PermuSuperExpanP]]
Out[•]= 21.5339
```

```
In[•]:= Spec3NE = {{Around[40, 60 - 40], 44}, {Around[31, 51 - 31], 57},
           {Around[40, 58 - 40], 22}, {Around[21, 38 - 21], 3}, {Around[40, 57 - 40], 51}};
```

```
In[•]:= BarChart[Spec3NE, ChartStyle → {Darker[Green], Gray}, PlotTheme → "Web",
          BarSpacing → {0.2, 1.2}, ImageSize → 300, AspectRatio → 1/3]
```

Out[•]=



Auxotrophy

```
In[∘]:= hkkF = {StringTake[jko[[#]][[1]], {3}], StringTake[jko[[#]][[2]], {3}]} & /@
          Range[Length[jko]]

Out[∘]= {{R, H}, {R, H}, {R, H}, {R, H}, {R, H}, {R, H}, {R, H}, {R, H}, {R, H},
         {R, H}, {R, H}, {R, H}, {R, H}, {R, H}, {R, H}, {R, H}, {H, W}, {H, W},
         {H, W}, {H, W}, {H, W}, {H, W}, {H, W}, {H, W}, {H, W}, {H, W}, {H, W}, {H, W},
         {H, W}, {H, W}, {H, W}, {H, W}, {W, R}, {W, R}, {W, R}, {W, R}, {W, R}, {W, R},
         {W, R}, {W, R}, {W, R}, {W, R}, {W, R}, {W, R}, {W, R}, {W, R}, {W, R}, {W, R},
         {W, L}, {W, L}, {W, L}, {W, L}, {W, L}, {W, L}, {W, L}, {W, L}, {W, L}, {W, L},
         {W, L}, {W, L}, {W, L}, {W, L}, {W, L}, {W, L}, {W, L}, {W, L}, {W, L}, {W, L},
         {L, R}, {L, R}, {L, R}, {L, R}, {L, R}, {L, R}, {L, R}, {L, R}, {L, R}, {L, R},
         {L, R}, {L, R}, {L, R}, {L, R}, {L, R}, {L, R}, {L, R}, {L, R}, {L, R}, {L, R},
         {H, L}, {H, L}, {H, L}, {H, L}, {H, L}, {H, L}, {H, L}, {H, L}, {H, L}, {H, L},
         {H, L}, {H, L}, {H, L}, {H, L}, {H, L}, {H, L}, {H, L}, {H, L}, {H, L}, {H, L}}
```

```
In[∘]:= DeleteDuplicates[Flatten[hkkF]]

Out[∘]= {R, H, W, L}
```

```
In[∘]:= PosR = MemberQ[hkkF[[#]], "R"] & /@ Range[Length[hkkF]]

Out[∘]= {True, True, True, True, True, True, True, True, True, True, True, True, True, True,
         True, True, False, False, False, False, False, False, False, False, False,
         False, False, False, False, False, False, False, True, True, True, True, True,
         True, True, True, True, True, True, True, True, True, True, True, False, False,
         False, False, False, False, False, False, False, False, False, False, False,
         False, False, False, False, False, False, False, True, True, True, True, True,
         True, True, True, True, True, True, True, True, True, True, True, True, True,
         True, True, False, False, False, False, False, False, False, False, False,
         False, False, False, False, False, False, False, False, False, False, False}
```

```
In[∘]:= PosH = MemberQ[hkkF[[#]], "H"] & /@ Range[Length[hkkF]];

In[∘]:= PosW = MemberQ[hkkF[[#]], "W"] & /@ Range[Length[hkkF]];

In[∘]:= PosL = MemberQ[hkkF[[#]], "L"] & /@ Range[Length[hkkF]];
```

```
(***)
```

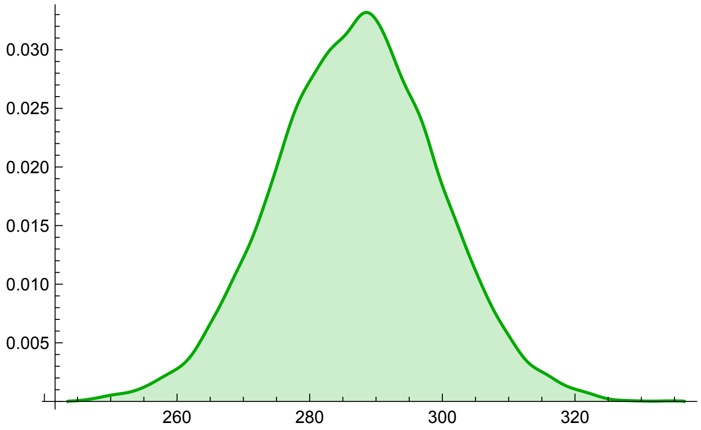
```
In[∘]:= Count[PosR, True]

Out[∘]= 52
```

```
In[•]:= Count[PosH, True]
Out[•]= 52

In[•]:= Count[PosW, True]
Out[•]= 52

In[•]:= Count[PosL, True]
Out[•]= 60



In[•]:= Length[PosR]
Out[•]= 108

        (***)
In[•]:= Count[PosR, True] / Length[PosR] // N
Out[•]= 0.481481

In[•]:= Count[PosH, True] / Length[PosH] // N
Out[•]= 0.481481

In[•]:= Count[PosW, True] / Length[PosW] // N
Out[•]= 0.481481

In[•]:= Count[PosL, True] / Length[PosL] // N
Out[•]= 0.555556

In[•]:= jooR = PosR /. {True → 1, False → 0}
Out[•]= {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
         1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

In[•]:= jooH = PosH /. {True → 1, False → 0};
        jooW = PosW /. {True → 1, False → 0};
        jooL = PosL /. {True → 1, False → 0};



        (*Expansion*)

        (** R: 297  *)
```

In[•]:=

```
PermuExpanR = Table[(
    hoz = Tally[Sort[RandomInteger[{1, Length[expan]}, Total[expan]]]];
    guq = ConstantArray[0, Length[expan]];
    hoz1 = #[[1]] → #[[2]] & /@ hoz;
    expanSimul = ReplacePart[guq, hoz1];
    Total[expanSimul jooR]), {10 000}];

SmoothHistogram[PermuExpanR, PlotStyle → Darker[Green], Filling → Axis]
```

Out[•]=



```
(# ≥ 297) & /@ PermuExpanR
```

In[•]:=

```
Count[(# ≤ 297) & /@ PermuExpanR, True]
```

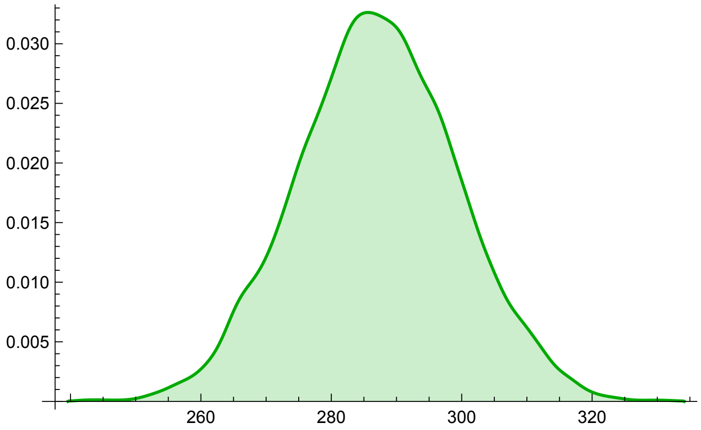
Out[•]=

```
8062
```

```
(** H: 324  *)
```

```
In[•]:= PermuExpanH = Table[(
            hoz = Tally[Sort[RandomInteger[{1, Length[expan]}, Total[expan]]]];
            guq = ConstantArray[0, Length[expan]];
            hoz1 = #[[1]] → #[[2]] & /@ hoz;
            expanSimul = ReplacePart[guq, hoz1];
            Total[expanSimul jooH]), {10 000}];

        SmoothHistogram[PermuExpanH, PlotStyle → Darker[Green], Filling → Axis]
```

Out[•]=



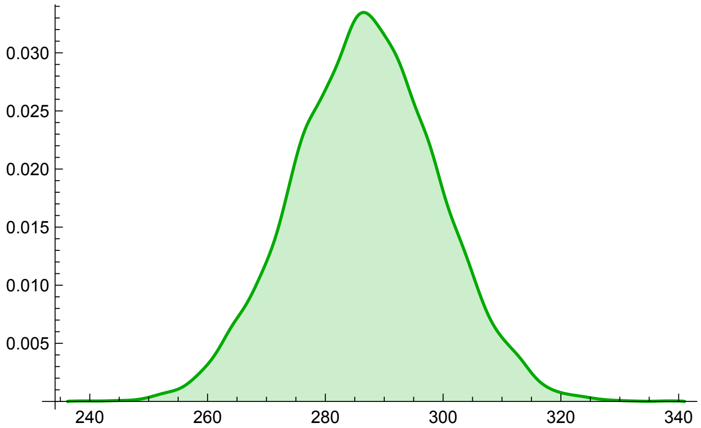
```
In[•]:= N[Count[(# ≥ 324) & /@ PermuExpanH, True] / Length[PermuExpanH]]
Out[•]= 0.0013

In[•]:= Mean[PermuExpanH] // N
Out[•]= 287.011
```

```
(** W: 271  *)
```

```
In[•]:= Floor[N[Mean[PermuExpanW]]]
Out[•]= 286
```

In[•]:=
```
PermuExpanW = Table[(
    hoz = Tally[Sort[RandomInteger[{1, Length[expan]}, Total[expan]]]];
    guq = ConstantArray[0, Length[expan]];
    hoz1 = #[[1]] → #[[2]] & /@ hoz;
    expanSimul = ReplacePart[guq, hoz1];
    Total[expanSimul jooW]), {10 000}];

SmoothHistogram[PermuExpanW, PlotStyle → Darker[Green], Filling → Axis]
```

Out[•]=



In[•]:=
```
N[Count[(# ≤  271) & /@ PermuExpanW, True] / Length[PermuExpanW]]
```

Out[•]=
```
0.1038
```

In[•]:=
```
Mean[PermuExpanW] // N
```

Out[•]=
```
286.861
```

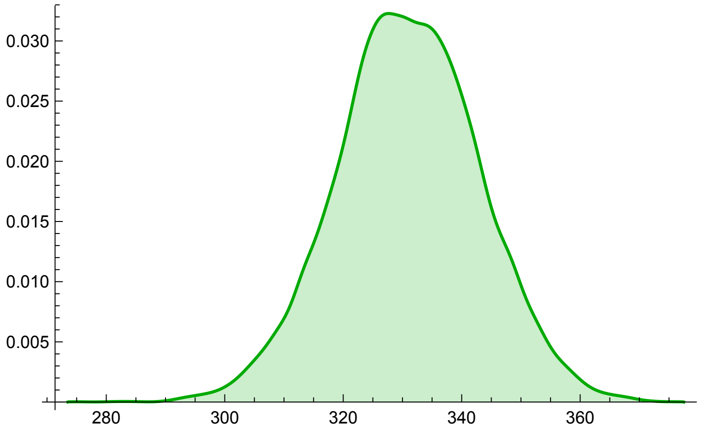
```
(** L: 300  *)
```

In[•]:=
```
Floor[N[Mean[PermuExpanL]]]
```

Out[•]=
```
330
```

```
In[•]:= PermuExpanL = Table[(
            hoz = Tally[Sort[RandomInteger[{1, Length[expan]}, Total[expan]]]];
            guq = ConstantArray[0, Length[expan]];
            hoz1 = #[[1]] → #[[2]] & /@ hoz;
            expanSimul = ReplacePart[guq, hoz1];
            Total[expanSimul jooL]), {10 000}];

        SmoothHistogram[PermuExpanL, PlotStyle → Darker[Green], Filling → Axis]
```

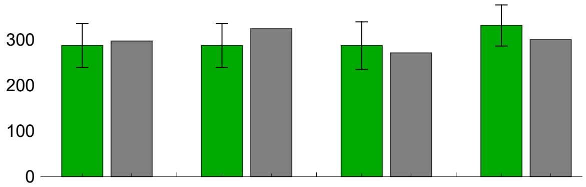
Out[•]=



```
In[•]:= N[Count[(# ≤ 300) & /@ PermuExpanL, True] / Length[PermuExpanL]]

Out[•]= 0.0064
```

```
In[•]:= Spec3F = {{Around[287, 335 - 287], 297}, {Around[287, 239 - 287], 324},
           {Around[287, 339 - 287], 271}, {Around[331, 376 - 331], 300}};

In[•]:= BarChart[Spec3F, ChartStyle → {Darker[Green], Gray}, PlotTheme → "Web",
         BarSpacing → {0.2, 1.2}, ImageSize → 300, AspectRatio → 1 / 3]
```

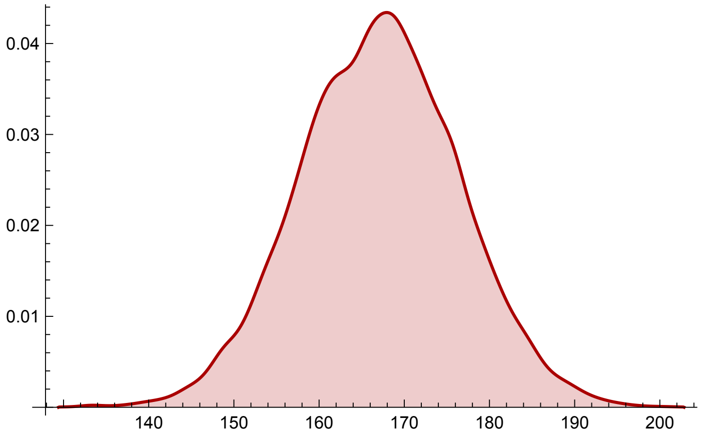
Out[•]=



```
(*Contraction*)
```

```
(** R: 181  *)
```

In[•]:=
```
PermuContracR = Table[(
    hoz = Tally[Sort[RandomInteger[{1, Length[contrac]}, Total[contrac]]]];
    guq = ConstantArray[0, Length[contrac]];
    hoz1 = #[[1]] → #[[2]] & /@ hoz;
    contracSimul = ReplacePart[guq, hoz1];
    Total[contracSimul jooR]), {10 000}];

SmoothHistogram[PermuContracR, PlotStyle → Darker[Red], Filling → Axis]
```

Out[•]=


```
(# ≥  181) & /@ PermuContracR
```

In[•]:=
```
Count[(# ≥  181) & /@ PermuContracR, True]
```

Out[•]=
```
741
```

In[•]:=
```
N[Count[(# ≥  181) & /@ PermuContracR, True] / Length[PermuContracR]]
```

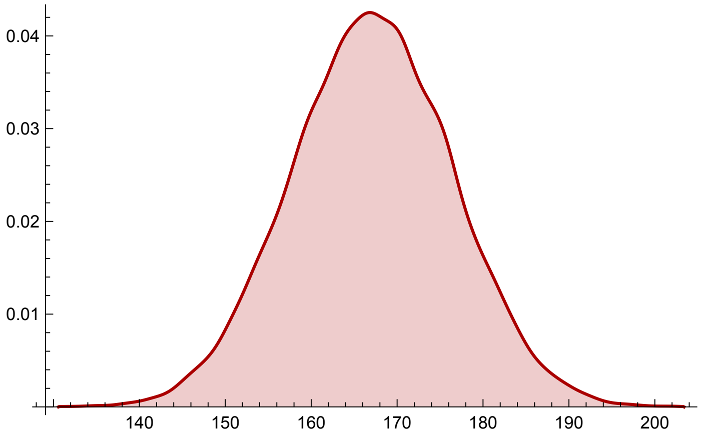
Out[•]=
```
0.0741
```

```
(** H: 96  *)
```

In[•]:=

```
PermuContracH = Table[(
    hoz = Tally[Sort[RandomInteger[{1, Length[contrac]}, Total[contrac]]]];
    guq = ConstantArray[0, Length[contrac]];
    hoz1 = #[[1]] → #[[2]] & /@ hoz;
    contracSimul = ReplacePart[guq, hoz1];
    Total[contracSimul jooH]), {10 000}];

SmoothHistogram[PermuContracH, PlotStyle → Darker[Red], Filling → Axis]
```

Out[•]=



In[•]:=

```
N[Count[(# ≤ 96) & /@ PermuContracH, True] / Length[PermuContracH]]
```

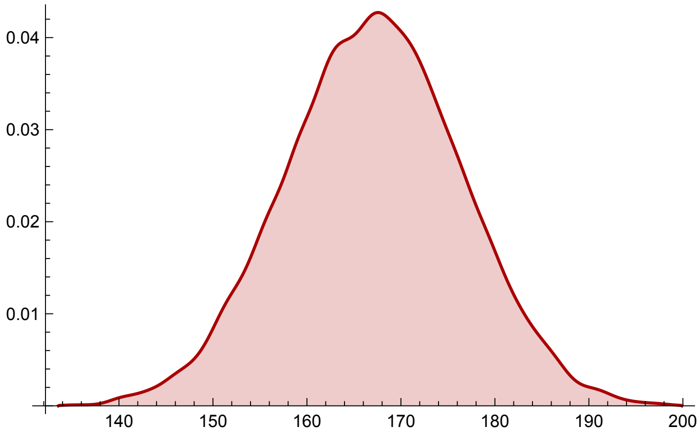
Out[•]=

```
0.
```

```
(** W: 200  *)
```

```
In[•]:= PermuContracW = Table[(
            hoz = Tally[Sort[RandomInteger[{1, Length[contrac]}, Total[contrac]]]];
            guq = ConstantArray[0, Length[contrac]];
            hoz1 = #[[1]] → #[[2]] & /@ hoz;
            contracSimul = ReplacePart[guq, hoz1];
            Total[contracSimul jooW]), {10 000}];

        SmoothHistogram[PermuContracW, PlotStyle → Darker[Red], Filling → Axis]
```

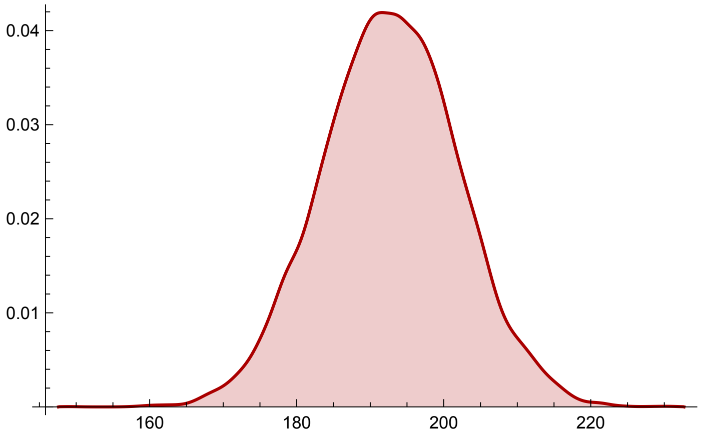
Out[•]=



```
In[•]:= N[Count[(# ≥  200) & /@ PermuContracW, True] / Length[PermuContracW]]
Out[•]= 0.

(** L: 217  *)
```

```
In[•]:= PermuContracL = Table[(
            hoz = Tally[Sort[RandomInteger[{1, Length[contrac]}, Total[contrac]]]];
            guq = ConstantArray[0, Length[contrac]];
            hoz1 = #[[1]] → #[[2]] & /@ hoz;
            contracSimul = ReplacePart[guq, hoz1];
            Total[contracSimul jooL]), {10 000}];

        SmoothHistogram[PermuContracL, PlotStyle → Darker[Red], Filling → Axis]
```

Out[•]=

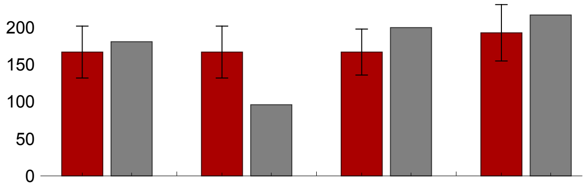


```
In[•]:= N[Count[(# ≥  217) & /@ PermuContracL, True] / Length[PermuContracL]]
Out[•]= 0.0043
```

```
In[•]:= Spec3ConF = {{Around[167, 202 - 167], 181}, {Around[167, 202 - 167], 96},
           {Around[167, 198 - 167], 200}, {Around[193, 231 - 193], 217}};
```

```
In[•]:= BarChart[Spec3ConF, ChartStyle →  {Darker[Red], Gray}, PlotTheme → "Web",
          BarSpacing → {0.2, 1.2}, ImageSize → 300, AspectRatio → 1 / 3]
```

Out[•]=



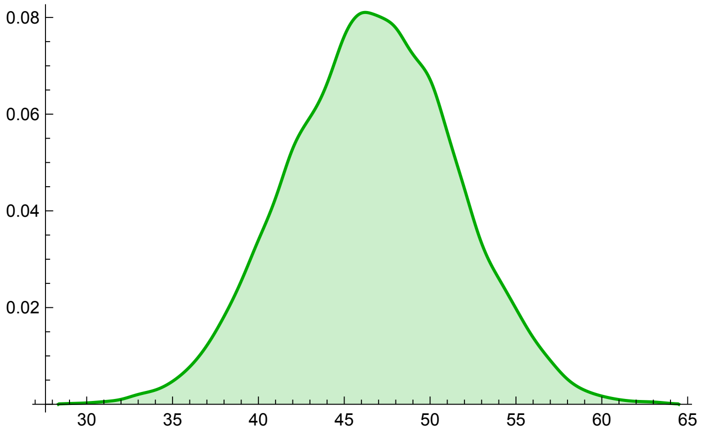
```
(*Double Expansion*)
```

(** R: 45 *)

In[•]:=

```
PermuSuperExpanR = Table[(
    hoz = Tally[Sort[RandomInteger[{1, Length[supexp]}, Total[supexp]]]];
    guq = ConstantArray[0, Length[supexp]];
    hoz1 = #[[1]] → #[[2]] & /@ hoz;
    SuperexpanSimul = ReplacePart[guq, hoz1];
    Total[SuperexpanSimul jooR]), {10 000}];

SmoothHistogram[PermuSuperExpanR, PlotStyle → Darker[Green], Filling → Axis]
```

Out[•]=



```
(# ≥ 45) & /@ PermuSuperExpanR

Count[(# ≥ 45) & /@ PermuSuperExpanR, True]
```

Out[•]=

```
7373
```

In[•]:=

```
N[Count[(# ≥ 45) & /@ PermuSuperExpanR, True] / Length[PermuSuperExpanR]]
N[Count[(# ≤  45) & /@ PermuSuperExpanR, True] / Length[PermuSuperExpanR]]
```

Out[•]=

```
0.6725
```

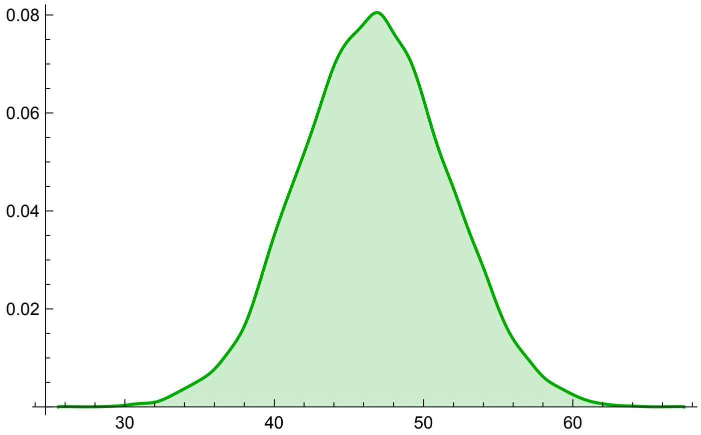
Out[•]=

```
0.4055
```

(** H: 57 *)

```
In[•]:= PermuSuperExpanH = Table[(
            hoz = Tally[Sort[RandomInteger[{1, Length[supexp]}, Total[supexp]]]];
            guq = ConstantArray[0, Length[supexp]];
            hoz1 = #[[1]] → #[[2]] & /@ hoz;
            SuperexpanSimul = ReplacePart[guq, hoz1];
            Total[SuperexpanSimul jooH]), {10 000}];

        SmoothHistogram[PermuSuperExpanH, PlotStyle → Darker[Green], Filling → Axis]
```

Out[•]=

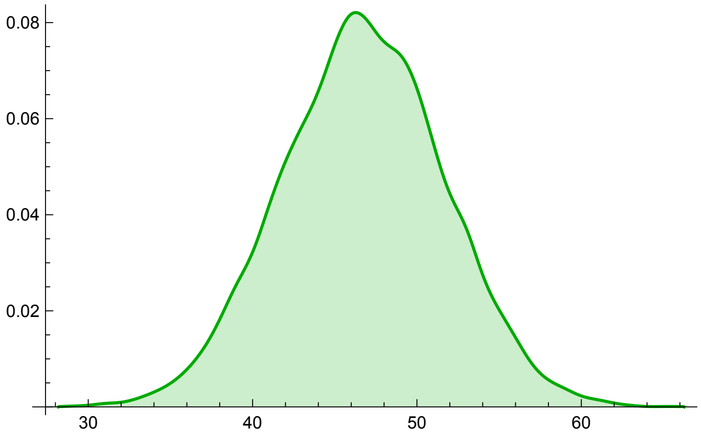


```
In[•]:= N[Count[(# ≥  57) & /@ PermuSuperExpanH, True] / Length[PermuSuperExpanH]]

Out[•]= 0.0243

In[•]:= Mean[PermuSuperExpanH] // N

Out[•]= 46.659
```

```
(** W: 40  *)
```

```
In[•]:= PermuSuperExpanW = Table[(
             hoz = Tally[Sort[RandomInteger[{1, Length[supexp]}, Total[supexp]]]];
             guq = ConstantArray[0, Length[supexp]];
             hoz1 = #[[1]] → #[[2]] & /@ hoz;
             SuperexpanSimul = ReplacePart[guq, hoz1];
             Total[SuperexpanSimul jooW]), {10 000}];

         SmoothHistogram[PermuSuperExpanW, PlotStyle → Darker[Green], Filling → Axis]
```

Out[•]=

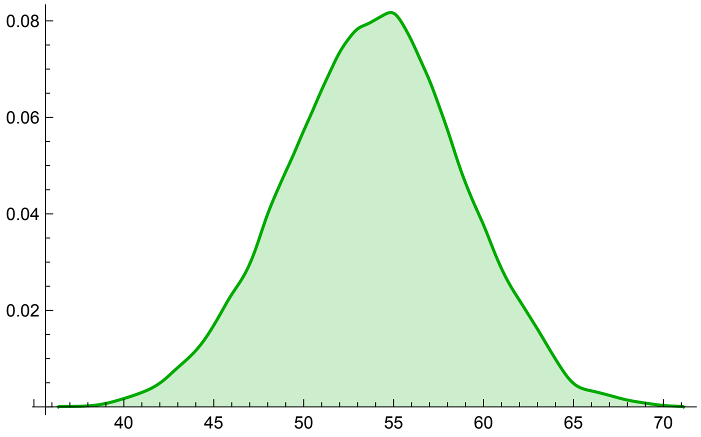


```
In[•]:= N[Count[(# ≤  40) & /@ PermuSuperExpanW, True] / Length[PermuSuperExpanW]]
Out[•]= 0.1049

In[•]:= Mean[PermuSuperExpanW] // N
Out[•]= 46.7115
```

```
(** L: 52  *)
```

```
In[•]:= PermuSuperExpanL = Table[(
            hoz = Tally[Sort[RandomInteger[{1, Length[supexp]}, Total[supexp]]]];
            guq = ConstantArray[0, Length[supexp]];
            hoz1 = #[[1]] → #[[2]] &/@ hoz;
            SuperexpanSimul = ReplacePart[guq, hoz1];
            Total[SuperexpanSimul jooL]), {10 000}];

        SmoothHistogram[PermuSuperExpanL, PlotStyle → Darker[Green], Filling → Axis]
```

Out[•]=



```
In[•]:= N[Count[(# ≤  52) &/@ PermuSuperExpanL, True] / Length[PermuSuperExpanL]]

Out[•]= 0.3843

In[•]:= Mean[PermuSuperExpanL] // N

Out[•]= 53.881

In[•]:= Spec3NEF = {{Around[47, 64 - 47], 45}, {Around[47, 67 - 47], 57},
           {Around[47, 66 - 47], 40}, {Around[54, 71 - 54], 52}};


In[•]:= BarChart[Spec3NEF, ChartStyle →  {Darker[Green], Gray}, PlotTheme → "Web",
         BarSpacing → {0.2, 1.2}, ImageSize → 300, AspectRatio → 1 / 3]
```

Out[•]=