**COMMUNITY SEEDING PROTOCOL**

We have 121 isolates from flower communities. We may want to discard some of them if we find that the 16S of two or more species is very similar (to avoid confusion if we want to sequence later on).

To seed the communities, it is easiest to work with symmetries. We can use 96-well plates but leave the first and last two columns blank as to effectively have a symmetric 8x8 plate. Thus, if we want to start off with two 8x8 plates of isolates, we need 64x2=128 isolates. We can use lab isolates to complete the set of 121 flower ones (minus those we discard).

Example of plate layout:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *isolates\_plate\_1* | | | | | | | | | | | |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| A |  |  |  |  |  |  |  |  |  |  |  |  |
| B |  |  |  |  |  |  |  |  |  |  |  |  |
| C |  |  |  |  |  |  |  |  |  |  |  |  |
| D |  |  |  |  |  |  |  |  |  |  |  |  |
| E |  |  |  |  |  |  |  |  |  |  |  |  |
| F |  |  |  |  |  |  |  |  |  |  |  |  |
| G |  |  |  |  |  |  |  |  |  |  |  |  |
| H |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
|  | Blank well |
|  | Isolates |

We want our protocol to yield results that are as similar as possible to what a true random sampling of species would produce. Suppose that we want to seed 64 communities (one 8x8 plate) with 8 species each (if we want to use only plate rotations, it is not possible to increase the number of species without repetitions as we will see later).

**True random sampling**

We have that:

* Our species pool size is Ns = 128
* We want to seed 64 communities (Nc = 64)
* We want each community to have ns = 8 species

The probability p(k) of a species appearing in k communities is:

The probability ppair(k) of a species pair appearing together in k communities is:

We want our sampling process to be as true as possible to these probabilities to ensure that:

1. no species is over- or under-represented in the seeded communities
2. there is no bias as to which combinations of species are or are not present in the seeded communities

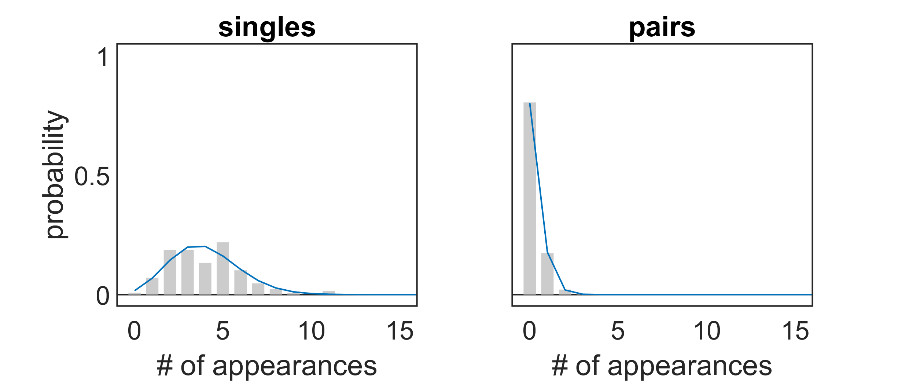


Fig. 1. True random seeding. Simulation of 64 communities seeded by randomly picking 8 isolates out of a pool of size 128 each time. Left: isolate frequency of appearance. Right: pairs frequency of appearance. Blue lines correspond to the analytical p(k) and pparis(k).

**Protocol 1: only rotations**

If we want to seed the communities by just rotating and combining the two plates of isolates (isolates\_plate\_1 and isolates\_plate\_2), the maximum number of species per community we can achieve without species repetitions is 8. This can be seen by looking at the plate corners. We have 2 plates of isolates, that is 8 corners. Rotations allow for corners to mix with one another, but never with a species that is not in a corner. Therefore, once every corner has been mixed with every other corner, there are no further combinations that can be done without repetitions of species. This happens when all 8 species in the corners have been mixed.

Example of such a protocol:

pairs\_plate = isolates\_plate\_1 + isolates\_plate\_2

quartets\_plate = pairs\_plate + pairs\_plate.rotate\_left()

octets\_plate = quartets\_plate + quartets\_plate.rotate\_left().rotate\_left()

There are many other ways to generate the final octets\_plate, but they all lead to the same final result. E.g., this protocol leads to the same final octets\_plate:

pairs\_plate\_1 = isolates\_plate\_1 + isolates\_plate\_2

pairs\_plate\_2 = isolates\_plate\_1 + isolates\_plate\_2.rotate\_left().rotate\_left()

quartets\_plate = pairs\_plate\_1 + pairs\_plate\_2.rotate\_right()

octets\_plate = quartets\_plate + quartets\_plate.rotate\_left().rotate\_left()

But involves a higher number of plates and operations (rotation/mixing).

As shown in figure 2, this protocol leads to a set of 64 communities with 8 species each, such that every species appears in exactly 4 communities. This is arguably better than a true random sampling, as it negates the risk of species over- or under-representation. As for species pairs, most pairs are not represented in the community set (as it happens when doing true random sampling), but a few pairs are heavily over-represented.

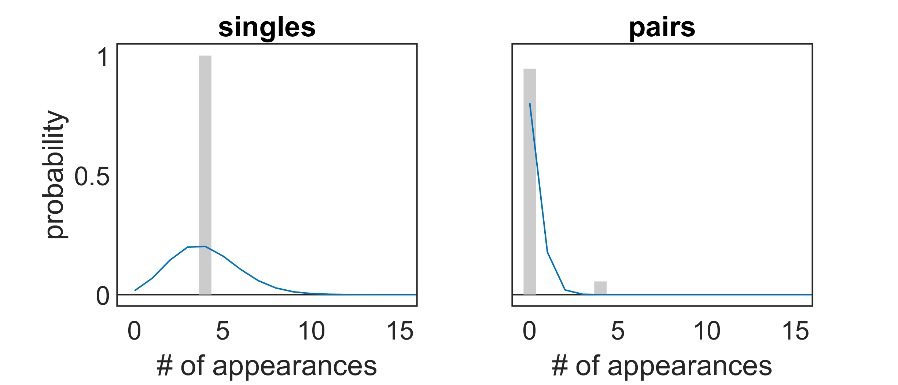


Fig.2. Protocol 1. Same plots as in figure 1, but this time communities were assembled according to protocol 1.

This protocol leads to repeated communities being assembled. Even though there are no species repetitions (i.e. the same species being added to any given community more than once), some communities are indeed repeated (i.e. contain the same species). Specifically, each community is repeated 4 times due to the symmetries. This is what leads to the over-representation of some species pairs (figure 2, right panel). Figure 3 shows the pattern of repeated communities within the final plate.

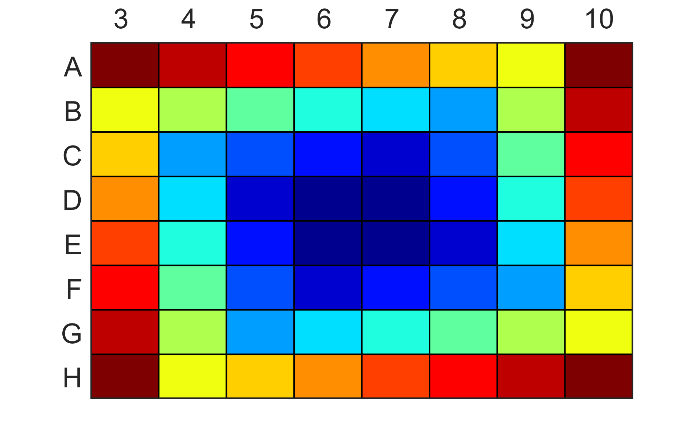


Fig.3. Plate layout resulting from protocol 1. Each color represents a different community, note the repeated communities.

**Protocol 2: shuffling rows + rotations**

These issues can be bypassed if we incorporate more complex, symmetry-breaking operations to our protocol. For example, we could “shuffle” the rows of our plates. From here on, we will call “shuffling” to the operation of rearranging the rows in a plate such that:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *original plate* | | | | | | | | | | | |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| A |  |  | row 1 | | | | | | | |  |  |
| B |  |  | row 2 | | | | | | | |  |  |
| C |  |  | row 3 | | | | | | | |  |  |
| D |  |  | row 4 | | | | | | | |  |  |
| E |  |  | row 5 | | | | | | | |  |  |
| F |  |  | row 6 | | | | | | | |  |  |
| G |  |  | row 7 | | | | | | | |  |  |
| H |  |  | row 8 | | | | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *shuffled plate* | | | | | | | | | | | |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| A |  |  | row 5 | | | | | | | |  |  |
| B |  |  | row 6 | | | | | | | |  |  |
| C |  |  | row 7 | | | | | | | |  |  |
| D |  |  | row 8 | | | | | | | |  |  |
| E |  |  | row 1 | | | | | | | |  |  |
| F |  |  | row 2 | | | | | | | |  |  |
| G |  |  | row 3 | | | | | | | |  |  |
| H |  |  | row 4 | | | | | | | |  |  |

With this operation, we can design the following protocol:

pairs\_plate\_1 = isolates\_plate\_1 + isolates\_plate\_1.shuffle()

pairs\_plate\_2 = isolates\_plate\_2 + isolates\_plate\_2.shuffle()

pairs\_plate\_3 = isolates\_plate\_1 + isolates\_plate\_2

pairs\_plate\_4 = isolates\_plate\_1.shuffle() + isolates\_plate\_2.shuffle()

quartets\_plate\_1 = pairs\_plate\_1 + pairs\_plate\_2.rotate\_left()

quartets\_plate\_2 = pairs\_plate\_3 + pairs\_plate\_4.rotate\_left()

octets\_plate = quartets\_plate\_1 + quartets\_plate\_2.rotate\_left().rotate\_left()

This protocol produces 64 unique communities with no species repetitions and is reasonably close to a true random sampling in terms of the frequencies of appearance of species pairs (figure 4).

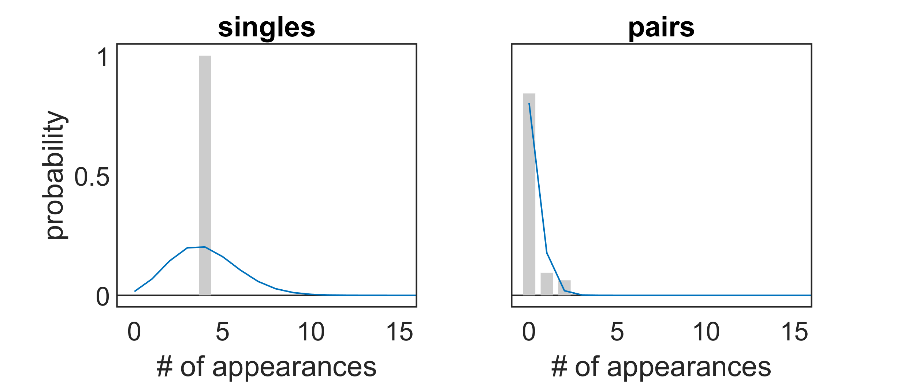


Fig.4. Protocol 2. Same plots as in figure 1, but this time communities were assembled according to protocol 2.

**Protocol 3: shuffle and rotate, simplified**

This is just a simplified version of protocol 2 involving less plates and steps.

pairs\_plate\_1 = isolates\_plate\_1 + isolates\_plate\_1.shuffle()

pairs\_plate\_2 = isolates\_plate\_2 + isolates\_plate\_2.shuffle()

quartets\_plate = pairs\_plate\_1 + pairs\_plate\_2.rotate\_left()

octets\_plate = quartets\_plate + quartets\_plate.rotate\_left()

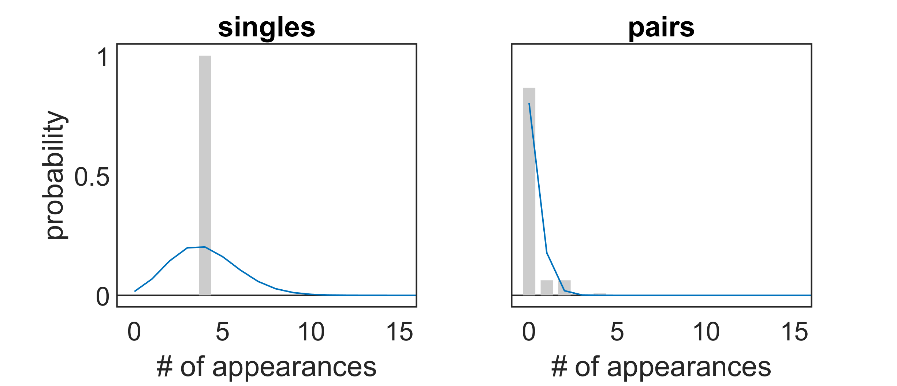


Fig.5. Protocol 3. Same plots as in figure 1, but this time communities were assembled according to protocol 3.

**Protocol 4: best so far**

This protocol involves rotations and then a slightly more complex shuffling step, but produces 64 unique communities without repetitions in species sampling and with pair frequencies that closely resemble those of a true random sampling (figure 6).

pairs\_plate\_1 = isolates\_plate\_1 + isolates\_plate\_2

pairs\_plate\_2 = isolates\_plate\_1.rotate\_left() + isolates\_plate\_2.rotate\_right()

quartets\_plate = pairs\_plate\_1 + pairs\_plate\_2

octets\_plate = quartets\_plate + quartets\_plate.uneven\_shuffle()

Here, the “uneven\_shuffle” operation means rearranging rows such that:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *original plate* | | | | | | | | | | | |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| A |  |  | row 1 | | | | | | | |  |  |
| B |  |  | row 2 | | | | | | | |  |  |
| C |  |  | row 3 | | | | | | | |  |  |
| D |  |  | row 4 | | | | | | | |  |  |
| E |  |  | row 5 | | | | | | | |  |  |
| F |  |  | row 6 | | | | | | | |  |  |
| G |  |  | row 7 | | | | | | | |  |  |
| H |  |  | row 8 | | | | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *shuffled plate* | | | | | | | | | | | |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| A |  |  | row 3 | | | | | | | |  |  |
| B |  |  | row 4 | | | | | | | |  |  |
| C |  |  | row 5 | | | | | | | |  |  |
| D |  |  | row 6 | | | | | | | |  |  |
| E |  |  | row 7 | | | | | | | |  |  |
| F |  |  | row 8 | | | | | | | |  |  |
| G |  |  | row 1 | | | | | | | |  |  |
| H |  |  | row 2 | | | | | | | |  |  |

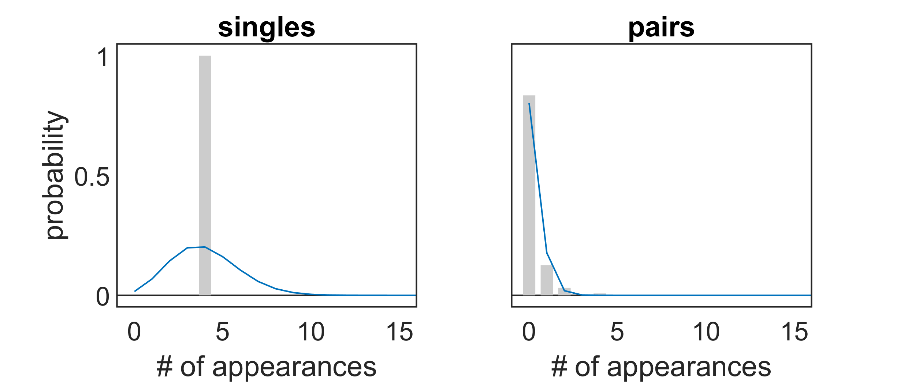


Fig.6. Protocol 4. Same plots as in figure 1, but this time communities were assembled according to protocol 4.