SUP MAT: Evolution of cross-tolerance to metals in yeast

Anna L.Bazzicalupo, Penelope C. Kahn, Eully Ao, Joel Campbell, Sarah P.Otto

Data

Within the analysis for a particular metal (or for all metals combined), identical mutations (same exact SNP) were eliminated, as were mutations in MnBM14 and MnBM42. Rows do not sum to the "all" column because identical mutations observed in different metals were also removed when considering the dataset as a whole ("all"), but not from each metal considered in isolation. To assess parallelism across metals (but not within a metal), "difmetal" counts genes repeatedly hit ONLY in different metals (i.e., mutations in the same gene that occur within a metal are counted only once).

In[•]:=
```
data = {{"#hit", "all", "cd", "co", "cu", "mn", "ni", "zn", "difmetal"},
   {1, 186, 43, 63, 18, 47, 17, 20, 200}, {2, 15, 2, 5, 0, 2, 2, 2, 9},
   {3, 0, 1, 1, 0, 1, 1, 2, 1}, {4, 3, 0, 1, 0, 1, 0, 0, 0}, {5, 3, 0, 0, 0, 0, 0, 1, 0},
   {6, 1, 0, 1, 0, 0, 0, 0, 0}, {7, 0, 0, 0, 0, 0, 0, 0, 0}, {8, 1, 0, 0, 0, 0, 0, 0, 0},
   {9, 0, 0, 0, 0, 0, 0, 0, 0}, {10, 0, 0, 0, 0, 0, 0, 0, 0}, {11, 0, 0, 0, 0, 0, 0, 0, 0},
   {12, 0, 0, 0, 0, 0, 0, 0, 0}, {13, 1, 0, 0, 0, 1, 0, 0, 0}};
MatrixForm[data]
```

Out[•]//MatrixForm=

#hit	all	cd	co	cu	mn	ni	zn	difmetal
1	186	43	63	18	47	17	20	200
2	15	2	5	0	2	2	2	9
3	0	1	1	0	1	1	2	1
4	3	0	1	0	1	0	0	0
5	3	0	0	0	0	0	1	0
6	1	0	1	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0
13	1	0	0	0	1	0	0	0

The number of independent mutations (leading to different SNPs) across all metals:

In[•]:=
```
Drop[data[[All, 2]], 1].Drop[data[[All, 1]], 1]
```

Out[•]=

```
270
```

The number of independent mutations (leading to different SNPs) across all metals, lumping together any mutations in the same gene that occur in the same metal (for testing parallelism across metals):

```
In[•]:= Drop[data[[All, 9]], 1].Drop[data[[All, 1]], 1]
Out[•]=
221
```

Analyses

Cadmium - significant parallelism

```
In[•]:= data[[All, 3]][[1]]
Out[•]=
cd
```

```
In[•]:= obsdata = Drop[data[[All, 3]], 1]
Out[•]=
{43, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```

```
In[•]:= nummut = obsdata.Drop[data[[All, 1]], 1]
Out[•]=
50
```

Of these, the number of multiply hit genes were:

```
In[•]:= nummultiple = Total[Drop[data[[All, 3]], 2]]
Out[•]=
3
```

Randomizing the genes in which the mutations are observed:

```
In[•]:= SeedRandom[129 831]
Out[•]=
RandomGeneratorState[ Method: ExtendedCA
                      State hash: −4 754 988 249 305 394 038 ]
```

Generating random draws of nummut mutations, each of which could occur in any one of 6607 possible genes (with an equal probability of mutating), repeating this 1000 times:

```
In[•]:= tab = Table[1 / 6607, {i, 1, 6607}];
        rantab = Table[BinCounts[
            RandomInteger[MultinomialDistribution[nummut, tab]], {0, 14, 1}], {i, 1, 1000}];
```

The mean number of hits observed (j counts the number of mutations, accounting for genes hit 1, 2, ...13 times):

```
In[•]:= obsdata.Table[j (j), {j, 1, 13}] / obsdata.Table[j, {j, 1, 13}] // N
Out[•]=
1.2
```

Is higher than the mean numbers of all of 1000 randomizations:

```
In[•]:= meantab = Table[rantab[[i]].Table[j^2, {j, 0, 13}] / rantab[[i]].Table[j, {j, 0, 13}], {i, 1, 1000}] // N;
        Max[meantab]
Out[•]=
        1.12
```

The 95% quantile for mean number of genes hit:

```
In[•]:= {Quantile[meantab, 0.025], Quantile[meantab, 0.5], Quantile[meantab, 0.975]}
Out[•]=
        {1., 1., 1.04}
```

Thus the mean number of hits per mutated gene is significant (p<0.001).

Cobalt - significant parallelism

```
In[•]:= data[[All, 4]][[1]]
Out[•]=
        CO
```

```
In[•]:= obsdata = Drop[data[[All, 4]], 1]
Out[•]=
        {63, 5, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0}
```

```
In[•]:= nummut = obsdata.Drop[data[[All, 1]], 1]
Out[•]=
        86
```

Of these, the number of multiply hit genes were:

```
In[•]:= nummultiple = Total[Drop[data[[All, 4]], 2]]
Out[•]=
        8
```

Randomizing the genes in which the mutations are observed:

```
In[•]:= SeedRandom[54 219]
Out[•]=
        RandomGeneratorState[ Method: ExtendedCA
                              State hash: 466 435 363 971 759 394 ]
```

Generating random draws of nummut mutations, each of which could occur in any one of 6607 possible genes (with an equal probability of mutating), repeating this 1000 times:

```
In[•]:= tab = Table[1 / 6607, {i, 1, 6607}];
        rantab = Table[BinCounts[
           RandomInteger[MultinomialDistribution[nummut, tab]], {0, 14, 1}], {i, 1, 1000}];
```

The mean number of hits observed (j counts the number of mutations, accounting for genes hit 1, 2, ...13 times):

In[•]:=
```
obsdata.Table[j (j), {j, 1, 13}]
---------------------------------  // N
obsdata.Table[j, {j, 1, 13}]
```

Out[•]=
```
1.67442
```

Is higher than the mean numbers of all of 1000 randomizations:

In[•]:=
```
meantab = Table[rantab[[i]].Table[j^2, {j, 0, 13}] / rantab[[i]].Table[j, {j, 0, 13}], {i, 1, 1000}] // N;
Max[meantab]
```

Out[•]=
```
1.09302
```

The 95% quantile for mean number of genes hit:

In[•]:=
```
{Quantile[meantab, 0.025], Quantile[meantab, 0.5], Quantile[meantab, 0.975]}
```

Out[•]=
```
{1., 1., 1.04651}
```

Thus the mean number of hits per mutated gene is significant (p<0.001).

Copper - no parallelism

In[•]:=
```
data[[All, 5]][[1]]
```

Out[•]=
```
cu
```

In[•]:=
```
obsdata = Drop[data[[All, 5]], 1]
```

Out[•]=
```
{18, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```

In[•]:=
```
nummut = obsdata.Drop[data[[All, 1]], 1]
```

Out[•]=
```
18
```

Of these, the number of multiply hit genes were:

In[•]:=
```
nummultiple = Total[Drop[data[[All, 5]], 2]]
```

Out[•]=
```
0
```

Manganese - significant parallelism

For manganese, we drop the putative mutators (MnBM14 and MnBM42)

```
In[•]:= data[[All, 6]][[1]]
Out[•]=
mn

In[•]:= obsdata = Drop[data[[All, 6]], 1]
Out[•]=
{47, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1}

In[•]:= nummut = obsdata.Drop[data[[All, 1]], 1]
Out[•]=
71
```

Of these, the number of multiply hit genes were:

```
In[•]:= nummultiple = Total[Drop[data[[All, 6]], 2]]
Out[•]=
5
```

Randomizing the genes in which the mutations are observed:

```
In[•]:= SeedRandom[77 127]
Out[•]=
RandomGeneratorState[ Method: ExtendedCA
                      State hash: 8 628 400 527 046 742 293 ]
```

Generating random draws of nummut mutations, each of which could occur in any one of 6607 possible genes (with an equal probability of mutating), repeating this 1000 times:

```
In[•]:= tab = Table[1 / 6607, {i, 1, 6607}];
rantab = Table[BinCounts[
   RandomInteger[MultinomialDistribution[nummut, tab]], {0, 14, 1}], {i, 1, 1000}];
```

The mean number of hits observed (j counts the number of mutations, accounting for genes hit 1, 2, ...13 times):

```
In[•]:= obsdata.Table[j (j), {j, 1, 13}] / obsdata.Table[j, {j, 1, 13}] // N
Out[•]=
3.50704
```

Is higher than the mean numbers of all of 1000 randomizations:

```
In[•]:= meantab = Table[rantab[[i]].Table[j^2, {j, 0, 13}] / rantab[[i]].Table[j, {j, 0, 13}], {i, 1, 1000}] // N;
Max[meantab]
Out[•]=
1.08451
```

The 95% quantile for mean number of genes hit:

```
In[•]:= {Quantile[meantab, 0.025], Quantile[meantab, 0.5], Quantile[meantab, 0.975]}
Out[•]=
{1., 1., 1.05634}
```

Even if we drop the gene hit 13 times in manganese (*CDC25*), the result is highly significant and outside the range of all 1000 randomizations:

```
In[•]:= Drop[obsdata, -1].Table[j (j), {j, 1, 12}] / Drop[obsdata, -1].Table[j, {j, 1, 12}] // N
Out[•]=
1.37931
```

Thus the mean number of hits per mutated gene is significant (p<0.001).

Nickle - significant parallelism

```
In[•]:= data[[All, 7]][[1]]
Out[•]=
ni

In[•]:= obsdata = Drop[data[[All, 7]], 1]
Out[•]=
{17, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0}

In[•]:= nummut = obsdata.Drop[data[[All, 1]], 1]
Out[•]=
24
```

Of these, the number of multiply hit genes were:

```
In[•]:= nummultiple = Total[Drop[data[[All, 7]], 2]]
Out[•]=
3
```

Randomizing the genes in which the mutations are observed:

```
In[•]:= SeedRandom[32 412]
Out[•]=
RandomGeneratorState[ Method: ExtendedCA
                      State hash: −8 692 798 250 509 348 668 ]
```

Generating random draws of nummut mutations, each of which could occur in any one of 6607 possible genes (with an equal probability of mutating), repeating this 1000 times:

```
In[•]:= tab = Table[1 / 6607, {i, 1, 6607}];
rantab = Table[BinCounts[
   RandomInteger[MultinomialDistribution[nummut, tab]], {0, 14, 1}], {i, 1, 1000}];
```

The mean number of hits observed (j counts the number of mutations, accounting for genes hit 1, 2,

...13 times):

In[•]:=
```
obsdata.Table[j (j), {j, 1, 13}]
-------------------------------- // N
obsdata.Table[j, {j, 1, 13}]
```

Out[•]=
```
1.41667
```

Is higher than the mean numbers of all of 1000 randomizations:

In[•]:=
```
meantab = Table[rantab[[i]].Table[j^2, {j, 0, 13}] / rantab[[i]].Table[j, {j, 0, 13}], {i, 1, 1000}] // N;
Max[meantab]
```

Out[•]=
```
1.08333
```

The 95% quantile for mean number of genes hit:

In[•]:=
```
{Quantile[meantab, 0.025], Quantile[meantab, 0.5], Quantile[meantab, 0.975]}
```

Out[•]=
```
{1., 1., 1.08333}
```

Thus the mean number of hits per mutated gene is significant (p<0.001).

Zinc - significant parallelism

In[•]:=
```
data[[All, 8]][[1]]
```

Out[•]=
```
zn
```

In[•]:=
```
obsdata = Drop[data[[All, 8]], 1]
```

Out[•]=
```
{20, 2, 2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0}
```

In[•]:=
```
nummut = obsdata.Drop[data[[All, 1]], 1]
```

Out[•]=
```
35
```

Of these, the number of multiply hit genes were:

In[•]:=
```
nummultiple = Total[Drop[data[[All, 8]], 2]]
```

Out[•]=
```
5
```

Randomizing the genes in which the mutations are observed:

In[•]:=
```
SeedRandom[82 712]
```

Out[•]=
```
RandomGeneratorState[ Method: ExtendedCA
                      State hash: −1 951 316 623 039 499 653 ]
```

Generating random draws of nummut mutations, each of which could occur in any one of 6607 possible genes (with an equal probability of mutating), repeating this 1000 times:

In[•]:=
```
tab = Table[1 / 6607, {i, 1, 6607}];
rantab = Table[BinCounts[
   RandomInteger[MultinomialDistribution[nummut, tab]], {0, 14, 1}], {i, 1, 1000}];
```

The mean number of hits observed (j counts the number of mutations, accounting for genes hit 1, 2, ...13 times):

In[•]:=
```
obsdata.Table[j (j), {j, 1, 13}] / obsdata.Table[j, {j, 1, 13}] // N
```

Out[•]=
```
2.02857
```

Is higher than the mean numbers of all of 1000 randomizations:

In[•]:=
```
meantab = Table[rantab[[i]].Table[j^2, {j, 0, 13}] / rantab[[i]].Table[j, {j, 0, 13}], {i, 1, 1000}] // N;
Max[meantab]
```

Out[•]=
```
1.11429
```

The 95% quantile for mean number of genes hit:

In[•]:=
```
{Quantile[meantab, 0.025], Quantile[meantab, 0.5], Quantile[meantab, 0.975]}
```

Out[•]=
```
{1., 1., 1.05714}
```

Thus the mean number of hits per mutated gene is significant (p<0.001).

All metals together - significant parallelism

In[•]:=
```
data[[All, 2]][[1]]
```

Out[•]=
```
all
```

In[•]:=
```
obsdata = Drop[data[[All, 2]], 1]
```

Out[•]=
```
{186, 15, 0, 3, 3, 1, 0, 1, 0, 0, 0, 0, 1}
```

In[•]:=
```
nummut = obsdata.Drop[data[[All, 1]], 1]
```

Out[•]=
```
270
```

Of these, the number of multiply hit genes were:

```
In[•]:= nummultiple = Total[Drop[data[[All, 2]], 2]]
Out[•]=
24
```

Randomizing the genes in which the mutations are observed:

```
In[•]:= SeedRandom[2120]
Out[•]=
RandomGeneratorState[ Method: ExtendedCA
                      State hash: 1510253275902383272 ]
```

Generating random draws of nummut mutations, each of which could occur in any one of 6607 possible genes (with an equal probability of mutating), repeating this 1000 times:

```
In[•]:= tab = Table[1 / 6607, {i, 1, 6607}];
        rantab = Table[BinCounts[
            RandomInteger[MultinomialDistribution[nummut, tab]], {0, 14, 1}], {i, 1, 1000}];
```

The mean number of hits observed (j counts the number of mutations, accounting for genes hit 1, 2, ...13 times):

```
In[•]:= obsdata.Table[j (j), {j, 1, 13}] / obsdata.Table[j, {j, 1, 13}] // N
Out[•]=
2.36296
```

Is higher than the mean numbers of all of 1000 randomizations:

```
In[•]:= meantab = Table[rantab[[i]].Table[j^2, {j, 0, 13}] / rantab[[i]].Table[j, {j, 0, 13}], {i, 1, 1000}] // N;
        Max[meantab]
Out[•]=
1.11111
```

The 95% quantile for mean number of genes hit:

```
In[•]:= {Quantile[meantab, 0.025], Quantile[meantab, 0.5], Quantile[meantab, 0.975]}
Out[•]=
{1.00741, 1.03704, 1.07407}
```

Even if we drop the gene hit 13 times in manganese (*CDC25*), the result is highly significant and outside the range of all 1000 randomizations:

```
In[•]:= Drop[obsdata, -1].Table[j (j), {j, 1, 12}] / Drop[obsdata, -1].Table[j, {j, 1, 12}] // N
Out[•]=
1.8249
```

Double hits: The 95% quantile for expected number of double hit genes is 1-10 (median of 5), whereas

15 were observed:

```
In[•]:= {Quantile[rantab[[All, 3]], 0.025],
 Quantile[rantab[[All, 3]], 0.5], Quantile[rantab[[All, 3]], 0.975]}
Out[•]=
{1, 5, 10}
```

More than two hits: The 95% quantile for expected number of triple-plus hit genes is 0-1 (median of 0), whereas 9 were observed:

```
In[•]:= {Quantile[Sum[rantab[[All, i]], {i, 4, 10}], 0.025],
 Quantile[Sum[rantab[[All, i]], {i, 4, 10}], 0.5],
 Quantile[Sum[rantab[[All, i]], {i, 4, 10}], 0.975]}
Out[•]=
{0, 0, 1}
```

Number of hits per gene and count of the maximum time that # of hits was observed:

```
In[•]:= Table[{i - 1, Max[rantab[[All, i]]]}, {i, 1, 10}]
Out[•]=
{{0, 6351}, {1, 270}, {2, 12}, {3, 2}, {4, 1}, {5, 0}, {6, 0}, {7, 0}, {8, 0}, {9, 0}}
```

Only 6.7% of simulations had any genes hit more than twice, whereas 9 were observed:

```
In[•]:= Total[Sum[rantab[[All, i]], {i, 4, 10}]] / 1000.
Out[•]=
0.067
```

Thus both the mean number of hits per gene (the main test) is significant (p<0.001), as is the number of genes hit twice or more than twice.

All metals together (dropping repeated hits in the same gene) - significant parallelism

```
In[•]:= data[[All, 9]][1]
Out[•]=
difmetal

In[•]:= obsdata = Drop[data[[All, 9]], 1]
Out[•]=
{200, 9, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

In[•]:= nummut = obsdata.Drop[data[[All, 1]], 1]
Out[•]=
221
```

Of these, the number of multiply hit genes were:

```
In[•]:= nummultiple = Total[Drop[data[[All, 9]], 2]]
Out[•]=
10
```

Randomizing the genes in which the mutations are observed:

```
In[•]:= SeedRandom[8329]
Out[•]=
RandomGeneratorState[ Method: ExtendedCA
                      State hash: −4 294 397 200 794 781 781 ]
```

Generating random draws of nummut mutations, each of which could occur in any one of 6607 possible genes (with an equal probability of mutating), repeating this 1000 times:

```
In[•]:= tab = Table[1 / 6607, {i, 1, 6607}];
        rantab = Table[BinCounts[
            RandomInteger[MultinomialDistribution[nummut, tab]], {0, 14, 1}], {i, 1, 1000}];
```

The mean number of hits observed (j counts the number of mutations, accounting for genes hit 1, 2, ...13 times):

```
In[•]:= obsdata.Table[j (j), {j, 1, 13}] / obsdata.Table[j, {j, 1, 13}] // N
Out[•]=
1.1086
```

Is above the highest of all 1000 randomizations:

```
In[•]:= meantab = Table[rantab[[i]].Table[j^2, {j, 0, 13}] / rantab[[i]].Table[j, {j, 0, 13}], {i, 1, 1000}] // N;
        Max[meantab]
Out[•]=
1.0905
```

The 95% quantile for mean number of genes hit:

```
In[•]:= {Quantile[meantab, 0.025], Quantile[meantab, 0.5], Quantile[meantab, 0.975]}
Out[•]=
{1., 1.02715, 1.0724}
```

Double hits: The 95% quantile for expected number of double hit genes is 0-7 (median of 3) is less than the 9 observed:

```
In[•]:= {Quantile[rantab[[All, 3]], 0.025],
         Quantile[rantab[[All, 3]], 0.5], Quantile[rantab[[All, 3]], 0.975]}
Out[•]=
{0, 3, 7}
```

More than two hits: The 95% quantile for expected number of triple-plus hit genes is 0-1 (median of

0), whereas 1 was observed:

```
In[•]:= {Quantile[Sum[rantab⟦All, i⟧, {i, 4, 10}], 0.025],
         Quantile[Sum[rantab⟦All, i⟧, {i, 4, 10}], 0.5],
         Quantile[Sum[rantab⟦All, i⟧, {i, 4, 10}], 0.975]}
Out[•]=
{0, 0, 1}
```

Number of hits per gene and count of the maximum time that # of hits was observed:

```
In[•]:= Table[{i - 1, Max[rantab⟦All, i⟧]}, {i, 1, 10}]
Out[•]=
{{0, 6396}, {1, 221}, {2, 10}, {3, 1}, {4, 0}, {5, 0}, {6, 0}, {7, 0}, {8, 0}, {9, 0}}
```

Only 3.3% of simulations had any genes hit more than twice, whereas one was observed:

```
In[•]:= Total[Sum[rantab⟦All, i⟧, {i, 4, 10}]] / 1000.
Out[•]=
0.033
```

Thus both the mean number of hits per gene (the main test) is significant (p<0.001), as is the number of genes hit more than twice (PMA1)