Chapter 2

Alan R. Vazquez

Excercises

```
import numpy as np
```

Many of the exercises in this book use the Python `oapackage` (Eendebak and Vazquez 2019). You may need to install the package first by

```
# pip install oapackage
```

At the start of most of the programs, you need to import the package with `import oapackage`. We like to abbreviate lengthy names using

```
# import oapackage as oap
```

So whenever you see `oap` in the book or in the programs, you now know where it refers to.

Once installed and imported, you can use all kinds of convenient functions for the characterization and enumeration of the arrays. A few additional functions are provided in `oafunctions.py`. We use `import oafunctions as oaf` in our programs.

1. The function `strength` in `oafunctions.py` takes an array as input and returns the strength of the array. Have a look at this function and explain how the function works. After that, answer the following questions:

- What would a strength of 1 imply? What a strength of 0?
- Does a strength-4 array also have strength 3?
- Can you detect a strength greater than 4 with the program?

2. In many exercises, you need to read files with orthogonal arrays to do something with them. The files all have a special structure that permit easy reading by the function `oap.readarrayfile`. In particular, this is how we arranged all the files:

- First line: 3 numbers specifying c:# columns of a single array, r: # rows of a single array and a:# orthogonal arrays in the file.
- Line 2: ID of the first array.
- Line 3 up to $(2 + r)$: the first array.
- ⋮
- Line $2 + a(r + 1) - r$ up to $1 + a(r + 1)$: array #a.
- Last line: -1.

We assume that the symbols in any column of an array are coded $0, 1, \ldots, (s - 1)$, where s is the number of different symbols in that column.

The file "result-12.2-2-2-2-2-2.oa" contains two different OA(12, 2^6, 2). Check the file's structure by opening it in an editor. You read the file with

```
#N12n6 = oap.readarrayfile(`result-12.2-2-2-2-2-2.oa')
```

The command returns a tuple called `N12n6` with two so-called `array_link` objects. Don't worry! They don't bite. We will gently introduce them later on. For now, turn, say, the second of them into a `numpy` array with

```
# A = np.array(N12n6[1])
```

and determine the strength using the function of the previous exercise.

3. Consider pure-level arrays with level numbers $s = 2, 3, 4$.

- Make a table with the smallest possible run sizes for arrays with strength $2 \leq t \leq 4$.
- What is the smallest possible run size for an OA$(N, 4 \times 3 \times 2^4, 3)$?

4. Construct the following OAs by hand or show that the array does not exist.

- $OA(16, 4^2 2^1; 2)$.
- $OA(16, 4^2 2^2; 3)$.
- $OA(16, 4^1 2^3; 2)$.
- $OA(16, 4^1 2^3; 3)$.
- $OA(18, 3^2 2^2; 2)$.

5. The research and development team of an important company is investigating the biological decolorization of Malachite Green, a hazardous industrial dye. The goal of this project is to optimize the conditions for effective removal of color from dye-bearing wastewater using a biodegradation system. Previous research has shown that the process of color removal by biodegradation can be complex and influenced by various operational parameters. To better understand the interactions between these factors and their impact on decolorization efficiency, the laboratory aims to investigate three critical variables that

are described as follows. The first variable is the temperature at 5, 15, 25, and 35 ^oC. The second variable is dye concentration at 2.5, 5, 10, 12.5 $mgl^{-}1$. The third variable is the algae type: Chlamydomonas, Euglena, Cosmarium, and Chlorella. All the experimental tests will use a pH of 6. Design the experiment using an orthogonal array with 16 runs. *Answer*: $OA(16, 4^3; 2)$

6. The microbiology laboratory at a large hospital is facing a challenging outbreak of Pseudomonas aeruginosa (P. aeruginosa) infections among its patients. To better understand the spread and control the infection, the lab wants to genotype this bacterium using the AP^PCR (Amplified Fragment Length Polymorphism) typing technique. The AP^PCR technique is a molecular biology method that amplifies specific DNA fragments to detect variations in the bacterial genome. However, to achieve reliable and accurate results, it's essential to optimize the reaction conditions. After consulting with experts in the field, the lab has identified four critical factors that might affect the AP^PCR typing performance. The first factor is **Mg2þ concentration**, which is the amount of magnesium ions in the reaction mixture. The second factor is the **nucleotide concentration**, which is the amount of nucleotides available for amplification. The third factor is the **primers (CagA2+CMVin2) concentration**, meaning the amount of primer molecules used to target specific DNA sequences. The last factor is the **DNA template concentration**. Specifically, it is the initial amount of P. aeruginosa DNA in the reaction mixture. Each of the four factors will be studied at three different levels. Design the experiment using an orthogonal array with 9 runs. *Answer*: $OA(9, 3^4; 2)$

7. A group of scientists is developing an Enzyme-Linked Immunosorbent Assay (ELISA) system for detecting phages expressing specific antibodies. The goal of this project is to optimize the ELISA protocol to achieve high sensitivity and specificity in phage detection. To date, significant variation exists in the ELISA protocols used across different laboratories, making it essential to tailor the method to the specific application. In this case, the laboratory aims to develop an optimized ELISA system for detecting phages expressing a ScFv monoclonal antibody against adenovirus hexon protein. The research team has identified four critical factors that may impact the performance of the ELISA system. The first factor is the Phage concentration in coat buffer, which will be studied at 6300, 630, 157, 63, 6.3, and 0.063 ng/ml. The second factor is the dilution of anti-phage polyclonal antibody at 1/5000, 1/10000, and 1/20000 mol/L. The third factor is the dilution of anti-rabbit biotinylated antibody at 1/5000, 1/10000, and 1/20000 mol/L. The fourth factor is the development time of the OPD substrate, which will be tested at 5, 8, and 12 min. Design the experiment using an orthogonal array with 18 runs. *Answer*: $OA(18, 6^1 3^3; 2)$

8. Foldover of a three-level design. Show that you start from strength two and you arrive at a strength three array.

9. Consider two OAs with eight runs and four factors in Table 1 and Table 2.

Table 1: An orthogonal array with 8 runs and four two-level factors.

0	1	1	0
1	0	1	0
0	0	0	0
1	1	0	0
0	1	0	1
1	0	0	1
0	0	1	1
1	1	1	1

Table 2: Another orthogonal array with 8 runs and four two-level factors.

0	1	1	0
1	0	1	0
0	0	0	1
1	1	0	1
0	1	0	0
1	0	0	0
0	0	1	1
1	1	1	1

- Use vertical splitting in each column of Table 1 to produce a pair of arrays with three factors and four runs. Are all these pairs of arrays OAs of strength 2?
- Similarly, apply vertical splitting to each column of Table 2 to produce smaller arrays. Are all these pairs of arrays OAs of strength 2?
- Explain why is that the different OAs in Table 1 and Table 2 result in better and worse OAs.

10. Exercise on collapsing in Section 2.3.
11. This problem illustrates another type of collapsing.

- Consider the $OA(16, 4^4; 2)$ in Table 3. Check that the OA has a strength of 2.
- Collapse the first two four-level columns into two two-level columns as follows: $0, 1 \to 0$ and $2, 3 \to 1$. Check that the resulting array is a $OA(16, 4^2 2^2; 2)$
- Collapse the other two four-level columns into two-level columns using the same collapsing as before. Check that the resulting array is now an $OA(16, 2^4; 2)$.

Table 3: An orthogonal array with 16 runs and 4 four-level factors.

2	2	2	2
1	1	1	2
0	2	2	1
3	1	1	1
2	0	2	0
1	3	1	0
0	0	2	3
3	3	1	3
2	2	0	2
1	1	3	2
0	2	0	1
3	1	3	1
2	0	0	0
1	3	3	0
0	0	0	3
3	3	3	3

The OA just created is known as an strong orthogonal array. We refer to He and Tang (2013) for more details on these OAs.

References

Eendebak, Pieter, and Alan Vazquez. 2019. "OApackage: A Python Package for Generation and Analysis of Orthogonal Arrays, Optimal Designs and Conference Designs." *Journal of Open Source Software* 4: 1097. https://doi.org/10.21105/joss.01097.

He, Yuanzhen, and Boxin Tang. 2013. "Strong Orthogonal Arrays and Associated Latin Hypercubes for Computer Experiments." *Biometrika* 100 (1): 254–60.