23257
Compositional Data Analysis with Applications in Genomics

Lecture notes

Spring 2023

Christian Brinch
DTU Food

Genomic Epidemiology

Fourth edition, Spring 2023

Contents

1. Introduction to compositional data

Compositional data are data where the elements are parts of a whole. Whenever a fraction is measured, whether it is as a percentage, a concentration, ppm, or a frequency, it is part of a composition. Probabilities are also compositional, making compositional data analysis particularly useful in the field of statistics. Compositional data are always positive, and the sum of a composition is a constant, which is usually, but not necessarily, 1. The sum is often irrelevant, and only the *relative* magnitude of the parts has meaning.

Compositional data are encountered in many aspects of daily life and are dealt with implicitly. Political surveys, election polls, nutrition declarations on packaged food, and many types of probabilistic games are all examples of everyday compositional data. They are also commonly encountered in most sciences, in particular chemistry, geology, and life science.

In this chapter, we will learn how to identify compositional data and how to distinguish between compositional data and ordinary real multivariate data. Compositional data need to be analyzed using compositional methods, and we will show how failure to do so can lead to meaningless or even paradoxical results.

1.1 Identifying compositional data

A geologist studies meteorites, and she wants to compare two meteorites that she has found. The smaller of the two contains 100 g of FeNi alloys, which amounts to 95% of the total weight of the meteorite. The remaining 5% are silicates. The larger meteorite contains 400 g of iron, which is 23% of the total weight, with the remaining 77% consisting of silicates. Despite the fact that the second meteorite contains more iron (by weight), she correctly identifies it as a chondrite and the smaller one as an iron meteorite. The larger meteorite contains more iron because it is bigger; therefore, the absolute amount of iron has no significance. The only thing that matters is the amount of iron relative to the amount of silicates. She reports the content as percentages because the weight depends on the size

of the meteorites and is therefore irrelevant.

In another example, a meteorologist (who does not study meteoritics) is out to compare the weather at two different locations. He measures the temperature, pressure, and wind speeds. If he measures a higher temperature at one location, he will correctly conclude that it is warmer there than at the other location, regardless of the measured pressure and wind speeds. These quantities can in principle take any positive value, and they are not constrained by one another; therefore, the sum of these numbers is meaningless. Meteorological data are not compositional, but rather what is called real multivariate data.

1.2 Relevance for metagenomics

When studying genetic samples containing DNA pooled from a (large) number of organisms, it is called metagenomics as opposed to ordinary WGS genomics, which is done using cultivated clonal isolates from a single organism. The practical process of extracting, preparing, and sequencing the DNA is beyond the scope of this course, but in the end, ideally, metagenomic high-throughput sequencing produces compositional data by identifying which organism all the DNA fragments belong to and counting the number of fragments associated with a certain organism. Effectively, this is not always the case, and the outputted data is pseudo-compositional for reasons discussed later. In this course, we will discuss more aspects of compositional data analysis than what is relevant for metagenomics, but emphasis will be placed on the techniques that apply to genomic data.

1.2.1 Genomic data

Genomic data are generally a set of short nucleotide sequences (reads), which are packed into a so-called *FASTQ*-file. At a first glance, it can be difficult to see how such data are compositional. It is crucial to understand what the reads represent in order to accept their compositional nature. A bottle of sea water contains cells from hundreds or maybe thousands of different organisms: a lot of bacteria, possibly parasites and viruses, plant matter and algae, fungi, archaea, and maybe remnants of fish or even mammals. There are billions of cells in the sample, each containing a strand of DNA, but only a random fraction of these cells will be opened during the process of DNA extraction. The process may be biased towards bacterial cells or eukaryotic cells, depending on the kit used, but in the end, it is a random process. The DNA gets fragmented and loaded onto the flow cell in the sequencing machine, where it is read and decoded. Again, only a random subset of all the fragments gets sequenced. In the end, we may end up with 127 reads belonging to a certain species. Does this number mean anything? Absolutely not. It is the same as walking to a highway bridge and, for five minutes, counting all the white cars driving underneath. Maybe it is 85. Is that a lot? That depends on how many non-white cars passed during the same five minutes. Only by counting some other part – blue cars, all cars, or motor bikes – can we conclude anything about the observation of 85 white cars. It is the proportion that matters, not the absolute number. Therefore, the reads in the *FASTQ*-files are parts of a composition, and they can be grouped (amalgamated) in various ways using bioinformatic techniques such as mapping, assembly, and binning. But it is important to remember that the only meaningful quantity is the ratio between parts and not the absolute magnitude of each part, which is based on arbitrary properties such as sample size, sample storage

conditions, DNA extraction efficacy, sequencing depth, etc.

1.3 Proportions

We are taught to work with proportions already in primary school, and to most people, dealing with (simple) compositional data is intuitive. However, as the following quotation shows, not everybody has an intuitive comprehension of proportional data and the mathematical rules that apply to it. On November 16, 1999, a member of the Danish parliament, Folketinget, Aase D. Madsen, spoke the following during a debate about usage of public libraries:

> "[...] Og med hensyn til, hvem der kommer på bibliotekerne, og hvem der ikke kommer, er der en tabel 18 med en gruppe delt ind efter alder, og dér står, at 39 pct. af den mandlige del af befolkningen aldrig kommer på bibliotekerne, og at 30 pct. af den kvindelige del af befolkningen, altså fordelt gennemsnitligt over alder, aldrig kommer der. Og når jeg lægger mænd og kvinder sammen - det skal man være lidt forsigtig med, men på det her område tør jeg godt - så giver 39 pct. af mændene og 30 pct. af kvinderne i befolkningen tilsammen, og det må være 69 pct. Tager jeg fejl?"

It is easy to laugh about this, but once the compositions become bigger and more complex, mathematical errors are easily made, even by people who find the quotation above amusing.

Proportional data behave differently from real data, which is illustrated in the following three examples.

1.3.1 Negative proportions

A proportion is the ratio between two real and positive numbers, while a fraction is the ratio between two real but not necessarily positive numbers. It is important to understand the difference. Fractions can be negative, while proportions are strictly positive. You cannot ask someone to cut minus one quarter of a pizza. The following example shows what can happen when applying normal statistics to proportional data.

Before the elections, eight opinion polls are conducted in order to predict the success of a certain political party. The resulting 8 percentages of people who claim they will vote for the party are {2%,2%,2%,3%,3%,4%,11%,21%}. Calculating the mean and the standard deviation, assuming a normal distribution, gives $6\% \pm 6.3$, suggesting that there is a 17% probability that the party will receive a negative proportion of the votes, which is clearly nonsense.

A similar problem arises when food producers want to test if their product is free of a certain ingredient. This could, for instance be a brewery making alcohol-free beer or a dairy plant making lactose-free milk. These products are rarely 100% free of the ingredient they claim to be free of, but contain trace amounts, which is allowed as long as no single item contains more than a fixed (and small) proportion. For beer, this is typically around 0.03% alcohol. In order to comply with these rules, companies sample their product and measure how much alcohol or lactose it contains. However, because the proportions are so small, the ordinary standard deviation will typically extend into the range of negative numbers, making their statistics invalid if they do not take the compositionality of the data into account.

1.3.2 Small proportions

Oftentimes, small proportions are important but hard to quantify accurately in absolute terms. Cooking recipes are usually not given as proportions but rather as absolute amounts (weight or volume). However, in many cases, the exact amount of salt is not given explicitly, because the ratio of salt to water in, say, a soup, is very small, and even a small measuring error in the amount of salt could render the soup inedible. If a dish requires 1 g of salt, and by accident you add 2 g, then you have doubled the amount of salt and spoiled the dish even if you only added a single gram too much. A small absolute change in a small proportion can lead to a large proportional change.

1.3.3 Proportional changes

Reporting changes as a proportion can lead to false impressions of the significance of the change. If, on a given day, a stock on the market has a value of \$100 per stock, and on the following day its value has increased to \$220, we could report the change as a 120% increase. On the third day, the value decreases by 70%, which at a glance appears to be a smaller decrease than the increase the day before. However, the value of the stock on day 3 will be \$66 which is much less than the \$100 that it started out with.

1.4 Simpson's paradox

A pitfall often encountered when working with compositional data is the amalgamation paradox, or Simpson's paradox, named after mathematician Edward Simpson, who first described the effect in 1951. The effect is illustrated in the table 1.1, where the success rates of two treatments for kidney stones, A and B, are compared. The data shows that the success rate of treatment B is higher when all cases are considered, but when the cases are split into two parts, small and large kidney stones, treatment A works better in both situations.

	Treatment A	**Treament B**
Small kidney stone	(81/87) 93%	(234/270) 87%
Large kidney stone	(192/263) 73%	(55/80) 69%
Total	(273/350) 78%	(289/350) 83%

Table 1.1: The success rates of two different treatments for kidney stone, when calculated from the total number of cases and when the cases are split into parts.

The problem lies in the fact that there are many more cases of large kidney stones being treated with method A and small kidney stones being treated with method B than vice versa, and the total represents a weighted arithmetic average of the cases, which is a property that is ill-behaved when applied to proportions.

1.5 Correlations

A question that often arises in metagenomics is whether certain organisms within a number of samples are correlated in some manner. For instance, in a study of gut microbiomes, it

could be relevant to look for bacterial species that are in high abundance whenever another (set of) species is abundant. Correlations in compositional data are, however, not as easily interpretable as they are for real data, and very often, if correlations show up, they are artifacts and not real.

1.5.1 Spurious correlations

Consider three randomly distributed variables: x, y, and z. By construction, these are all uncorrelated, but it turns out that the ratios x/z and y/z can be highly correlated, giving the false impression that there is a relationship between x and y. This situation is commonly encountered in metagenomics, where gene counts are often "normalized" by organism counts. If, for example, x and y represent the counts of two different bacterial genes across a number of samples and z represents the total number of bacteria, then it is common practice to express the gene abundance as x/z and y/z, a ratio sometimes referred to as *FPKM*, in which case it may seem like the occurrence of gene 1 and gene 2 is highly correlated even if that is not the case at all. An example is shown in Fig. 1.1.

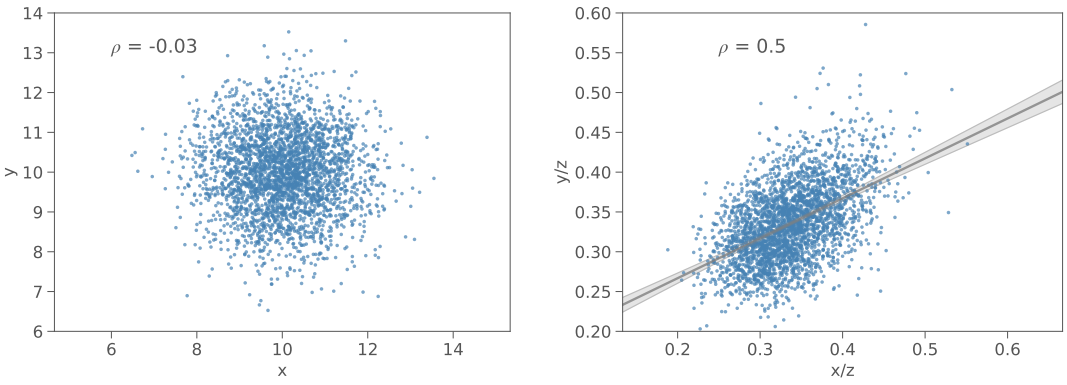

Figure 1.1: Two random and normal distributed variables x and y are shown in the left panel to be uncorrelated. By dividing both with a third random variable z, the ratios are seen to be positively correlated in the right panel.

The spurious correlation occurs because the two ratios share a common denominator, and since compositional data are, by construction, made up of proportions, spurious correlations are widespread in genomics.

1.5.2 The negative correlation bias

Apart from spurious correlations, compositional data suffer from a so-called negative correlation bias. Consider the composition of results from N coin tosses. If the number of heads is n, the number of tails must be $N-n$. The more heads you get, the fewer tails you will get. Heads and tails will always be perfectly negatively correlated. This effect extends to larger compositions: if a set of parts goes up, another set of parts must go down, and this negative correlation is fundamentally indistinguishable from true negative correlations.

1.5.3 Compositional correlations

If two variables, x and y, are correlated, we expect that a linear relationship exists between them, such that $y = \alpha x + \beta$. For real data, α and β are irrelevant. The strength of the

correlation is determined by how closely the data points follow the relation. This is seen in the left panel of Fig. 1.2, where three sets of variables show equally good correlation.

For compositional data, in order for parts to be correlated, the ratio between the parts must be constant. For the red and the black data sets in Fig. 1.2, the ratios between the variables are constant and equal to 1 and 5, respectively. In the blue data set, however, the ratio is not constant, but equal to $y/x = 1 + 20/x$. If we plot the three data sets in log-log space, as is seen in Fig. 1.2 panel b, it is clear that there is a linear relationship within the red and the black data sets, but not within the blue. Thus for parts in compositional data to be correlated, they must follow a linear relationship in log-space of the form $y = x + \beta$, where the intercept β in log-space equals the slope α in Euclidean space (and is irrelevant for the strength of the correlation).

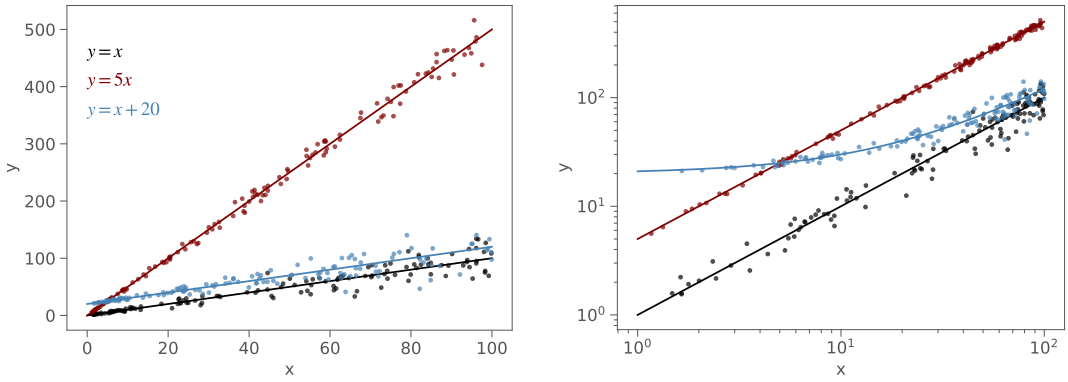

Figure 1.2: Parts in a set of composition correlates if they have a constant ratio, in which case they will have a linear relationship in log-space.

1.6 A brief history of compositional data analysis

The phenomenon of spurious correlations was the main motivation for the development of compositional analysis. Karl Pearson was the first to point out the problems of spurious correlations when applying standard statistical methods to proportions, back in 1897. Pearson and other prominent statisticians of that time was worried that conclusions would be drawn from correlations that are artifacts of the analysis method, rather than actual relationships between variables. His warning, however, was largely ignored until 1960. Around this time, Felix Chayes, who was a geologist, noted that standard multivariate analysis should not be applied to compositional data, and scientists began to move away from multivariate correlations within the field of geology. Only in the 1980s did John Aitchison, a Scottish statistician, establish the modern methods of compositional data analysis and noted

> It seems surprising that the warnings of three such eminent statistician-scientists as Pearson, Galton and Weldon should have largely gone unheeded for so long: even today uncritical applications of inappropriate statistical methods to compositional data with consequent dubious inferences are regularly reported.

Although this quote is almost 40 years old, "normalized" count data and correlations between them are still widely reported in scientific publications today, particularly in the fields of bio- and life-science.

2. Basic mathematical concepts

2.1 Defining compositions

In this chapter, we will define compositions as mathematical objects. We will start with some formal definitions.

Definition 2.1.1 — Composition. A composition is defined by a vector of positive, non-zero values, which only carry relative information. The composition is said to contain D parts if the length of the vector equals D:

$$\mathbf{x} = (x_1, x_2, ..., x_D), \qquad x_i \in \mathbb{R}_+ \tag{2.1}$$

Relative information means that each individual part of the composition carries no information on its own. If we did a poll among students on campus to see whether they would like to have lectures on Saturdays and you were only told a single part of the result, let's say 25 students agreed, this would not provide any information on the outcome of the poll. You would have to know the number of students who answered 'no', in order to extract any information. If the number of students who say 'no' is 17, only then would you know that a majority of students would like to have lectures on Saturdays. The number of yaysayers only carries information relative to the number of naysayers.

In the above example, we asked 42 students about their opinions. If we had asked twice as many, 84, and 50 had said 'yes' and 34 had said 'no', the result would have been exactly the same. Multiplying two numbers by a constant does not change the ratio between them. Thus, we can define compositions as equivalence classes.

Definition 2.1.2 — Compositional equivalence. Two compositions $\mathbf{x}, \mathbf{y}$ are compositionally equivalent if a positive, real constant λ exists, so that $\mathbf{x} = \lambda \cdot \mathbf{y}$.

The total number of students that were asked is irrelevant; only the ratio of 'yes' to 'no'

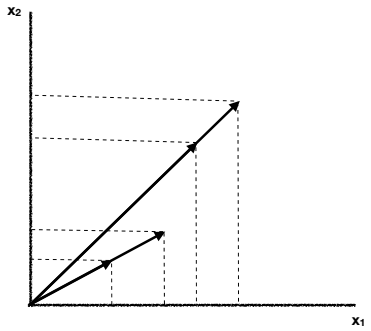

Figure 2.1: Compositional equivalence means that vectors that point in the same direction are equivalent. The length of compositional vectors is irrelevant.

answers matters. However, if we wish to compare several compositions, it is sometimes useful to rescale the compositions to the same total sum, for instance 1 (for proportions) or 100 (for percentages). Rescaling corresponds to changing the unit of the parts. When we rescale a composition to a certain constant, κ, we say that we close the composition to κ.

Definition 2.1.3 — Closure. The closure of a composition $\mathbf{x}$ to a positive, real number κ is defined as

$$\mathscr{C}(\mathbf{x}) = \frac{\kappa}{\sum_{i=1}^{D} x_i} \cdot \mathbf{x} \tag{2.2}$$

As an example of closure, let us consider the situation where, on two consecutive days, we ask a number of students whether they would favor lectures on Saturdays. The first day, we ask 42 students, and their replies are $\mathbf{x} = (25, 17)$. The second day, we ask 33 student, and their answer is $\mathbf{y} = (19, 14)$. Does the answer differ on day two? It is difficult to judge by sight, because the total number of respondents differs. However, we can close both compositions to 100 using definition 2.1.3, so that $\mathbf{x} = (60, 40)$ and $\mathbf{y} = (58, 42)$. Now the compositions are given as percentages, and it is immediately clear that the answers have shifted by two percentage points on day 2. Notice that even if it seems that the parts carry absolute information after closure, this is not the case. If you were told that 40% answered 'no', this does not mean that 60% said 'yes'. It could be that 30% said 'yes' and 30% said 'don't know', in which case 'no' would be in the majority. The parts still only carry relative information.

It should be clear already that compositions are different from ordinary real vectors. They can only be positive, and their lengths are determined by an arbitrary closure constant. We can therefore now understand the mathematical reason why the politician talking about library usage in the last chapter was wrong. She was dealing with two compositions, with parts describing non-users and users of libraries, $\mathbf{m} = (39, 61)$ and $\mathbf{f} = (30, 70)$. Adding them would give the composition $(69, 131)$, which is not closed to 100 and therefore does not have the unit 'percent'. If we close the sum back to 100, we get $(34.5, 65.5)$, which

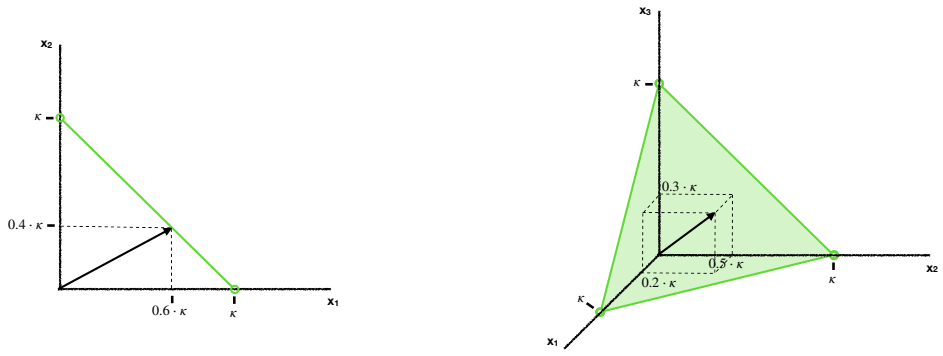

Figure 2.2: 1- and 2-simplices (green) as the sampling space for a 2- and a 3-part composition.

really is the average and not the sum. Furthermore, in order for 'addition' to be a valid operation, we need a 'neutral element' and 'inverse elements', defined such that the sum of an element and the neutral element gives the element itself, and the sum of an element and its inverse element gives the neutral element. For real numbers, the neutral element is 0 and the inverse elements are the negative numbers, i.e., $a+0=a$ and $a+(-a)=0$. From the definition of compositions, we can see that we have neither the inverse elements nor the neutral element (only positive numbers are allowed), so adding compositions together is ill-defined.

With these definitions, we can define the space in which all compositions exist, the so-called sample space.

Definition 2.1.4 — Sample space. The sample space, i.e., the space containing all possible compositions, is the *simplex*, defined as

$$\mathscr{S}^D = \left\{ \mathbf{x} = (x_1, x_2, ...x_D) \,\middle|\, x_i \in \mathbb{R}_+, i = 1,2,...,D; \sum_{i=1}^{D} x_i = \kappa \right\} \tag{2.3}$$

The simplex is a generalization of a triangle to arbitrary dimensions. A triangle is a 2-simplex defined by its three vertices. The simplex that defines the sample space of a particular composition has a dimensionality that is always 1 lower than the number of parts. This is because the last part is constrained by the closure relation, so that only D-1 parts are independent. Figure 2.2 show the 1- and 2-simplices. The 3-simplex is a tetrahedron, and the 4-simplex is known as a 5-cell (and so on for higher dimensions), which can only be viewed in projection.

In many circumstances, we are only interested in some parts of a composition, either because the remaining parts are irrelevant or impossible to obtain. In the example from above with the student poll, we might get a certain number of 'yes' answers, another number of 'no' answers, and in addition, a few who answer 'don't know'. We are not interested in the 'don't know's, so we chose to consider only the 'yes' and the 'no' answers. This is called a subcomposition.

Definition 2.1.5 — Subcomposition. Given a composition $\mathbf{x}$ and a set of indices $S = i_1, i_2, ...i_s$, a subcomposition is obtained by applying the closure operation to the subvector $\mathbf{x}_s$.

If our poll shows $\mathbf{x} = (60, 30, 30)$ for 'yes'/'no'/'don't know', we can form the subcomposition $\mathbf{x}_{yes/no} = (66.67, 33.33)$ by closing the subvector to 100. Most compositions are already subcompositions. For instance, the 'yes'/'no'/'don't know'-composition is already a subcomposition of the composition 'yes'/'no'/'don't know'/'don't care' which again can be seen as a subcomposition of a composition that contains even more possible answers.

Another way to reduce the dimensionality of a composition is to amalgamate parts by summing them into a new part.

Definition 2.1.6 — Amalgamation. Given a composition $\mathbf{x}$ of D parts and a set of indices $A = i_1, i_2, ..., i_a$ and another set $\bar{A} = D_i \setminus A$, the composition

$$\mathbf{x}' = (\mathbf{x}_{\bar{A}}, x_A), \qquad x_A = \sum_{i \in A} x_i \tag{2.4}$$

is called the amalgamated composition in $\mathscr{S}^{D-a+1}$.

An example of an amalgamated composition is when we take the poll from above and include 5 'don't care'-responses, so that we get $\mathbf{x} = (60, 30, 30, 5)$ for 'yes'/'no'/'don't know'/'don't care'. We can then amalgamate the last two parts into one new part, which we could call 'Other', and we would get $\mathbf{x}_{yes/no/other} = (48, 24, 28)$, by summing the parts we want to amalgamate and applying closure. The original 4-part compositions is defined on $\mathscr{S}^4$, whereas after amalgamating two parts, the new composition is defined on $\mathscr{S}^{4-2+1} = \mathscr{S}^3$

Subcompositions and amalgamations are often encountered in metagenomic data analysis. When the read sequences are mapped or aligned against reference databases, it is expected that a certain number of reads do not map, either because the databases are incomplete or because the reads originate from DNA that belongs to an organism which we are not interested in, e.g., mammal DNA if we only map against microorganisms. These unassigned reads will form a part of themselves in the composition, but they are, in most cases, irrelevant for the downstream analysis. We could then choose to look at the subcomposition that is everything but unassigned reads.

An example of amalgamation in metagenomics is when we look at reads that map to antimicrobial resistance gene references. Many different genes give bacteria resistance to the same class of antimicrobial agent, and it is sometimes more useful to look at the composition where these genes have been amalgamated so that, rather than having a composition where the parts represent individual genes, the parts represent phenotypical resistance classes.

2.2 Principles of compositional data analysis

Three principles should be respected by any mathematical method applied to compositional data. The first two principles, scale and permutation invariance, are simple and easily accepted as necessary truths, whereas the third principle, subcompositional coherence, requires a bit more insight to understand.

2.2.1 Scale invariance

Scale invariance follows from the fact that only ratios between parts of a composition matter. It shouldn't matter if we conduct our student poll among 50, 117, or 500 students (if we disregard counting noise and uncertainties); the result should be the same. A function f that operates on a composition $\mathbf{x}$ is called scale-invariant if $f(\lambda\mathbf{x}) = f(\mathbf{x})$ for any positive real value $\lambda \in \mathbb{R}_+$. The function should give the same result for all compositionally equivalent vectors. Many mathematical functions obey this criterion, but some are a more practical choice than others. One example of such a function is $f(\mathbf{x}) = x_1/x_2$ since $x_1/x_2 = (\lambda x_1)/(\lambda x_2)$. It is clear that the constant λ cancels and the function is scale invariant. This function corresponds to changing the unit, for example, from percent to ppm, and is essentially the function we use for closure. The downside of ratios is that they are strictly positive and that they depend on the ordering of the parts, since $x_1/x_2 \neq x_2/x_1$. One can get around this conveniently by taking the logarithm of the ratio, $f(\mathbf{x}) = \ln(x_z/x_2)$. This transformation is symmetric with respect to the (arbitrary) ordering of the parts and maps to the entire set of real numbers.

One can define more complex logratios, for instance the so-called logcontrast,

$$f(\mathbf{x}) = \sum_{i=1}^{D} \alpha_i \ln(x_i), \qquad \sum_{i=1}^{D} \alpha_i = 0. \tag{2.5}$$

This is also a scale-invariant function and can be used to determine equilibrium conditions in, for instance, chemical reactions and thermodynamics. Combustion of methane, for instance, can be described by the balanced equation,

$$1\mathrm{CH}_4 + 2\mathrm{O}_2 \rightarrow 1\mathrm{CO}_2 + 2\mathrm{H}_2\mathrm{O},$$

where the stoichiometric coefficients are $1+2$ on the left and $1+2$ on the right, so that their total sum is 0. If $\mathbf{x}$ is a 4-part composition, describing the concentration of methane, oxygen, carbon dioxide, and water, we can write the reaction as a logcontrast,

$$1\ln(x_1) + 2\ln(x_2) - 1\ln(x_3) - 2\ln(x_4) = \ln\left(\frac{x_1 \cdot x_2^2}{x_3 \cdot x_4^2}\right)$$

When the reaction is in equilibrium, the logcontrast will stay constant, no matter the concentration of the parts.

2.2.2 Permutation invariance

Permutation invariance means that the order of parts in a composition does not influence the result of the analysis. Obviously, if two compositions are to be compared, the order of the parts needs to be the same in both, but if the order is changed in both, the result will be the same. The function above, $f(\mathbf{x}) = x_1/x_2$, which provides scale invariance, does not provide permutation invariance, since $x_1/x_2 \neq x_2/x_1$. By taking the logarithm of the ratio, inverting the ratio (rearranging the parts) only produces a sign change and thus gives symmetry to f with respect to permutation. We can square the log-ratios to obtain perfect permutational invariance.

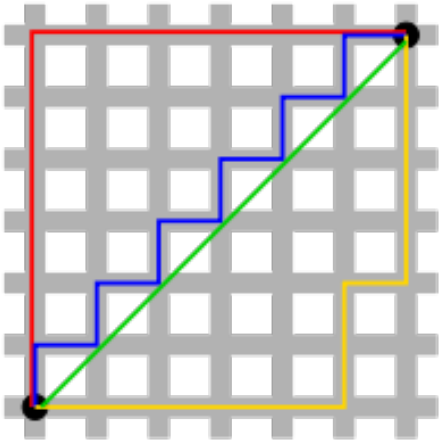

Figure 2.3: Comparison between the Euclidean metric and the taxicab metric on the plane. In the taxicab metric, the red, blue, and yellow routes all give the shortest distance, whereas in the Euclidean metric, only a single distance exists (green).

2.2.3 Subcompositional coherence

The equivalent of a subcomposition in real analysis is an orthogonal projection. We know that the length of the projection of a real vector onto a subspace is always shorter or, at most, equal to the length of the full vector. In compositional analysis, the length of vectors is meaningless, but a similar principle should apply. This is known as *subcompositional coherence*. Subcompositional coherence (sometimes also referred to as subcompositional dominance) means that the distance between two arbitrary subcompositions should always be smaller or, at most, equal to the distance between the full compositions. Also, the principle of scale invariance must be preserved within an arbitrary subcomposition. The distance between two compositions depends on the choice of metric, also known as the distance function. While different metrics exist, the best known is probably the Euclidean metric, which gives the "straight-line" distance between two points in Euclidean space. The general definition of the Euclidean metric gives the distance between two points $\mathbf{p}$ and $\mathbf{q}$ as $d(\mathbf{p},\mathbf{q}) = \sqrt{(\sum_{i=1}^{n}(q_i - p_i)^2}$, which reduces to the famous Pythagorean theorem, $c^2 = a^2 + b^2$ for the two points $\mathbf{p} = (a,0)$ and $\mathbf{q} = (0,b)$ in $\mathbb{R}^2$.

The Euclidean metric, when applied to compositional data, does not obey the principle of subcompositional coherence (see Exercise 2.5), and we therefore can not directly base any analysis of compositional data on this metric. In the field of ecology, the Euclidean and Manhattan (taxicab) metrics (shown in figure 2.3) are often used when comparing samples, but neither provides compositional coherence. Even worse, the Bray-Curtis dissimilarity,

$$BC_{ij} = 1 - \frac{2C_{ij}}{S_i S_j}$$

is often used to provide a distance measure between samples. While it is certainly valid to apply Bray-Curtis to compositional data, it is not a metric since it does not satisfy the triangle inequality. It is a dissimilarity and not a distance and should not be used as such, for instance, in machine learning applications.

The proper metric, which provides compositional coherence, is called the Aitchison metric, and we will introduce this in the next chapter.

2.3 Exercises

Exercise 2.1 During the COVID-19 pandemic, SSI (Statens Serum Institut) would publish the results of the national test effort on a daily basis. They provided, among other things, the number of positive tests. They also provided the number of tests conducted, from which one could calculate the positive percentage. There has been some discussion, particularly in the media, about which of these two numbers is the appropriate one to report. When is it appropriate to use the number of positive tests, and when is it appropriate to use the positive percentage? Why? (Discuss in class.) ■

Exercise 2.2 When applying closure to a composition, the choice of κ depends on the unit of the data. What are the values of κ when data is measured as percentages? As parts per million (ppm)? What unit is metagenomic data measured in, and what is the corresponding κ? ■

Exercise 2.3 Consider this table of faux data:

	1	2	3	4	5
x_1	79.07	31.74	18.61	49.51	29.22
x_2	12.83	56.69	72.05	15.11	52.36
x_3	8.10	11.57	9.34	35.38	18.42

Verify that the data could be treated as compositional. ■

Exercise 2.4 Form a two-part amalgamated composition, $(x_1, x_2 + x_3)$ of the data from Exercise 2.3. Does amalgamation preserve closure? ■

Exercise 2.5 Compute the Euclidean distance between the first two vectors in the data table from Exercise 2.3. Imagine that originally a fourth variable, x_4, was measured, constant for all samples, and equal to 5%. Take the first three vectors, close them to 95%, add the fourth variable (so that they sum up to 100%), and compute the Euclidean distance between the vectors. Is the distance greater or smaller than the distance between the three-part compositions? What about the Manhattan distance ($d_{pq} = \sum_i |p_i - q_i|$)? ■

3. Log-ratio transformation

In most aspects of metagenomic analysis, we are interested in comparing the content of two or more samples, either to determine how identical or how different they are, or to explore if certain variables associated with the samples can explain the variance in the set of samples. In order to make this comparison on a quantitative level, we need to define a distance between two samples.

3.1 Linear algebra

Before we can define the distance between two compositions, we need to define the simplex as a normed vector space. Let us first recall how real vector spaces in $\mathbb{R}^D$ are defined. In any real space, that is for non-compositional data, distances are given by the Euclidean metric,

Definition 3.1.1 — Euclidean distance.

$$d_e(\mathbf{x},\mathbf{y}) = ||\mathbf{x}-\mathbf{y}||_e = \sqrt{\sum_{i=1}^{D}(q_i - p_i)^2}, \qquad \mathbf{x},\mathbf{y} \in \mathbb{R}^D, \tag{3.1}$$

where subscript e denotes the Euclidean distance. The metric is induced by the norm,

Definition 3.1.2 — Euclidean Norm.

$$||\mathbf{x}||_e = \sqrt{\mathbf{x}\cdot\mathbf{x}} \qquad \mathbf{x} \in \mathbb{R}^D, \tag{3.2}$$

which, for real vectors, is the length of a vector $\mathbf{x}$, where the dot product is defined as,

Definition 3.1.3 — Dot product (Euclidean inner product).

$$\mathbf{x} \cdot \mathbf{y} = \langle \mathbf{x}, \mathbf{y} \rangle_e = \sum_{i=1}^{D} x_i y_i, \qquad \mathbf{x}, \mathbf{y} \in \mathbb{R}^D. \tag{3.3}$$

These formulas are familiar to anyone who has studied ordinary linear algebra, but they are not appropriate when working with compositional data, as illustrated by the following example.

■ **Example 3.1** Consider the four compositions

$$\begin{bmatrix} 5 \\ 65 \\ 30 \end{bmatrix} \begin{bmatrix} 10 \\ 60 \\ 30 \end{bmatrix} \quad \begin{bmatrix} 50 \\ 20 \\ 30 \end{bmatrix} \begin{bmatrix} 55 \\ 15 \\ 30 \end{bmatrix}$$

Using definition 3.1.1, we can determine the Euclidean distance between the first two and the last two compositions. In both cases, $d_e \approx 7.07$, meaning that from a Euclidean geometric point of view, the first two and the last two compositions are equally far apart. However, if we look at the proportions, as is appropriate for compositions, the first part doubles between the first two compositions, whereas for the last two compositions, it only increases by 10%. The second part decreases by 7.7% between the two first compositions, while it decreases by 30% between the second two. From a compositional point of view, the latter two are much closer than the former two. ■

A vector space is defined as a set V that is closed under the operations of addition and scalar multiplication. 'Closed under' means that the addition of two elements of V should yield a new element that is also part of V and for scalar multiplication, that $\lambda \mathbf{x} \in V$ for any $\mathbf{x} \in V$ and $\lambda \in \mathbb{R}$. For real vectors $\mathbf{r}, \mathbf{q}$ in a real vector space on $\mathbb{R}^D$, the following algebraic rules apply,

Definition 3.1.4 — Sum and scalar multiplication.

$$\mathbf{r} + \mathbf{q} = (r_1 + q_1, r_2 + q_2, ..., r_D + q_D) \tag{3.4}$$

$$\alpha \mathbf{x} = (\alpha x_1, \alpha x_2, ..., \alpha x_D) \tag{3.5}$$

These are rules we are familiar with from linear algebra, and this is how we know real vectors behave. However, as we saw in the previous chapter, this kind of addition does not work on compositions as there are neither neutral nor inverse elements. What about scalar multiplication? This is exactly how we defined equivalence classes in the previous chapter, which means that scalar multiplying a composition by a real number does not yield a new composition, but the same. It is clear that we cannot use these operations to define the simplex as a vector space

3.2 The Aitchison geometry

We need find a metric on the compositional sampling space, the simplex, that gives it Euclidean properties. The resulting geometry is called Aitchison geometry, named after the Scottish mathematician who first defined it in the early 1980s.

3.2.1 Vector space properties

We should define two new operations that are analogous to addition and scalar multiplication, which operate on elements in the simplex to yield a new element in the simplex. They should obey the following eight axioms:

$$
\begin{aligned}
&1. && \mathbf{v}+\mathbf{w}=\mathbf{w}+\mathbf{v} && (\textit{commutative law of addition}) && (3.6)\\
&2. && (\mathbf{v}+\mathbf{w})+\mathbf{x}=\mathbf{v}+(\mathbf{w}+\mathbf{x}) && (\textit{associative law of addition}) && (3.7)\\
&3. && \mathbf{v}+\mathbf{0}=\mathbf{v} && (\textit{additive identity law}) && (3.8)\\
&4. && \mathbf{v}+(-\mathbf{v})=\mathbf{0} && (\textit{additive inverse law}) && (3.9)\\
&5. && r(\mathbf{v}+\mathbf{w})=r\mathbf{v}+r\mathbf{w} && (\textit{distributive law}) && (3.10)\\
&6. && (r+s)\mathbf{v}=r\mathbf{v}+s\mathbf{v} && (\textit{distributive law}) && (3.11)\\
&7. && r(s\mathbf{v})=(rs)\mathbf{v} && (\textit{associative law of multiplication}) && (3.12)\\
&8. && 1\mathbf{v}=\mathbf{v} && (\textit{scalar identity law}) && (3.13)
\end{aligned}
$$

for $\mathbf{v},\mathbf{w},\mathbf{x}\in\mathscr{S}^D$ and $r,s\in\mathbb{R}$. It should be noted that the $\mathbf{0}$ vector in axiom 3, is not necessarily a vector with zeros in each entry (which does not exist in the simplex). It is the generalized neutral element, which, when added to any other element, returns the same element. Likewise, the minus sign in axiom 4 does not necessarily mean the negative-valued vector but can be any inverse element, which when added to the element itself yields the neutral element. We can quickly convince ourselves that ordinary addition and scalar multiplication break several of these axioms. Instead, we have two other operations, perturbation and powering, that work. They are defined as,

Definition 3.2.1 — Perturbation and powering.

$$
\begin{aligned}
\mathbf{x}\oplus\mathbf{y} &= \mathscr{C}\left(x_1y_1,x_2y_2,\ldots,x_Dy_D\right) && (3.14)\\
\alpha\odot\mathbf{x} &= \mathscr{C}\left(x_1^{\alpha},x_2^{\alpha},\ldots,x_D^{\alpha}\right) && (3.15)
\end{aligned}
$$

The circle around the plus and multiplication signs symbolizes that the operations are analogous to addition and scalar multiplication but not the same. It is left as an exercise to check that they obey the eight axioms of vector spaces. The meaning of perturbation and powering is hard to visualize, but they are used to scale and center a composition, something that we will encounter in a later lecture.

Just like for Euclidean real vector spaces, we can define an inner product,

Definition 3.2.2 — Aitchison inner product.

$$
\langle\mathbf{x},\mathbf{y}\rangle_a=\frac{1}{2D}\sum_{i=1}^{D}\sum_{j=1}^{D}\ln\frac{x_i}{x_j}\ln\frac{y_i}{y_j}\qquad\mathbf{x},\mathbf{y}\in\mathscr{S}^D \tag{3.16}
$$

for compositions on $\mathscr{S}^D$. This function obeys the three rules, scalar invariance, permutation invariance, and subcompositional coherence. The inner product gives us the norm,

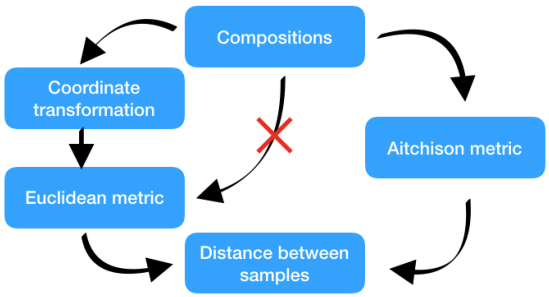

Figure 3.1: The two routes to obtaining the distance between compositions.

Definition 3.2.3 — Aitchison norm.

$$||\mathbf{x}||_a = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_a} = \sqrt{\frac{1}{2D} \sum_{i=1}^{D} \sum_{j=1}^{D} \left(\ln \frac{x_i}{x_j} \right)^2} \qquad \mathbf{x} \in \mathscr{S}^D \tag{3.17}$$

and from the norm we get the Aitchison distance between compositions,

Definition 3.2.4 — Aitchison distance.

$$d_a(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} \ominus \mathbf{y}||_a = \sqrt{\frac{1}{2D} \sum_{i=1}^{D} \sum_{j=1}^{D} \left(\ln \frac{x_i}{x_j} - \ln \frac{y_i}{y_j} \right)^2} \qquad \mathbf{x}, \mathbf{y} \in \mathscr{S}^D. \tag{3.18}$$

The generalized minus in the above definition denotes perturbation by the inverse. Together with the Aitchison geometry, the simplex becomes a linear vector space known as a (finite-dimensional) real Hilbert space, where standard Euclidean properties are valid, such as the triangle inequality, Pythagoras theorem, and the Cauchy-Schwartz inequality. We denote this vector space $(\mathscr{S}^D, \oplus, \odot)$ as opposed to ordinary real vector spaces $(\mathbb{R}^D, +, \cdot)$.

3.3 Transformations

Unfortunately, the Aitchison distance is somewhat cumbersome to work with due to the double summations. We will therefore introduce certain coordinate transformations, which will transform the compositions from the simplex onto the real Euclidean space, where the ordinary Euclidean metric applies. This means that rather than calculating the distance between compositions using the above-defined Aitchison distance, we transform the compositions from $\mathscr{S}^D$ onto a subspace of $\mathbb{R}^D$ and then use all the normal methods that apply to ordinary real vectors. The reason why this is a preferred route is that many statistical and explorative data analysis methods implicitly assume Euclidean distances, and they come pre-implemented in many modern programming languages, such as R, Python, Matlab, etc. So in order to not have to reimplement everything using Aitchison geometry, we can just transform our compositional data and work with the packages as they are.

As of today, we know of three different coordinate transformations that allow us to apply the Euclidean metric after transforming the data. Each transformation has pros and

cons, and in the end, it is up to the analyst to choose the one most appropriate for the problem at hand. There are no strict rules for which transformation should be used in a given situation, but sometimes one choice is more obvious than another.

3.3.1 Additive logratio transformation (ALR)

The first and most intuitive transformation is the *Additive logratio transformation*. This transformation maps a composition from $\mathscr{S}^D$ onto $\mathbb{R}^{D-1}$, that is, to a subspace of the D-dimensional real space. Formally, it is defined as,

Definition 3.3.1 — Additive logratio (ALR) coordinates. Given a D-part composition $\mathbf{x} = (x_1, x_2, ..., x_D)$ and using x_D as reference part, the alr transformation is given by

$$\operatorname{alr}(\mathbf{x}) = \left(\ln \frac{x_1}{x_D}, \ln \frac{x_2}{x_D}, ..., \ln \frac{x_{D-1}}{x_D} \right) = \zeta. \tag{3.19}$$

and its inverse

$$\operatorname{alr}^{-1}(\zeta) = \mathscr{C}\left(\exp(\zeta_1), \exp(\zeta_2), ..., \exp(\zeta_{D-1}), 1\right) = \mathbf{x}. \tag{3.20}$$

In the ALR transformation, one part (by convention, usually the last part of the composition) is chosen as the denominator. The interpretation of this is that all parts are given relative to one particular part, i.e., for a nutrition table of a food product, fat per protein, carbohydrates per protein, salt per protein, etc. The choice of denominator is up to the analyst and must be chosen to be meaningful with respect to the question at hand.

The reference part becomes zero in ALR-space because the ratio x_D/x_D is 1 and the logarithm of 1 is zero. ALR is also not a unique transformation since it depends on the choice of denominator. Taking the logarithm ensures that the algebraic operations perturbation and powering on the simplex $\mathscr{S}^D$ translates into the algebraic operations sum and multiplication in $\mathbb{R}^{D-1}$

$$\operatorname{alr}(\alpha \odot \mathbf{x} \oplus \beta \odot \mathbf{y}) = \alpha \cdot \operatorname{alr}(\mathbf{x}) + \beta \cdot \operatorname{alr}(\mathbf{y}).$$

This relation makes the ALR transformation an isomorphism and it ensures that real space operations, sum and multiplication, can be used, but not Euclidean distance. The reason for this is that ALR does not provide an isometry between $\mathscr{S}^D$ and $\mathbb{R}^{D-1}$ and therefore, distances between vectors are not invariant under the ALR transformation. The ALR transformation is frequently used in genomics, even if it is used implicitly (e.g., in so-called log(FPKM) values), and it is easy to interpret, but one should be careful with ALR if any kind of intersample distance or metrics are involved in the analysis. The rule is that analysis methods can be applied to ALR values if they are *affine equivalent*. Methods are affine equivalent if the results are the same after translating, rotating, or scaling the data, so that the result is translated, rotated or scaled as well. This means that only algebraic vector-space operations are involved and no metric concepts. It is not always obvious when this is the case.

3.3.2 Centered logratio transformation (CLR)

Another transformation is the *Centered logratio transformation*. The CLR transformation uses a fixed constant as the denominator rather than a specific part and is formally defined

as,

Definition 3.3.2 — Centered logratio (CLR) values. Given a D-part composition $\mathbf{x} = (x_1, x_2, ..., x_D)$, CLR values are given by,

$$\text{clr}(\mathbf{x}) = \left(\ln \frac{x_1}{g_m(\mathbf{x})}, \ln \frac{x_2}{g_m(\mathbf{x})}, ..., \ln \frac{x_D}{g_m(\mathbf{x})} \right) = \zeta. \tag{3.21}$$

where g_m is the geometric mean

$$g_m(\mathbf{x}) = \left(\prod_{i=1}^{D} x_i \right)^{1/D} = \exp\left(\frac{1}{D} \sum_{i=1}^{D} \ln x_i \right) \tag{3.22}$$

and its inverse

$$\text{clr}^{-1}(\zeta) = \mathscr{C}\left(\exp(\zeta_1), \exp(\zeta_2), ..., \exp(\zeta_D)\right) = \mathbf{x}. \tag{3.23}$$

The definition is very similar to ALR. The only difference is that the denominator in the ratio is the geometric mean of $\mathbf{x}$, rather than a specific part. We could, in principle use any constant as denominator, due to the fact that compositions are equivalence classes, but when using the geometric mean, the sum of CLR values will always be 0. The interpretation of CLR values is that each part is given relative to the mean of all parts: positive values are larger than the mean, and negative parts are smaller than the mean. A CLR-transformed vector will, in general, have as many parts as the original composition, so the CLR transform maps coordinates from $\mathscr{S}^D$ onto $\mathbb{R}^D$. However, because the sum of a CLR vector is always 0, the vector is constrained to a subspace of $\mathbb{R}^D$ as illustrated in Fig. 3.2. CLR values always fall on a $(D-1)$-dimensional plane in $\mathscr{R}^D$ defined by the point $\mathbf{r}_0 = (0, 0, ..., 0)$ and the normal vector $\mathbf{n} = (1, 1, ..., 1)$. Because of this constraint, CLR values do not span $\mathbb{R}^D$ and they are therefore not coordinates on any orthonormal basis of $\mathbb{R}^D$. Hence we refer to CLR values as CLR *coefficients*, rather than CLR coordinates.

CLR values obey the rules

$$\text{clr}(\alpha \odot \mathbf{x} \oplus \beta \odot \mathbf{y}) = \alpha \cdot \text{clr}(\mathbf{x}) + \beta \cdot \text{clr}(\mathbf{y}) \tag{3.24}$$

$$\langle \mathbf{x}, \mathbf{y} \rangle_a = \langle \text{clr}(\mathbf{x}), \text{clr}(\mathbf{y}) \rangle_e \tag{3.25}$$

$$d_a(\mathbf{x}, \mathbf{y}) = d_e(\text{clr}(\mathbf{x}), \text{clr}(\mathbf{y})), \tag{3.26}$$

which means that the CLR transformation is both an isomorphism and an isometry, and it therefore conserves the metric properties of a composition.

The downside of the CLR transformation is that, due to the fact that they are confined to a subplane of $\mathbb{R}^D$, the sum of their values is zero, which makes the determinant of the covariance matrix of a set of compositions zero. This means that certain statistical methods that rely on the covariance matrix being non-singular cannot be applied to CLR values. When the determinant of the covariance matrix is zero, it means that the parts are perfectly correlated, which is true by construction for CLR values, as illustrated in the following example.

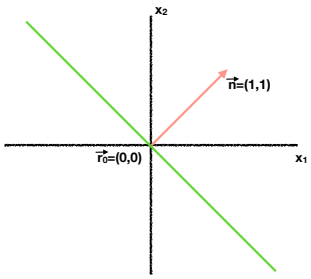

Figure 3.2: The green line is the subplane of $\mathbb{R}^2$ in which all 2-part CLR values exist. The plane is defined as the space going through (0,0) and having the vector (1,1) as normal. This generalizes to arbitrary dimensions.

■ **Example 3.2** Consider the 2 2-part compositions

$$\mathbf{x} = \begin{pmatrix} 40 \\ 60 \end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix} 20 \\ 80 \end{pmatrix}$$

From eq. 3.22 we can calculate the geometric means,

$$g_m(\mathbf{x}) \approx 49, \qquad g_m(\mathbf{y}) \approx 40$$

and then the CLR transform,

$$\text{clr}(\mathbf{x}) = \begin{pmatrix} -0.2 \\ 0.2 \end{pmatrix}, \qquad \text{clr}(\mathbf{y}) = \begin{pmatrix} -0.69 \\ 0.69 \end{pmatrix}$$

The CLR values are symmetric around 0 (as they should be), there is only 1 independent variable (they are confined to a 1D plane in 2D space), and there is a perfect correlation between the parts (one goes up and the other goes down, by the exact same amount). ■

Another problem with the CLR transformation is that the values are not subcompositionally coherent because the geometric mean will change if parts are removed, and thus the CLR values will generally be different for different subcompositions. This gives rise to a *very important* limitation of the CLR transformation: we cannot pre-calculate CLR values and choose to analyze a subset of the values. CLR values will have to be calculated for a given subcomposition, and they are only valid within the scope of that exact subcomposition.

3.3.3 Orthonormal coordinates

Coordinates of a vector are expressed with respect to an orthogonal basis, typically the canonical basis of $\mathbb{R}^D$, $\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_D\} = \{[1,0,...,0],[0,1,...,0],...,[0,0,...,1]\}$. Any vector $\mathbf{v} \in \mathbb{R}^D$ can be written in the form,

$$\mathbf{v} = v_1\mathbf{e}_1 + v_2\mathbf{e}_2 + ... + v_D\mathbf{e}_D = \sum_{i=1}^{D} v_i \cdot \mathbf{e}_i. \tag{3.27}$$

The canonical basis of $\mathbb{R}^D$ is not, however, a basis with respect to $\mathscr{S}^D$ since the vectors are not part of the simplex itself. Just look at all the zeros. But we can make the canonical basis into a spanning set for $\mathscr{S}^D$ by taking the closure of the exponentials,

$$\begin{aligned}\{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_D\} &= \{\mathscr{C}(\exp(\mathbf{e}_1)), \mathscr{C}(\exp(\mathbf{e}_2)), ..., \mathscr{C}(\exp(\mathbf{e}_D))\} \\ &= \{\mathscr{C}[e, 1, ..., 1], \mathscr{C}[1, e, ..., 1], ..., \mathscr{C}[1, 1, ..., e]\}. \end{aligned} \tag{3.28}$$

The vectors $\mathbf{w}_i$ span the simplex, but they are not linearly independent and therefore not a basis, since there are D vectors in the set and the dimensionality of $\mathscr{S}^D$ is $D-1$.

We can express compositions in the same form as eq. 3.27 by replacing operations with their simplex analogous,

$$\begin{aligned}\mathbf{x} &= \bigoplus_{i=1}^{D} \ln \frac{x_i}{g_m(\mathbf{x})} \odot \mathbf{w}_i \\ &= \ln \frac{x_1}{g_m(\mathbf{x})} \odot [e, 1, ..., 1] \oplus \ln \frac{x_2}{g_m(\mathbf{x})} \odot [1, e, ..., 1] \oplus ... \oplus \ln \frac{x_D}{g_m(\mathbf{x})} \odot [1, 1, ..., e] \\ &= \left[\frac{x_1}{g_m(\mathbf{x})}, \frac{x_2}{g_m(\mathbf{x})}, ..., \frac{x_D}{g_m(\mathbf{x})}\right] \\ &= [x_1, x_2, ..., x_D],\end{aligned} \tag{3.29}$$

where we have used the definition of compositions as equivalence classes in the final step. However, $\{\mathbf{w}\}$ is not a basis, and therefore, $\ln \frac{x_i}{g_m(\mathbf{x})}$, which are recognized as CLR values, are not coordinates. We can drop any one of the $\mathbf{w}_i$'s to get a proper basis, but, by taking the Aitchison dot product between any two vectors in this basis, it is easily seen that it is not an orthogonal basis. We will explore how to build an orthonormal basis in Sect. 3.3.5, but for now let us assume that such a basis exists and is denoted $\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_{D-1}\}$. Then we can define the *contrast matrix*,

Definition 3.3.3 — Contrast matrix. Given an orthonormal basis of the simplex $\mathscr{S}^D$, $\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_{D-1}\}$, the *contrast matrix* $\mathbf{\Psi}$ is a $(D, D-1)$-matrix where each row $\mathbf{\Psi}_i = \text{clr}(\mathbf{e}_i)$, $i = 1, 2, ..., D-1$. Each row is a logcontrast.

An orthonormal basis has the property that the dot-product between basis vectors is zero, while the dot-product between a basis vector and itself is 1,

$$\langle \mathbf{e}_i, \mathbf{e}_j \rangle_a = \langle \text{clr}(\mathbf{e}_i), \text{clr}(\mathbf{e}_j) \rangle_e = \delta_{ij}, \tag{3.30}$$

where δ_{ij} is the Kronecker-delta which is a function that equals 0 for $i \neq j$ and 1 for $i = j$. From this, it follows that,

$$\mathbf{\Psi}\mathbf{\Psi}^T = \mathbf{I}_{D-1},$$

that is, the identity matrix in $D-1$ dimensions.

3.3.4 Isometric logratio transformation (ILR)

We can now introduce a third transformation which is known as the *Isometric logratio transformation*. As the name suggests, it too conserves distances (it is an isometry), and contrary to CLR, it does not have a singular covariance matrix. It has all the benefits of

both ALR and CLR and none of their downsides. This comes at a cost, however: ILR values are expressed as coordinates in an orthonormal basis of the simplex, meaning that the ILR coordinates can be difficult to interpret.

As discussed above, we can obtain coordinates by expressing a compositions in an orthonormal basis according to Eq. 3.29. We can furthermore replace the Aichison operators with their Euclidean analogs, by using the CLR transform,

$$\mathrm{CLR}(\mathbf{x}) = \sum_{i=1}^{D} x_i^* \cdot \mathrm{CLR}(\mathbf{e}_i) = \mathbf{x}^* \cdot \mathbf{\Psi} \iff \mathbf{x}^* = \mathrm{CLR}(\mathbf{x}) \cdot \mathbf{\Psi}^T \tag{3.31}$$

This transformation is called ILR and is basically obtained by multiplying the CLR transformation with a contrast matrix $\mathbf{\Psi}$ with dimensions (D-1,D),

Definition 3.3.4 — Isometric logratio (ILR) coordinates. Given a D-part composition $\mathbf{x} = (x_1, x_2, ..., x_D)$ and a contrast matrix $\mathbf{\Psi}_{D-1,D}$ based on the basis $\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_{D-1}\}$, ILR coordinates are given by

$$\mathrm{ilr}(\mathbf{x}) = \mathrm{clr}(\mathbf{x}) \cdot \mathbf{\Psi}^T = \mathbf{x}^* \tag{3.32}$$

and its inverse

$$\mathrm{ilr}^{-1}(\mathbf{x}^*) = \mathscr{C}(\exp(\mathbf{x}^* \mathbf{\Psi})) = \mathbf{x} \tag{3.33}$$

ILR coordinates obey the same properties as CLR coefficients (3.24) but they are true coordinates in $\mathbb{R}^{D-1}$. The downside is that each part in the ILR-transformed vector is a linear combination of parts from the composition. ILR transforms are also not unique, since infinitely many orthonormal bases can be constructed. In any case, before the ILR transform can be applied, a basis has to be constructed. One option is to use *balances* as basis vectors. These are formed by applying a sequential binary partition of the parts of a composition. Many computational CoDa packages have the option to use this basis for ILR transformation.

3.3.5 Balances

A popular choice of basis to use in ILR transformation are balances. Balances are formed from the sequential binary partition of the composition and represent groups of features. This partition is a hierarchical division of the parts, splitting the parts into two groups, each of which is again split into two groups, and so on and so forth until there are $D-1$ balances. An example of a binary partition table of a six-part composition is

x_1	x_2	x_3	x_4	x_5	x_6
+1	+1	+1	-1	-1	-1
+1	+1	-1	0	0	0
+1	-1	0	0	0	0
0	0	0	+1	+1	-1
0	0	0	+1	-1	0

The coordinates of this partition are called balances, and the vectors are called balancing elements. In this example, we have chosen to make the first balance an even split between

parts: (x_1, x_2, x_3) versus (x_4, x_5, x_6), but we could just as well have chosen to split (x_1, x_2) versus (x_3, x_4, x_5, x_6).

When we have decided on a binary partition table, we can form the balances by counting the number of positive and negative entries per row. Let us denote the number of plus signs by r and the number of minus signs by s and let us refer to the row in the partition table by k. In the example above, for the first row, $k = 1$, $r = 3$ and $s = 3$. For $k = 2$, $r = 2$ and $s = 1$, and so on. A balance is defined as the (normalized) logratio of the geometric mean of the two groups (plus- and minus group),

$$b_k = \sqrt{\frac{rs}{r+s}} \ln \frac{(x_{i_1} x_{i_2} \ldots x_{i_r})^{1/r}}{(x_{j_1} x_{j_2} \ldots x_{j_s})^{1/s}}, \tag{3.34}$$

where the square root in front is the normalizing factor. We can use the logarithm rules to formulate this as,

$$\begin{aligned} b_k &= \ln \frac{(x_{i_1} x_{i_2} \ldots x_{i_r})^{a_+}}{(x_{j_1} x_{j_2} \ldots x_{j_s})^{a_-}} \\ &= \ln(x_{i_1} x_{i_2} \ldots x_{i_r})^{a_+} - \ln(x_{j_1} x_{j_2} \ldots x_{j_s})^{a_-} = \sum_{j=1}^{D} a_{kj} \ln x_j \end{aligned} \tag{3.35}$$

where

$$a_+ = +\frac{1}{r}\sqrt{\frac{rs}{r+s}}, \qquad a_- = -\frac{1}{s}\sqrt{\frac{rs}{r+s}}. \tag{3.36}$$

We recognize Eq. 3.35 as a logcontrast, as introduced in Chapter 2, Eq. 2.5. So if we form a matrix out of the balances, then each row k is a logcontrast, it is also a CLR transform (they sum to zero), they are a linearly independent spanning set, and they are normalized. Hence they form an orthonormal basis and the matrix is a contrast matrix $\mathbf{\Psi}$. For the example partition table above, the contrast matrix of balances is,

x_1	x_2	x_3	x_4	x_5	x_6
$+\frac{1}{3}\sqrt{\frac{3\cdot3}{3+3}}$	$+\frac{1}{\sqrt{6}}$	$+\frac{1}{\sqrt{6}}$	$-\frac{1}{\sqrt{6}}$	$-\frac{1}{\sqrt{6}}$	$-\frac{1}{\sqrt{6}}$
$+\frac{1}{2}\sqrt{\frac{2\cdot1}{2+1}}$	$+\frac{1}{\sqrt{6}}$	$-\sqrt{\frac{2}{3}}$	0	0	0
$+\frac{1}{1}\sqrt{\frac{1}{2}}$	$-\frac{1}{\sqrt{2}}$	0	0	0	0
0	0	0	$+\frac{1}{\sqrt{6}}$	$+\frac{1}{\sqrt{6}}$	$-\sqrt{\frac{2}{3}}$
0	0	0	$+\frac{1}{1}\sqrt{\frac{1}{2}}$	$-\frac{1}{\sqrt{2}}$	0

The sequential binary partition is not unique in the sense that we can decide how many parts should go into each group. In the above example, we chose to divide the composition into two groups of three parts each in the first step, but we could just as well have choose one group with two parts and one with four. It is up to the analyst to choose a partition that optimizes the interpretability of the result.

3.3.6 Example transformations

We will end this chapter by logratio transforming a composition, using all three different logratio transforms.

■ **Example 3.3** Consider the composition $\mathbf{x} = (25, 30, 45)$, which is closed to 100.

For the ALR transform, we need to choose a part. In this example, we choose the third part x_3.

$$\text{ALR}(\mathbf{x}) = \ln \begin{pmatrix} 25/45 \\ 30/45 \end{pmatrix} \approx \begin{pmatrix} -0.59 \\ -0.41 \end{pmatrix} \tag{3.37}$$

For the CLR transform, we need to calculate the geometric mean,

$$g_m(\mathbf{x}) = (25 \cdot 30 \cdot 45)^{1/3} \approx 32.32, \tag{3.38}$$

which then gives the CLR values,

$$\text{CLR}(\mathbf{x}) = \ln \begin{pmatrix} 25/32.32 \\ 30/32.32 \\ 45/32.32 \end{pmatrix} \approx \begin{pmatrix} -0.26 \\ -0.07 \\ 0.33 \end{pmatrix} \tag{3.39}$$

For the ILR transform, we need to build an orthonormal basis. We choose a basis made from balances, using the following sequential partition table,

x_1	x_2	x_3
+1	+1	-1
+1	-1	0

From this table we calculate the balances to form the contrast matrix,

$$\boldsymbol{\Psi} = \begin{pmatrix} \frac{1}{2}\sqrt{\frac{1\cdot 2}{1+2}} & \frac{1}{2}\sqrt{\frac{1\cdot 2}{1+2}} & -\frac{1}{1}\sqrt{\frac{1\cdot 2}{1+2}} \\ \frac{1}{1}\sqrt{\frac{1\cdot 1}{1+1}} & -\frac{1}{1}\sqrt{\frac{1\cdot 1}{1+1}} & 0 \end{pmatrix} \approx \begin{pmatrix} 0.41 & 0.41 & -0.82 \\ 0.71 & -0.71 & 0 \end{pmatrix}$$

The ILR coordinates are obtained by matrix multiplying the CLR values by the transposed contrast matrix,

$$\text{ILR}(\mathbf{x}) = \begin{pmatrix} -0.26 \\ -0.07 \\ 0.33 \end{pmatrix} \cdot \begin{pmatrix} 0.41 & 0.41 & -0.82 \\ 0.71 & -0.71 & 0 \end{pmatrix}^T \approx \begin{pmatrix} -0.41 \\ -0.13 \end{pmatrix} \tag{3.40}$$

If we plot the ILR coordines in a cartesian coordinate system, the axis would represent the logarithm of $x_1 + x_2 - x_3$ and $x_1 - x_2$, respectively. ■

3.4 Exercises

Exercise 3.1 Consider the two vectors $\mathbf{x} = [0.7, 0.5, 0.8]$ and $\mathbf{y} = [0.25, 0.75, 0.5]$. Perturb one vector by the other, with and without previous closure. Are there any differences? ∎

Exercise 3.2 Calculate the Aitchison inner product of $\mathbf{x} = \mathscr{C}[0.7, 0.4, 0.8]$ and $\mathbf{y} = \mathscr{C}[2, 8, 1]$. Are they orthogonal? ∎

Exercise 3.3 Consider the two six-part compositions, given in percentages,

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} 3.74 & 9.35 & 16.82 & 18.69 & 23.36 & 28.04 \\ 9.35 & 28.04 & 16.82 & 3.74 & 18.69 & 23.36 \end{pmatrix}.$$

Calculate the Aitchison norms and the Aitchison inner product. What is the angle between the two compositions?
HINT: $\cos(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}||||\mathbf{y}||}$ ∎

Exercise 3.4 Compute the Aitchison norm of $\mathbf{x} = \mathscr{C}[0.7, 0.4, 0.8]$ and call it a. Compute $\alpha \odot \mathbf{x}$ with $\alpha = 1/a$. Compute the Aitchison norm of the resulting composition. How do you interpret the result? ∎

Exercise 3.5 Redo Exercise 2.5 from Chapter 2, but using the Aitchison distance. Is it subcompositionally dominant? ∎

Exercise 3.6 Using the data from Exercise 2.3, compute the CLR coefficients. Verify that the sum of the transformed components equals zero. ∎

Exercise 3.7 Using the data from Exercise 2.3, apply the ALR transformation to the compositions. Plot the transformed data in $\mathbb{R}^2$. Now do it using a different part as the denominator for the ALR transformation. Compare the results. ∎

Exercise 3.8 Show that the exponential of the canonical basis,

$$\begin{aligned} \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{D-1}\} &= \{\mathscr{C}(\exp(\mathbf{e}_1)), \mathscr{C}(\exp(\mathbf{e}_2)), ..., \mathscr{C}(\exp(\mathbf{e}_{D-1}))\} \\ &= \{\mathscr{C}[e, 1, ..., 1], \mathscr{C}[1, e, ..., 1], ..., \mathscr{C}[1, ..., e, 1]\} \end{aligned}$$

is not orthogonal on $\mathscr{S}^D$. ∎

Exercise 3.9 Build a sequential partition table of a three-part composition and calculate the corresponding balances. Write out the contrast matrix. ∎

Exercise 3.10 Using the contrast matrix from the previous exercise, redo Exercise 3.7, with the ILR transformation. Plot the result. Compare with a different contrast matrix. ∎

4. Compositions with zero values

4.1 Why do zeros occur in compositions?

So far, compositions have been defined as vectors containing strictly positive real values that carry only relative information. It is because of this latter requirement that entries in a composition cannot be zero. It is not possible to infer information about a part relative to another zero-valued part since this ratio would approach infinite as the denominator value would approach zero. It is also not possible to log-transform a composition containing zeros since the logarithm of zero is undefined.

In many cases, parts that are zero are simply left out. If, for instance, we want to count the cars in different colors passing through a traffic light within an hour, we do not need to report that we didn't see any pink, teal, or khaki colored cars. In fact, if we didn't leave out zero-valued parts, our composition would become infinite, since there is a (near) infinite number of car colors that we did not see.

The problem arises when we want to compare two or more compositions, where not all parts are present in all compositions. If we count cars at two different intersections in order to compare the color distribution of the cars, we might find that at the first site we see red, blue, and green cars but no yellow cars, whereas at the second site we see red, blue, and yellow cars but no green cars. In this case, we need to deal with the zero values.

A vector that contains zero values is not a composition, and therefore we cannot use compositional data analysis on such a vector. One of the basic requirements for compositional data analysis is that the data be scale invariant, and that is not fulfilled if there is a zero. Also, when applying closure to a composition containing a zero, we need the total sum, which is then equal to the total sum of the subcomposition without the zero component, which means we will be closing the full composition using the sum of a subcomposition.

In practice, there are several reasons why a zero value may occur, and depending on what kind of zero it is, we need to get rid of it in the appropriate way.

4.1.1 Rounded values or values below the detection limit

When a composition is made up of continuous variables (say, weight, percentages, time, and not counts), rounded zeros may occur if the significant digit is below the number of digits used to represent the parts. The non-zero observed value gets rounded off and becomes zero.

A similar kind of zero is when a measurement is zero because it falls within the measurement noise level or if the measurement equipment is not sensitive enough to pick up very small quantities. An example of this is food products where the ingredient list says "may contain traces of nuts". The product should not contain any nuts, and if measured, it comes out at zero percent, but in very large batches of the product, small amounts of nut may still be present due to contamination. This is a zero due to the detection limitation.

A third variation of rounded zeros are censored values, which are the same as detection limit zeros but not limited to small values. If our measuring equipment saturates at a certain high value, e.g., if you overexpose a photograph, or if your device is limited to a certain maximum value, for instance, a volt meter that can measure up to 5 volts is used in a 230-volt wall socket, then we get a censored value. The measurement comes out at 5 volts, but we know that the true value is much greater, so the true value is missing.

For these kinds of zeros, as well as for censored values, a reasonable strategy is to replace the zero with a fixed value. This is called non-parametric replacement, since all zeros are replaced by the same value. A typical choice is to use a small fraction of the rounding/detection limit. For a D-part composition $\mathbf{x} = (x_1, x_2, ..., x_D)$ containing a number of rounded zeros and a constant sum s, the composition is replaced by the composition $\mathbf{x}'$ according to the formula

$$x'_i = \begin{cases} \delta_i & \text{if } x_i = 0 \\ x_i \left(1 - \frac{1}{s} \sum_{k|x_k=0} \delta_k\right) & \text{if } x_i > 0. \end{cases} \tag{4.1}$$

This ensures that after replacement, the composition's sum stays the same. The problem with this kind of replacement is that it distorts the covariance matrix by introducing false correlation between compositions where the same part(s) equal zero. This effect becomes larger as the number of zeros increases. It has been shown that non-parametric replacement works best if the compositions contain less than 10% zeros and if the replacement value is 65% of the detection limit.

In the case where the number of rounded zeros increases beyond 10%, we could consider using a parametric replacement method. This method only works if we have a number of samples represented as compositions, that we wish to compare. If a sample has a part that is zero due to rounding, we consider the distribution of non-zero values in the remaining samples and make the assumption that the part has a normal distribution across the samples. The sample(s) where the part is zero will then get a replacement value, which is randomly sampled for the range of this distribution that falls below the detection/rounding limit. Parametric replacement introduces less false correlation, but it only works if the zeros are of this kind.

4.1.2 Structural zeros

Structural zeros, also known as essential zeros, occur when a part cannot be observed in one of several samples. An example of structural zeros is how votes are distributed among political parties in different election districts, if some of the parties are not represented in all districts. In this case, those parties could not receive any votes in those districts, and the result is a part that is zero. Another example is the food consumption in a number of families, which includes some families on a vegan diet. In those samples, the part representing meat consumption would be structurally zero.

A structural zero should obviously not be replaced by a non-zero value, and there is, at the moment, no general way to deal with them. In some cases, we could turn the part into a binary categorical presence/absence variable, which takes one value if the part is non-zero and another value if it is zero, which however makes it difficult to maintain a meaningful closure. Another strategy is simply to leave out the parts of some samples that contain structural zeros. This is only useful if the number of structural zeros is small and the parts in which they are present are of no particular interest to the analysis.

4.1.3 Missing values

Missing values are related to the way that the data is obtained and can be minimized by careful data acquisition. Missing values can, for instance, occur if patients are asked to fill out a questionnaire after their treatment and they do not answer all questions. Missing values can be divided into three categories: *Not missing at random* (NMAR), *Missing at random* (MAR), and *Missing completely at random* (MCAR).

In the case of patients participating in a survey, an example of NMAR is when one question is particularly difficult, embarrassing, or takes a long time to answer. Then people might be inclined to skip it, and there will be a bias in the data set towards missing this value. MAR is the case when the patient is asked to skip a number of questions depending on their answer to another question. For example, if a question reads, “Were you born in Denmark?” then the survey could instruct the patient to only answer the next three questions in case the answer is no. MCAR is a simple situation where a question is simply overlooked and forgotten at random.

Depending on the type of missing value, we can adopt varying strategies to deal with them, but in general, missing values should be minimized by design. The only type of missing value that is relevant to genomics are NMAR, where the solution is either to re-sequence or re-map or discard the sample.

4.1.4 Amalgamated values

Amalgamated values are a special kind of missing value where some parts of one sample have been amalgamated and reported separately in another sample. In a metagenomic scenario, this could occur if two samples have been mapped differently, one against bacteria and protozoa separately and another where bacteria and protozoa have been merged and are called microorganisms. In this case, the first sample will contain a zero in the part called microorganisms, while the other will have zero values in the parts bacteria and protozoa. These zeros are extremely difficult to deal with because there is no way in which amalgamated values can be disentangled. The only real solution is to amalgamate the parts in the whole data set or discard the samples with amalgamated values. In any case, the problem with amalgamated values can be minimized by careful design of the study.

4.1.5 Counting zeros

Counting zeros is by far the most difficult type of zero to handle, and unfortunately, these are the zeros that are encountered in metagenomics. Counting zeros can occur in count data, where a count represents the number of times an event (part) occurs. All data, where a random sample is drawn from a population in order to represent the population distribution, are count data. In this case, a zero value may occur if the drawn sample is too small, so that the population distribution is not properly represented by the sample, simply because a part may be too rare for the random sampling to have picked it up.

A vector of counts may not even be a composition in the strict sense of the term since it may not obey the principle of scale invariance. If we conduct an opinion poll in two cities and find that a party receives 10 votes out of 20 people asked in the first city and 15 out of 20 votes in the second city, the principle of scale invariance says that we should see 1000 votes out of 2000 people asked in the first city and 1500 out of 2000 in the second. But what if a party receives zero out of 20 votes in the first city? Can we automatically assume that the number will still be zero when we ask 2000 people? If we find that 10 out of 2000 people would vote for the party, then the zero in the first survey would be considered a "below the detection limit"-zero, but if the number of votes among 2000 people is still zero, then it becomes more of a structural zero.

Likewise in metagenomics, if we have two samples that are sequenced to different depths, that is, one sample is sequenced to 1 million reads and the other to 1 billion reads, then an organism only found in the second sample at low counts is probably just below the detection limit in the first sample, whereas an organism found in the first sample but not in the second is probably structurally not present in the second sample.

Even when a count composition does not contain zero values, the non-zero parts may still be dependent on the size of the sample, that is, the total sum. If we toss a coin four times, we might get three heads and one tail, or a 3:1 ratio of the two parts. If we toss the same coin a thousand times (and if the coin is fair), we will find a heads-to-tail ratio closer to 1:1. The real composition is the true distribution of heads and tails on a coin (which is unknown to us), and the vector we observe, which is not a true composition, is a random realization of the underlying composition. This is called a latent composition model, and it assumes that the observed data is a known function of an underlying, unobserved composition. A wide range of methods exist to deal with latent compositions. The benefit is that we can analyze count data as if it were real compositions, while the downside is that it requires intimate knowledge of how the observations are linked to the latent composition.

4.2 Zero replacement in count data

We need to establish the function that maps the sample to the latent composition, and given that the sample is randomly drawn from the latent composition, our function needs to be stochastic by nature.

So far in this course, we have taken the frequentist approach to statistics. Frequentism is the paradigm in statistics where probability is defined as the limit of the relative frequencies after many repeated trials. If a coin is flipped 100 times and we get 49 heads, frequentism says that the probability of getting a head is 49/100 = 49%. If the coin is fair, this ratio will approach 1/2 = 50% as the number of coin flips approaches infinity. This is the kind of statistics with which most people are familiar. It has certain limitations however. If we roll

a dice six times and get (1,1,2,5,6,6), frequentism suggests that there is 0/6=0% chance of rolling 3 and 4, which is probably not true. Of course, we can roll the dice more times, and eventually, all frequencies will converge on 1/6, but in cases where we cannot do more trials and we get zero for some outcomes, we need to change our approach. This is exactly the situation when we have sequenced a sample to a certain depth (and cannot re-sequence deeper) and we did not record any reads belonging to a certain organism, which should be present but at a low abundance (i.e., the zero is not an essential zero).

▪ **Example 4.1 — Winning the lottery.** Frequentism can sometimes lead to paradoxical results. Let us say that we want to make a statistical test to see if it is more likely that you will win the lottery if you play, compared to if you don't play. A statistician asks two people to participate in an experiment. For 10 years, twice a week, one person is supposed to play the lottery and the other is supposed not to play the lottery. This results in more than 1000 trials for each person, with the outcome won/not won. Both individuals are asked to record the number of times they have won the lottery. When the experiment is over, neither person won, resulting in a winning probability of 0% in both cases. The frequentist will conclude that playing the lottery does not improve your chance of winning to a very high degree of confidence. ▪

Bayesian probability is a different approach to statistics in which probability is interpreted as a reasonable expectation. The lottery paradox does not exist in Bayesian statistics, because winning the lottery when you don't play is not a reasonable expectation. In Bayesian inference, a prior probability distribution is assigned to a hypothesis, which is then updated by observations to form the posterior distribution. One property of Bayesian inference is that if you have assigned any non-zero probability to an outcome as a prior (yes, it is actually possible to win the lottery), then no matter what the observed trials show, the posterior will also have a non-zero probability of winning, however small. Only if you assign zero as the prior probability of winning (because that is reasonable in the case where you don't play) will you get a zero posterior probability for winning, as long as your observations also don't show any wins. The larger your observed sample is, the less the choice of prior matters and vice versa.

Let $\mathbf{x}$ be our observed sample and θ be the parameters that determines the latent composition that we seek. Bayes theorem states that the posterior probability distribution equals the observed likelihood estimate times a prior distribution, normalized by a factor that ensures that the posterior integrates to 1,

Definition 4.2.1 — Bayes theorem.

$$p(\theta|\mathbf{x}) = \frac{1}{C} p(\mathbf{x}|\theta)p(\theta), \qquad C = p(\mathbf{x}) = \int p(\mathbf{x}|\theta')p(\theta')d\theta'. \tag{4.2}$$

$p(\theta|\mathbf{x})$ is the posterior and $p(\theta)$ is the prior. The normalization C is in general very difficult to compute. However, without it, the posterior distribution is not normalized. One convenient trick to avoid calculating C is by using a so-called conjugate prior as a prior. With a conjugate prior, the posterior distribution is guaranteed to have the same algebraic form as the prior, so if the prior is a known normalized distribution, then the posterior can be normalized in the same way, and we avoid an explicit calculation of C.

In the case of genomic sequencing, the observed sample will follow a multinomial distribution. The multinomial is a generalization of the binomial distribution, and it gives

the probability of counts of each of the D parts of the latent composition after sampling it n times. A single conjugate prior exists for the multinomial, namely the Dirichlet distribution, which is the multivariate generalization of the Beta distribution (which is the conjugate prior for the binomial distribution).

The Dirichlet distribution is parameterized by a vector α,

$$Dirichlet(\mathbf{r},\alpha) = \frac{1}{B(\alpha)}\prod_{i=1}^{D} r_i^{\alpha_i - 1}, \qquad B(\alpha) = \frac{\prod_{i=1}^{D}\Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{D}\alpha_i)} \tag{4.3}$$

It has support on the D-1 simplex, which means that a random sample taken from a D-dimensional Dirichlet distribution is a composition. α is known as the concentration parameter, and its components can take any real number greater than zero. If we have no prior knowledge of the abundance of the parts in our sample, it is most reasonable to use the the same value for all α_i. In that case we call it a symmetric Dirichlet distribution, and the scalar α is the concentration.

For a multinomial sample $p(\mathbf{x}|\theta)$ and a Dirichlet prior $Dirichlet(\alpha_0)$, it can be show that the posterior is given by $Dirichlet(\alpha')$, where $\alpha' = \alpha_0 + \mathbf{x}$. From this posterior, we can then extract the mean or the mode maximum likelihood point estimators for the latent compositions, as well as get an estimate of the variance in each part by drawing a number of random samples from the posterior.

4.2.1 The concentration parameter

When $\alpha = 1$, the symmetric Dirichlet distribution is equivalent to a uniform distribution over the simplex, and this is known as a flat Dirichlet distribution. α-values above 1 result in a dense distribution, where the values within a sample are more similar to each other, whereas with $\alpha < 1$, we get a sparse Dirichlet distribution, where most values are kept close to zero and only a few parts contain the mass of the composition.

In genomic applications, we probably prefer a sparse distribution, given that all genes (or species) are not equally likely to be found in the sample and certainly not in equal abundance. By choosing a sparse distribution, we acknowledge that zeroentries in the composition are zero, not by random chance but because that particular part is rare (or non-existing).

If we have prior knowledge of the relative distribution of components in our sample, we can chose α as $\alpha\mathbf{n}$, where $\mathbf{n}$ is a composition on the corresponding simplex.

We should keep in mind that we can never replace a zero with the "correct" number. We can at most hope to obtain a value – given our choice of concentration parameter – with which our zero value observation is statistically consistent. One should always use the same concentration parameter across a set of samples to make sure that zeros are weighted similarly throughout the data set.

4.2.2 Statistical estimates

There are two different point estimates that can be derived from a probability distribution: the expectation value or mean, E[$\mathbf{X}$], and the mode, Mode[$\mathbf{X}$], which gives the most frequent value.

The mean of the i'th part of a Dirichlet distribution is given by

$$\mathrm{E}[x_i] = \frac{\alpha_i'}{\sum_{k=1}^{D}\alpha_k'}. \tag{4.4}$$

The i'th mode is given by

$$\text{Mode}[x_i] = \frac{\alpha_i' - 1}{\sum_{k=1}^{D} \alpha_k' - D}, \quad \text{for} \quad \alpha_i' > 1. \tag{4.5}$$

The mode is not useful for replacing zeros because it returns zero if a zero is observed. One additional statistical estimate is the Aitchison mean, which is

$$\text{E}_a[\mathbf{X}] = \mathscr{C}[\exp[\Psi(\alpha_1')], \exp[\Psi(\alpha_2')], ..., \exp[\Psi(\alpha_D')]], \tag{4.6}$$

where Ψ is the the Euler digamma function, the logarithmic derivative of the Euler gamma,

$$\Psi(x) = \frac{d}{dx}\log(\Gamma(x)), \qquad \Gamma(x) = \int_0^\infty z^{x-1}\exp(-z)dz \tag{4.7}$$

Luckily, the digamma function is provided in special functions packages for most programming languages, such as Scipy for Python.

▪ **Example 4.2 — Estimating probabilities from counts.** Five parties run for the general elections. In order to predict the outcome, we ask 25 people how they intend to vote. Their reply is a composition, $\mathbf{x} = (12, 8, 3, 2, 0)$, describing the number of respondents who are going to vote for each of the five parties. Even though none of the 25 people answered party number five, it is unlikely that this party will receive 0% of the votes on election day. We therefore apply Bayesian statistics to obtain a point estimate of the outcome. Because we do not have any prior knowledge about how the parties will do, we choose a flat Dirichlet distribution as a prior, i.e., $\alpha_0 = (1, 1, 1, 1, 1)$. The posterior distribution is then also a Dirichlet distribution with $\alpha' = \alpha_0 + \mathbf{x} = (13, 9, 4, 3, 1)$. The three resulting statistics can be calculated from Eq. 4.4-4.7:

$$\text{E}[\mathbf{x}] = \left(\frac{13}{30}, \frac{9}{30}, \frac{4}{30}, \frac{3}{30}, \frac{1}{30}\right) \approx (0.43, 0.30, 0.13, 0.10, 0.03)$$

$$\text{Mode}[\mathbf{x}] = \left(\frac{12}{25}, \frac{8}{25}, \frac{3}{25}, \frac{2}{25}, \frac{0}{25}\right) = (0.48, 0.32, 0.12, 0.08, 0)$$

$$\begin{aligned}\text{E}_a[\mathbf{x}] &= \mathscr{C}[\exp[\Psi(13)], \exp[\Psi(9)], \exp[\Psi(4)], \exp[\Psi(3)], \exp[\Psi(1)]] \\ &\approx (0.45, 0.31, 0.13, 0.09, 0.02)\end{aligned}$$

The prior, $\alpha_0 = 1$, adds five additional multinomial trials on top of our 25 observed trials. We can be confident in our prior (that all parties are equally likely to receive a vote) in a ratio of 5/25 = 0.2, relative to the information that comes from our observations. ▪

4.3 Other imputation methods

A large number of alternative zero replacement methods exist and it is very difficult to answer the question of which one is better. Remember that we do not know *a priori* whether a value is an essential zero or if it is just below the detection limit. Let us briefly discuss a couple of these strategies.

4.3.1 k-Nearest neighbor replacement

It is possible to replace zeros by adopting a non-zero value for a part of the k samples that are most similar to the sample with a missing value. In order to identify the most similar compositions, we calculate the distance between them using the Aitchison distance, def. 3.8. If the compositions are not closed to the same value, we have to apply closure, and then we use the median value for the missing part from the k-nearest samples.

The result is somewhat dependent on the choice of k. A larger value of k means a greater smoothing of the compositions, i.e., they become overall more similar to each other. With smaller values, however, we risk that the k nearest samples are also missing that same part, in which case the median is still zero and no value can be replaced.

4.3.2 Iterative replacement

If we assume that the missing values are all "below the detection limit"-zeros, we can improve the estimates by using a linear model to iteratively update the missing values. The strategy goes as follows: First, replace zeros using one of the above-mentioned methods, e.g., Bayesian replacement or using k nearest neighbors, and then ILR transform the data set, using a balancing basis where we order the parts after the number of samples in which the part is missing, from high to low. That way, the first balance contains the part with most zeros on one side and all the remaining parts on the other, and so on until all parts have been split. We then need to solve a regression problem,

$$\mathbf{y} = \hat{\alpha}\mathbf{X} + \beta$$

where the vector $\mathbf{y}$ contains the non-zero values of the first left balance and the matrix $\mathbf{X}$ contains the corresponding values from the first right balance. We can solve this using our favorite solver, for instance, a least square fit. The zero values are now replaced by using the regression solution on the right balances for the missing values. Then proceed to the next part with the second-most missing values and do the same, and so on until all the missing values have been replaced. Once all zero values have been replaced with new estimates, we start over again, by re-estimating the replaced values using a new matrix $\mathbf{X}$, which now contains updated values. After a few iterations, the estimates should have converged and should no longer change and we then have our model-based replaced zeros after an inverse ILR-transformation.

Let us consider a simple example of this type of zero replacement. Give a 3-part data set with 4 samples,

	p1	p2	p3
s1	17.02	34.40	48.58
s2	0.00	36.44	63.56
s3	14.98	34.49	50.52
s4	16.14	31.58	52.28

(4.8)

We can see that sample 2 is missing a value in part 1. It is clear that the four compositions are rather identical, except for the missing part. If we assume that the distribution of parts is the same over the four samples, we can see that the missing value should fall somewhere in the range 14-17, and so we *ad hoc* replace the zero with the number 15, so that we can

perform an ILR transformation. The missing value is in the first part, so our balance basis becomes,

	p1	p2	p3
1	1	-1	-1
2	0	1	-1

(4.9)

which should be normalized as usual. Then we apply the ILR transformation to obtain,

	z1	z2
s1	-0.72	-0.24
s2	-0.95	-0.39
s3	-0.84	-0.27
s4	-0.75	-0.36

(4.10)

In this transformed matrix, the value -0.95 corresponds to the missing value, which we initialized to 15. We leave $s2$ out and solve the remaining set of linear equations,

$$\begin{pmatrix} \text{-0.72} \\ \text{-0.84} \\ \text{-0.75} \end{pmatrix} = \alpha \begin{pmatrix} \text{-0.24} \\ \text{-0.27} \\ \text{-0.36} \end{pmatrix} + \beta \tag{4.11}$$

using a least-squares fit. The best fit solution is $\alpha = -0.04$ and $\beta = -0.78$. The missing value can now be estimated to be $-0.39\alpha + \beta = -0.76$. Replacing this value in the ILR matrix and back transforming to the simplex, we get an estimate of the missing value of 15.88.

4.4 Exercises

Exercise 4.1 A latent 10-part composition has a linear probability distribution,

$$\mathbf{x} = [0.18, 0.16, 0.15, 0.13, 0.11, 0.09, 0.07, 0.05, 0.04, 0.02].$$

CLR transform the latent compositions and plot the result. ■

Exercise 4.2 100 multinomial samples are drawn from the latent composition in Exercise 4.1, with 20 trials in each. These can be found in the file `04_exercise_data.csv`. Replace zeros in the samples using different replacement schemes, perform the CLR transform, and plot the mean on top of the latent composition. Observe the effect of the various schemes.

Replace zeros by

- adding a pseudo-count
- using Eq. 4.1
- Bayesian replacement using different concentration parameters
- k nearest neighbors with different k's
- iterative replacement.

In your opinion, what replacement scheme gives the best result? ■

5. Visualizing compositions

An important part of working with data is knowing how to visualize it in a meaningful and informative way. In the case of compositional data, we need to take their compositional nature into account or risk giving erroneous impressions of the data.

5.1 Bar plots

One of the simplest ways of presenting a composition is to just plot the parts as bars in a bar chart. However, while there is nothing wrong with doing that, it is probably the least informative way of presenting compositions. Table 5.1 shows the nutritional content of a number of vegetables, and these data are presented in a simple bar chart in Fig. 5.1. The part named "other" in the table is an amalgamated part containing fibers, minerals, and vitamins, all of which are present in very small amounts. There is no natural ordering, and it is difficult to compare the vegetables that are more similar and the ones that are more different. There is no natural sorting of the samples or the parts in a bar plot like this, so the analyst presenting the data can choose the sorting in a way that facilitates interpretation of the data. This should be done with caution, however, since it is possible to fool the viewer into seeing a trend that may not be supported by the data itself. In a bar plot like Fig. 5.1, we do not take into account that the data are compositional, and consequently, it has limited use. However, it is unfortunately the most common way to see compositional data presented.

By taking the compositional nature of the data into account, we can create a stacked bar chart, which is a considerably better way of presenting the data. In a stacked bar chart, we can easily sort the vegetables on one of the parts, in this case protein, and we can immediately get an idea of how they compare (Fig. 5.2). For both cases, bars and stacked bars, it is absolutely necessary to apply closure with the same constant to all the samples shown in the plot since they need to have the same unit.

In principle, pie charts are a viable alternative to (stacked) bar charts. In most cases,

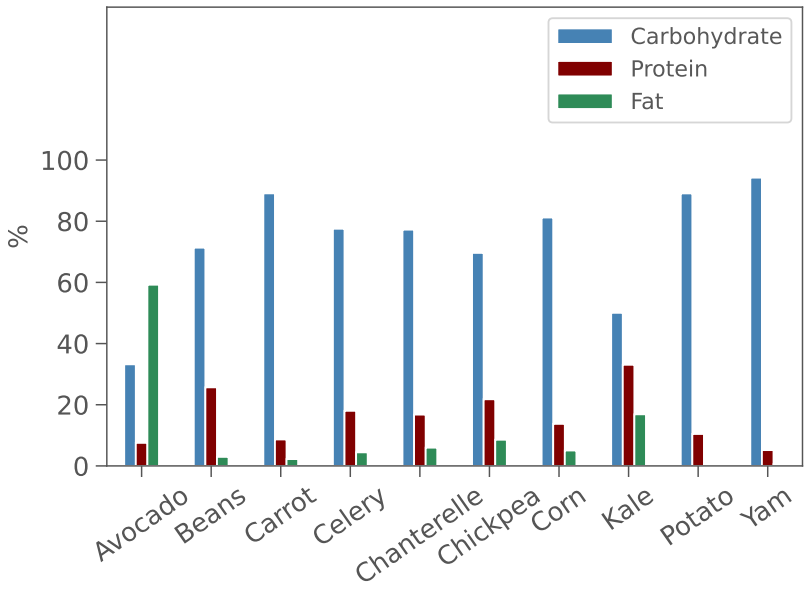

Figure 5.1: Bar chart showing the nutritional content of vegetables.

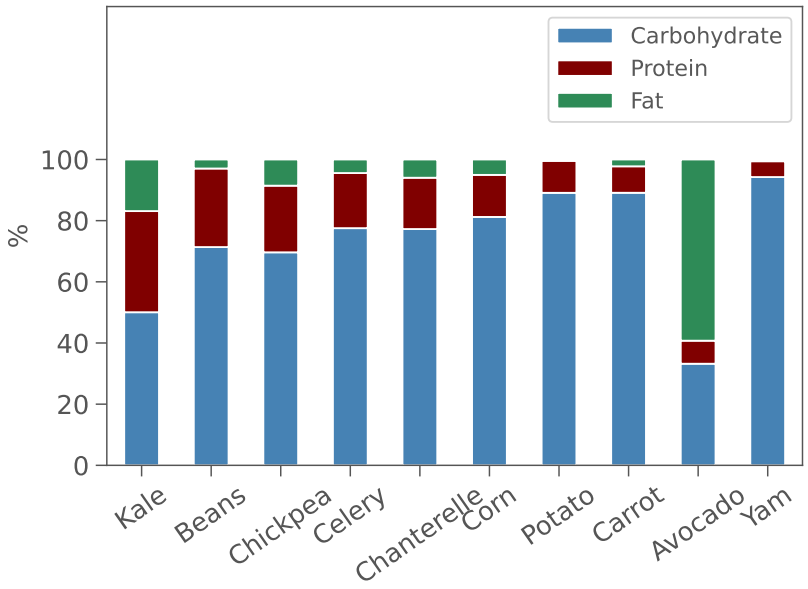

Figure 5.2: Stacked bar chart showing the nutritional content of vegetables.

Vegetable	Carbohydrate	Protein	Fat	Water	Other
Avocado	8.64	1.96	15.41	72.33	1.67
Beans	14.50	5.22	0.60	78.04	1.64
Carrot	9.58	0.93	0.24	88.29	0.96
Celery	2.97	0.69	0.17	95.43	0.74
Chanterelle	6.86	1.49	0.53	89.85	1.27
Chickpea	22.53	7.05	2.77	66.72	0.93
Corn	19.02	3.22	1.18	75.96	0.62
Kale	4.42	2.92	1.49	89.63	1.54
Potato	17.49	2.05	0.09	79.25	1.12
Yam	27.88	1.53	0.17	69.60	0.82

Table 5.1: Nutritional values, shown as percentages, of ten different vegetables (Data from US department of agriculture).

it is a matter of personal preference, but in general, pie charts are harder to read as there are no axis, they take up more space, and they are difficult to compare side by side. Many commercial software packages provide 3D versions of both bar and pie charts. While this may add a fancy look, it does not help to visually interpret the data and should be avoided in all serious applications. Never sacrifice data interpretability for fancy-looking graphics. As Fig. 5.3 shows, it is misleading and oftentimes silly.

A final note on stacked bar plots and pie charts. Because they are, by construction, plotted on a linear scale, it is only possible to discern one or at most two orders of magnitude. This means that very small proportions are just shown as a line and cannot be compared visually to other small proportions. This is why pie charts often have an 'other' category, where all the small parts are amalgamated. Likewise, if there is one dominant part in the composition, for instance, if one part makes up 99.5%, the entire bar (or pie) is just shown in the same color, and very little information can be drawn from the plot. Figure 5.4 shows the composition of the atmospheres of three planets in the solar system. The left panel is a stacked bar plot, and by just looking at that plot, it is extremely difficult to compare the fraction of oxygen on Mars to the fraction of CO_2 on Earth. To overcome this problem of low dynamic range in stacked bar plots, we can choose to plot CLR values instead for a much clearer view. The downside of this approach is, of course, that the reader does not necessarily know what CLR values are, and additional explanation is therefore required when presenting data in this way. The atmospheric compositions are shown as CLR values in the right panel of Fig. 5.4.

5.2 Log-ratio scatter plots

In the case of bar plots and stacked bar plots, it is possible to show compositions with as many parts as we want, although with an increasing number of parts, it becomes harder to extract information. In the special case where the composition has three parts, we have a few other visualization options. Even if our data contains more than three parts, it can often be useful to reduce the dimensionality, either by extracting a sub-composition or by amalgamating parts, so that we only have three parts to visualize.

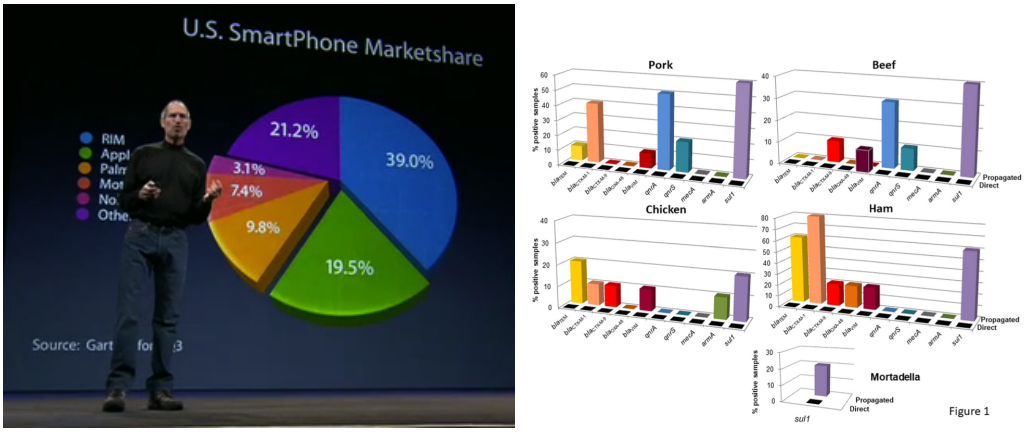

Figure 5.3: Left: Steve Jobs of Apple inc. presenting the iPhone market share. Due to the (unnecessary) 3D perspective, Apples market share of 19.5% appears bigger than the 21.2% share marked 'Other'. Right: Unnecessary use of 3D effects in a bar chart. Again, due to the depths effect, foreground features appear more prominent than background features despite having the same values (from Gómez-Gómez et al., Sci. Rep., 2019, 9, 13281).

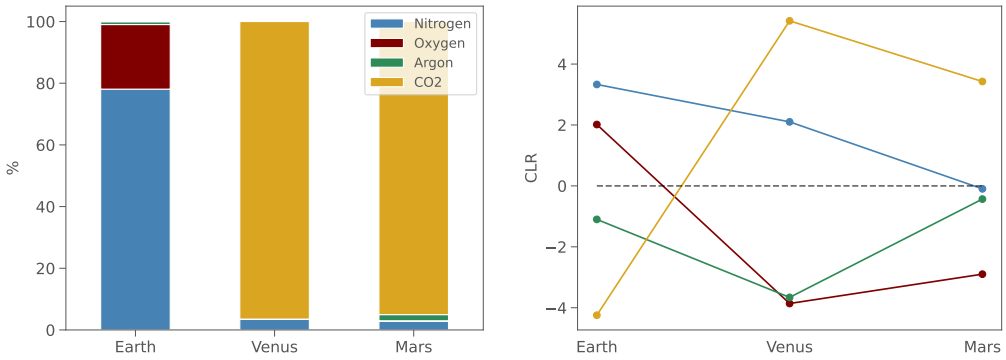

Figure 5.4: A stacked bar plot and categorical CLR values of the same data. It is much easier to compare small proportions in CLR space.

For a three-part composition, we can make ratio scatter plots. Although entirely optional, these plots can be shown on a logarithmic axis, making them log-ratio scatter plots. For a three-part composition, we can form three ratios (six, actually, but if we take the logarithm, only three are unique up to a sign), but of these three, only two are independent. The third ratio can be calculated from the two others,

$$\text{Protein}/\text{Fat} = \frac{\text{Carbohydrate}/\text{Fat}}{\text{Carbohydrate}/\text{Protein}} \tag{5.1}$$

or, by taking the logarithm on both sides,

$$\log(\text{Protein}/\text{Fat}) = \log(\text{Carbohydrate}/\text{Fat}) - \log(\text{Carbohydrate}/\text{Protein}) \tag{5.2}$$

If the two ratios are plotted against each other, as shown in Fig. 5.5, the third ratio is given by the orthogonal projection onto a line with a slope of -1.

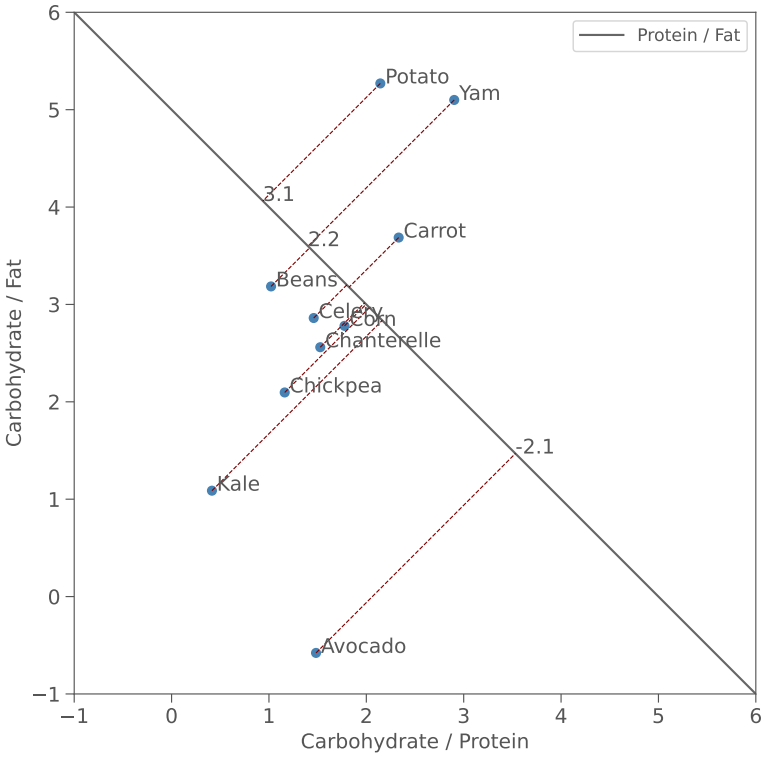

Figure 5.5: A log-ratio scatter plot, showing the vegetable data in two dimensions. In this type of plot, we can easily identify similar vegetables.

5.3 Ternary diagrams

The best way of visualizing a three-part composition is by plotting it in a ternary diagram. The ternary diagram is a direct visual representation of the simplex and is often referred to as a simplex plot. Unfortunately, ternary plots can only show three-part compositions, like in the case of the compositional scatter plot.

Figure 5.6, left panel, shows the vegetable nutrition data in a ternary diagram. It is easier to spot outlying points and, in general, identify samples that group together, which is very difficult to do by looking at the bar plots. The parts are read off the axis by following the grid lines that are parallel to the tick marks on the axis. The axes themselves and the triangle vertices are not part of the diagram since we do not allow parts to be zero.

In Chap. 3 we discussed the two arithmetic operations that make the simplex a vector space: perturbation and powering (Def. 3.2.1). Recall how these operations are analogous to addition and scalar multiplication in real vector spaces. By plotting the compositions in a ternary diagram, we can directly visualize the effects of perturbation and powering. An example is shown in Fig. 5.7. It is clear from this figure that perturbation shifts the points (translate them) while powering scales them, and thus we can visually interpret these operations as adding a vector and multiplying by a constant, respectively.

With perturbation and powering, we can define linear transformations,

$$\mathbf{y} = (\alpha \odot \mathbf{x}) \oplus \mathbf{x}_0,$$

which is the compositional equivalent to the well known $\mathbf{y} = a\mathbf{x} + b$ for real vectors.

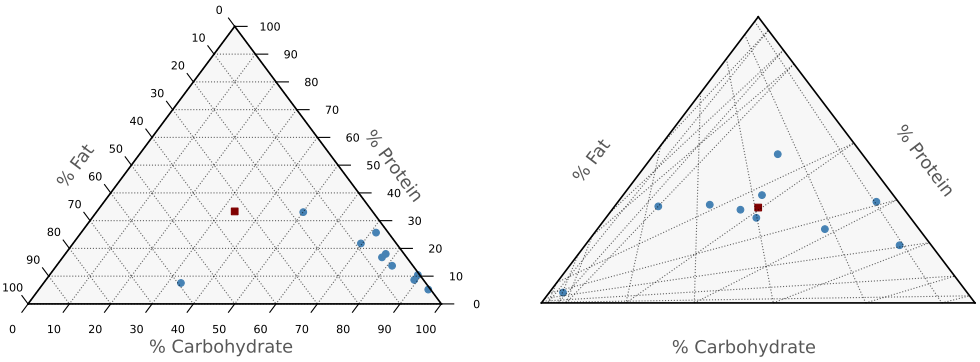

Figure 5.6: Left) Vegetable data plotted in a ternary diagram. Right) The same after centering of the data. The red square marks the barycenter of the triangle, which coincides with the neutral element of the simplex.

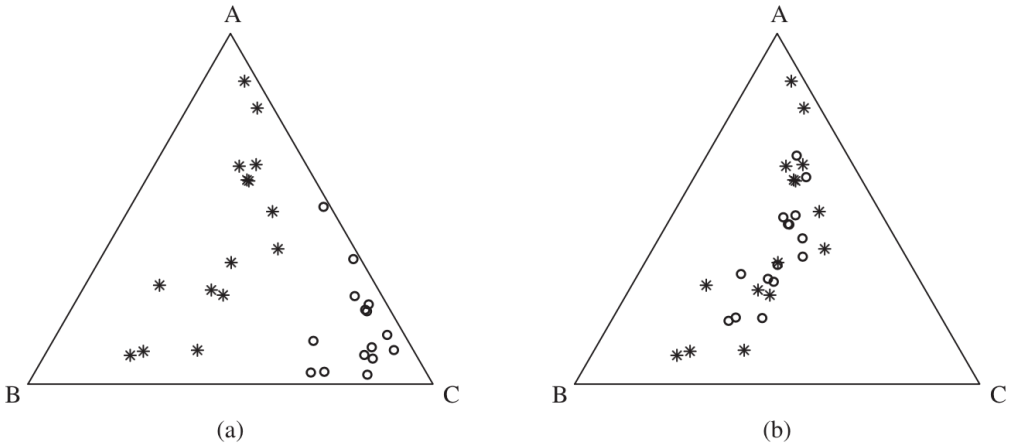

Figure 5.7: a) perturbation of a set of compositions by another composition. b) powering of a set of compositions by a constant.

Choosing a set of $\mathbf{x}_0$ and α which are equidistant in Aitchison distance, we can plot compositional lines in a ternary diagram, as shown in Fig. 5.8. These lines are the compositional equivalent of an orthogonal grid in real space. It can be shown that the Aitchison inner product between the leading vectors of intersecting lines is 0, so in a general way, the dashed and full lines in Fig. 5.8 are orthogonal.

5.3.1 Centering

Sometimes, a set of samples plotted in a ternary diagram will fall very close to an edge or a vertex, making it difficult to see the actual distribution of points. There are two ways we can deal with this problem. We can either cut out the area containing the points and magnify this part, or we can center the data.

In order to center the data, we need to apply a perturbation to the composition, which is the equivalent of translation in real space. If we perturb a composition by its inverse,

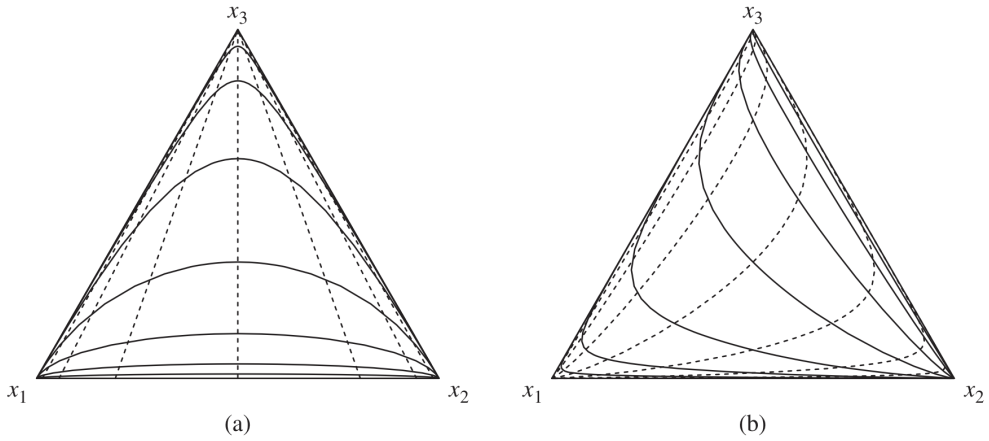

Figure 5.8: Orthogonal grids in the 3-simplex. The grid in b) is rotated by 45° with respect to the grid in a).

we obtain the neutral element according to the additive inverse law (Eq. 3.9). In the Aitchison geometry, the neutral element is the closure of a vector of 1's, equal to $\mathbf{0} = (1/D, 1/D, ..., 1/D)$. In the ternary plot, the neutral element coincides with the barycenter of the triangle, also known as its centroid. The centroid is the point of the intersection of the three lines that connect each vertex to the midpoint of the opposite edge. This means that if we perturb the samples by their inverse, we move the points to the centroid of the triangle. If we instead perturb each sample by the inverse of the geometric mean of all the samples, we find that the set of samples, after perturbation, will gravitate around the centroid. Their new geometric mean will coincide with the triangle centroid. The data has been centered. This makes it a lot easier to visually explore the distribution of the data in the simplex, as long as we remember to take into account that the axes have been centered as well. The right-hand side panel of Fig. 5.6 shows the vegetable data after centering.

5.3.2 Coordinate representation in ILR space

Finally, we can move out of the simplex by making a coordinate transformation into real Euclidean space. In this case, we need to make use of the ILR transformation because we need coordinates (recall that CLR values are coefficients and not coordinates) that can be represented in a coordinate system. ALR does also provide coordinates, but the ALR transform is not isometric (meaning it does not conserve distances), and this is also a requirement for proper visualization. Only ILR fulfills both of these requirements.

Figure 5.9 shows the vegetable nutritional data in the ternary diagram and in ILR coordinate space. We have used the sequential binary partition table to make the transformation,

x_1	x_2	x_3
1	1	-1
1	-1	0

which then needs to be normalized. The axes in the plot are unit-less and, as we discussed in Chap. 3, are defined by the balances of the contrast matrix. Euclidean distances are conserved, which means that two samples that are twice as far apart as two other samples

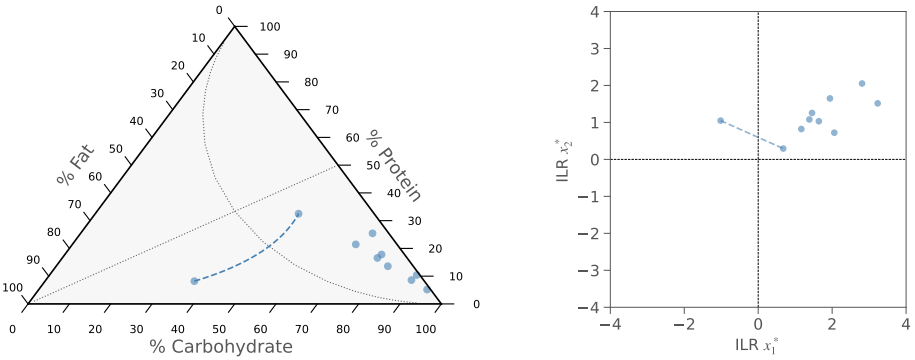

Figure 5.9: Left: The vegetable nutrition data in a ternary diagram. Right: The same data is represented in ILR space. The dotted line segment represents the straight line connecting avocado and kale.

are really twice as different. This can be utilized in the following way: if we want to know which vegetable is nutritionally most similar to kale, we can identify kale in this plot and measure the Euclidean distance between kale and the nine other vegetables. The vegetable with the shortest distance to kale is the one with the closest matching nutritional parameters. It turns out that avocado is the closest match to kale, and we have for illustrative purposes connected the two with a straight line in Fig. 5.9. The dotted lines in the ILR coordinate plot, both the blue and the black lines, have been inversely transformed and plotted in the ternary diagram as well, to illustrate how line segments behave in the ternary diagram. In fact, if we plotted the centered and uncentered data points and connected the two sets with trajectories, they would show up as straight lines in the ILR coordinate plot.

5.3.3 Line segments in ternary diagrams

As we have just seen above, Aitchison straight lines appear as curves in a ternary diagram. Plotting curves in ternary diagrams can be useful when working with time-dependent compositions (see Chapter 9), but it can be a little tricky to get right. The best way is to discretize the curve into a series of points and then make a piecewise linear approximation by connecting the dots. Arbitrary smoothness can be achieved by increasing the number of points, but for all practical purposes, 100 points are sufficient.

Using this technique, we can plot various shapes and see how they appear in the simplex. Figure 5.10 shows three examples. The top row shows an example of perturbation along a line segment. In ILR space, the solid line has been translated by the vector corresponding to the dotted line by adding the two vectors. In the simplex, this becomes perturbation. Notice that, while in ILR space it is clear that the dotted line was used, the same is not obvious in the simplex. Line segment lengths are not constant in the simplex! The middle row shows an example of powering (scaling in ILR space). A vector $(1, 0.5)$ has been scalar multiplied by a range of integers to form a continuous line segment in ILR space. When translated into the simplex, the operation becomes powering. Notice again how the position of the power integers is not equidistant along the line segment. The last row shows a collection of circles and ellipses in real space and their corresponding shapes in the simplex. The closer we are to the origin of the real space coordinate system, the more

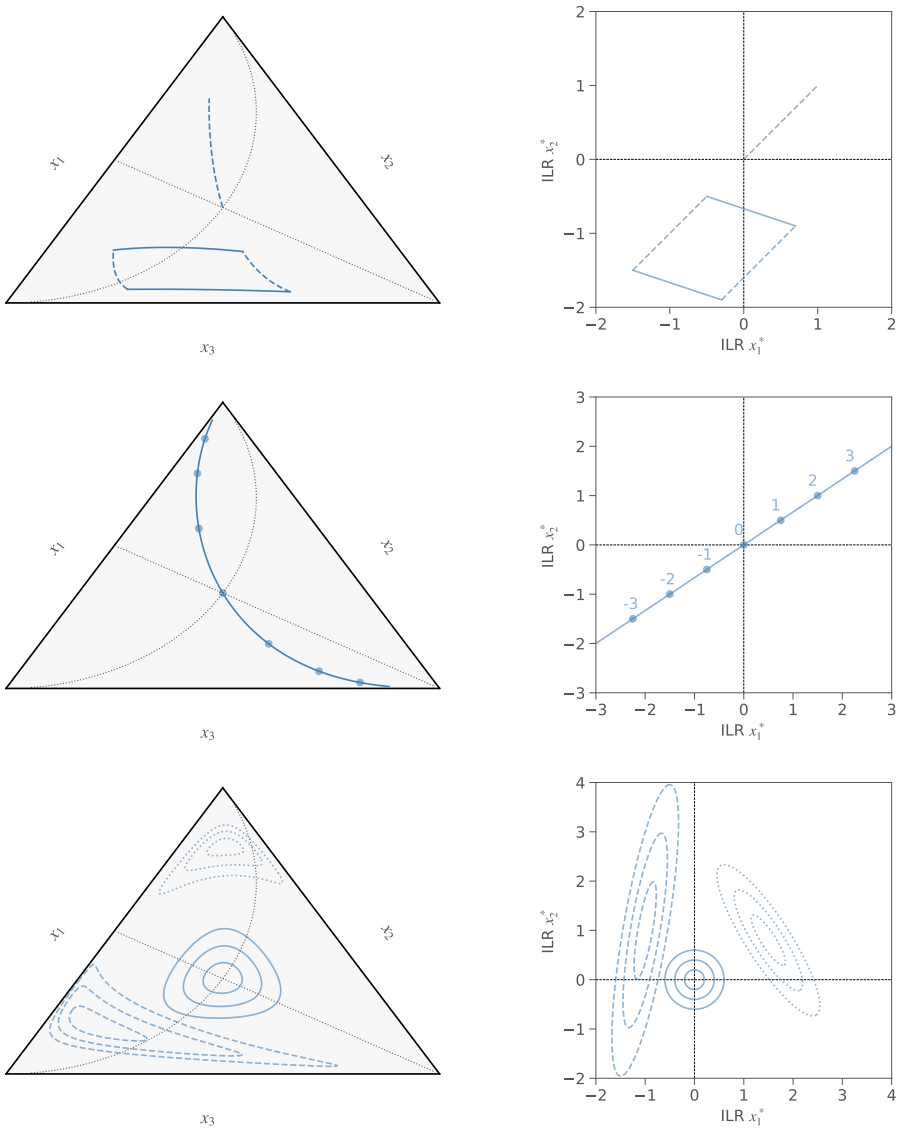

Figure 5.10: Top row) Line segments in the ternary plot and in ILR space. The solid line segment has been perturbed by the dotted line segment in the simplex, corresponding to a translation in real space. Middle row) A composition powered by a range of integers in the simplex and in ILR space. Bottom row) Circles and ellipses in ILR space and in the simplex.

similar the shape appears in the simplex, while the further we get from the origin, the more warped the transformed shape is, to the point where they are no longer recognizable as ellipses. This gives us a geometric understanding of why real vector algebra does not apply to compositions, and it also reveals that, sufficiently close to the origin (or centroid), real multivariate algebra gives approximately the correct result on the simplex.

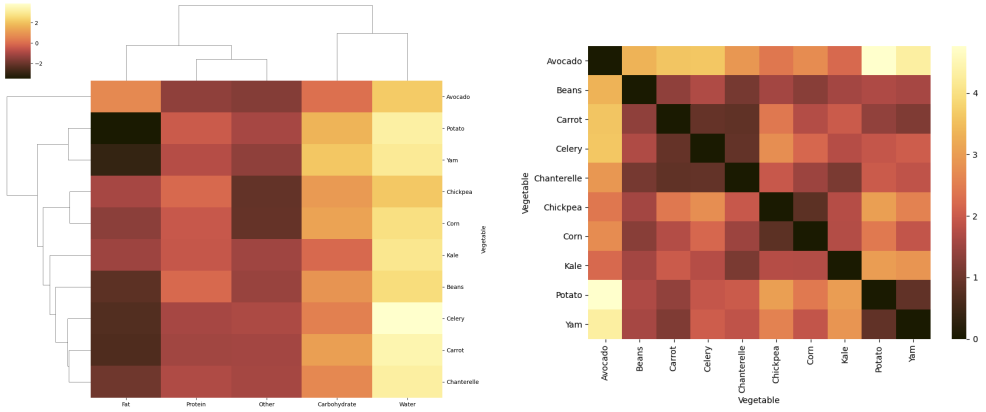

Figure 5.11: Left panel shows a cluster map of the vegetable data. Right panel shows a heatmap of the distance matrix. Darker colors mean they are closer together.

5.4 Heatmaps and cluster maps

The final visualization techniques that we will discuss in this chapter are the so-called heatmaps and the related cluster map. In all simplicity, heatmaps can be thought of as 2D versions of the ordinary bar chart, and they are often more confusing than helpful. In a heatmap, one cannot see the heights of the bars, so we assign a color that represents the height of a bar. A color palette is therefore needed, and choosing a proper palette is crucial for the information value of a heatmap. The details of choosing a good palette are beyond the present scope, but an interesting discussion on the topic can be found in Crameri, Shephard & Heron, Nature comm., 2020, 11, 5444. Heatmaps also suffer from the same dynamical range problem that we saw in the example with planetary atmospheres above. Again, this can be solved by transforming the data into CLR space, where a large dynamic range is shown on a linear scale.

A heatmap has no particular ordering of features or samples, but a cluster map does. The samples and/or the features can be linked and rearranged to form hierarchical clusters, which results in a heatmap where samples (or features) that are more similar, i.e., have a smaller Aitchison distance from each other, are grouped together. Various methods exist for calculating the linkage, but since the data has already been CLR transformed, the Euclidean metric seems an obvious choice. Cluster maps are often plotted with corresponding dendrograms along the axis to show the hierarchical clustering.

Other data set specific properties can be shown in a heatmap. The distance matrix, for instance, which is a symmetric matrix with zeros in the diagonal. It contains the Aichison distance between each pair of samples (or, if the data set is transposed, each pair of features), and this is sometimes plotted in a heatmap as well. Likewise, the variation matrix, which we will introduce in the following chapter, is ideally visualized in a heatmap. It is important to stress here that derived data products, such as the distance or variation matrices, are not suitable for cluster maps since calculating the distance between column or row vectors in these matrices is meaningless. Figure 5.11 shows a cluster map and a heatmap of the distance matrix of the full vegetable data set.

5.5 Exercises

In the following exercises, we will once again use the data from Exercise 2.3. For convenience, the data is given here as well,

	1	2	3	4	5
x_1	79.07	31.74	18.61	49.51	29.22
x_2	12.83	56.69	72.05	15.11	52.36
x_3	8.10	11.57	9.34	35.38	18.42

Exercise 5.1 Make a bar chart and a stacked bar chart of the data. ∎

Exercise 5.2 Make a log-ratio scatter plot of the data. ∎

Exercise 5.3 Make a ternary diagram of the data. ∎

Exercise 5.4 Build a basis for the data, ILR transform the data, and plot them in a cartesian coordinate system. ∎

Exercise 5.5 Perturb the data by $\mathbf{s} = [0.1, 0.1, 0.8]$ and plot the initial and the perturbed data set in a ternary diagram and in ILR coordinates. In each case, join each pair of samples (unperturbed and perturbed) by a line segment. Observe the effect of perturbation. ∎

Exercise 5.6 Apply powering with α ranging from -8 to $+8$ in steps of 1 to the composition $\mathbf{t} = [0.7, 0.5, 0.8]$ and plot the resulting set of compositions in a ternary diagram and in ILR coordinates. Observe the effect of powering. ∎

6. Exploratory data analysis I

In the following chapters, we will focus on exploratory analysis of compositional data. In general, exploratory data analysis involves searching for errors and outliers in the data set, looking for patterns, and reporting descriptive statistics. In this chapter, we will focus on the principal component analysis method and the results that can be derived from it. In the following lectures, we will look at some more advanced approaches.

We will consider a data set represented by a matrix $\mathbf{X}$, with n rows (samples) and D columns (parts). The data consists of food consumption in 25 European countries in the early 1980s, broken down into several categories. The values in each category are the percentage of protein provided by the category. The categories do not add up to 100 percent, since some fraction of the protein consumption is provided by food items that do not fall into any of the reported categories. The data is presented in table 6.1.

Along with the protein sources, there are also two descriptive variables listed in the table: EW and NS. These are not part of the compositions but rather two categorical variables that, somewhat arbitrarily, describe the location of the country within Europe: eastern or western Europe (E:1, W:2) and northern or southern Europe (N:1, S:2). This data set is from a time when the economic situation was vastly different between east and west, and the goal of this example is to see if this is reflected in the nutritional data.

6.1 Descriptive statistics

For standard real-value data, it would be normal procedure to calculate the (arithmetic) mean and the standard deviation (or variance) in order to describe the central trend and the sample dispersion in the data set. For compositional data, these properties, however, have no meaning, since they rely on Euclidean geometry, which we have seen in Chapter 3, which is not appropriate for compositional data. We need to make use of the geometric mean to describe the sample center and the compositional variation matrix to describe the dispersion.

▪ **Example 6.1 — Arithmetic versus geometric means.** Suppose we buy shares in a company for the price of 100 DKK and keep them for two years before we sell them again. In the first year, the value increased by 90%, while in the second year, the value dropped by 90%. What is our mean return for the two years? If we simply use the arithmetic mean, we find that the net return is 0%, suggesting that we can sell the shares for the price at which we bought them. This is incorrect. After the first year, our shares have the value 100 DKK $\times$ 1.9 = 190 DKK and after the second, 190 DKK $\times$ $(1-0.9)$ = 19 DKK, which is, we have lost 81 DKK in total when we sell. If we calculate the geometric mean, we get $\sqrt{1.9 \times 0.1} = 0.436 = 43.6\%$. We can check that this is correct: after the first year, the average value is 100 DKK $\times$ 0.436 = 43.6 DKK and after the second year 43.6 DKK $\times$ 0.436 = 19 DKK. ▪

The geometric mean is closely related to the arithmetic mean through logarithms, because the logarithm of the geometric mean of x_i equals the arithmetic mean of the log of x_i.

Definition 6.1.1 — Sample center. The sample center for a set of compositional samples is the closed composition of geometric means of parts. Given n samples with D parts each

$$\text{cen}[\mathbf{X}] = \mathscr{C}\left[\hat{g}_1, \hat{g}_2, ..., \hat{g}_D\right], \tag{6.1}$$

where

$$\hat{g}_j = \left(\prod_{i=1}^{n} x_{i,j}\right)^{1/n}, \qquad j = 1, 2, ..., D \tag{6.2}$$

is the geometric mean of the j'th part.

It is important to note that for the sample center, the geometric mean is considered by parts (column), as opposed to the geometric mean used in the CLR-transformation, where it is considered by sample (row).

Definition 6.1.2 — Variation matrix. The dispersion in the log-ratio of parts is given by the variation matrix,

$$\mathbf{T} = [t_{ij}], \qquad t_{ij} = \text{var}\left(\ln \frac{x_i}{x_j}\right), \qquad \text{var}(\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2 \tag{6.3}$$

and the total variance,

$$\text{totvar}[\mathbf{T}] = \frac{1}{2D}\sum_{i,j=1}^{D} t_{ij} \tag{6.4}$$

The variation matrix is always symmetric around the diagonal, and the diagonal elements are always zero. The sample center and variation matrix of the protein consumption data is shown in table 6.2. From the sample center, we can see which food items are the biggest contributors of protein in Europe. Cereals are the main contributor with almost $\sim$40% followed by dairy products at $\sim$20%.

The log-ratios with the smallest variability are red meat to milk and white meat to eggs. Having small variability means that the parts are almost proportional, which in this case can be explained by the fact that producing red meat (cows) also results in milk, while producing white meat (chicken) also results in eggs. The largest contributors to variation are the log-ratios involving fish and nuts, which means that there is little correlation across Europe in food consumption involving these food sources.

Finally, we can use the sample center and the total variance to center and rescale our data set for further analysis. This is done in order to bring all the parts to the same scale. As we saw in Chapter 5, centering and scaling the data preserved the metric properties.

6.2 Principal component analysis

The next step in exploring our data is to do principal component analysis (PCA). Principal components are displayed in a biplot, which is also sometimes referred to as a PCA plot. Principal components are the singular value decomposition (SVD) of the data after centering and scaling the variance to 1. SVD algorithms, particularly the ones that are implemented in modern programming languages like R and Python, rely on real Euclidean vector space algebra and are therefore not appropriate for compositional data. However, CLR coefficients preserve the metric properties of the data and obey Euclidean geometry, and they are therefore ideal for PCA on compositional data.

SVD results in two sets of eigenvectors and a set of eigenvalues. The row eigenvectors form an orthonormal basis for the dataset. These were denoted $\mathbf{\Psi}$ in Chapter 3 and can be used to obtain the ILR-coordinates when multiplied with the CLR-coefficients. In the PCA framework, they are known as *loadings*. The column eigenvectors are known as *scores*. When the PCA is performed on the (centered) CLR coefficient matrix, the diagonal of the score matrix is exactly the ILR coordinates in the basis spanned by the loadings. The eigenvalues are proportional to the sample variance of these coordinates.

The biplot is drawn by plotting the two first components of the eigenvectors. Typically, the D loadings are plotted as vectors, often called *rays*, while the n scores are plotted as points called *markers*. The length of the rays is proportional to the standard deviation of the CLR-coefficients. Longer rays imply greater variance in that part. The interpretation of the PCA is that it reveals the internal structure of the data in a way that best explains the variance of the data. It is a projection of the data matrix onto a two-dimensional space, viewed from the most informative viewpoint.

As mentioned above, SVD is a generalization of the eigendecomposition of a square matrix to a general $m \times n$-matrix. We need this generalization because we do not *a priori* expect to have the same number of samples and parts. The limitation is that we only get as many eigenvalues as we have columns (samples), and we therefore cannot display more parts in the biplot than the number of samples that we have. If we have more parts than samples, we need to get rid of some of the parts before we can do PCA, typically by extracting a sub-composition or by amalgamation. If we have no other way to prioritize the parts, we want to get rid of the parts that contribute the least to the variation matrix.

6.2.1 Scree plots

The amount of variation that is explained in a biplot is a measure of how well the data is represented by the biplot. The fraction of retained variability is the sum of the first two

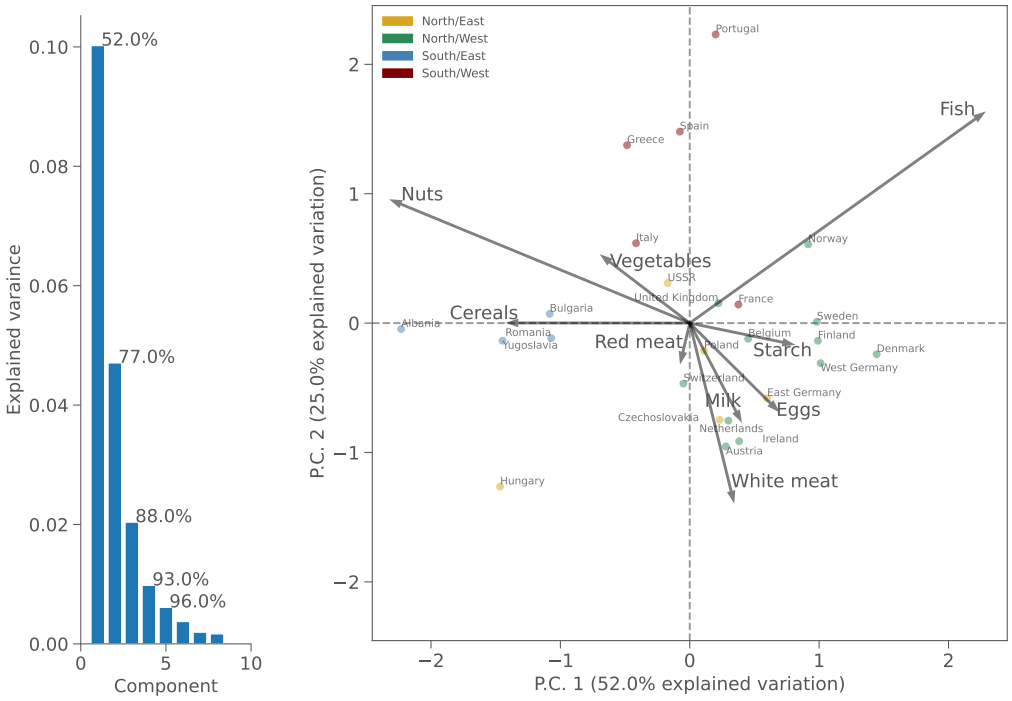

Figure 6.1: Scree plot and biplot of the protein data set.

eigenvalues, divided by the sum of all the eigenvalues. If this ratio is low, then the biplot does not provide a particularly good view of the data set. In order to explore the explained variance per dimension, we can draw a so-called Scree plot. Ideally, we would like to see that the majority of the variance is explained by the first two components. Figure 6.1 shows the Scree plot for the protein data set. We can see that the first two components explain 77% of the total variation in the data set. This means that almost 25% of the variation in the data set is not shown in the biplot.

6.2.2 Interpretation of biplots

The actual biplot is shown next to the Scree plot in Fig. 6.1. PCA biplots are incredibly rich and dense in information. The origin of the biplot (0,0) represents the geometric mean of the parts. All loadings and scores are shown relative to the mean (because we centered the data prior to the CLR transformation). The length of each ray is proportional to the standard deviation of that part, so long rays mean a large variance and vice versa for short rays. Moreover, the line segment that connects two rays, called links, is proportional to the standard deviation of the log-ratio of the two parts. The longer the links (i.e., the rays are further apart), the larger the variance in the log-ratio of those parts. Consequently, short links correspond to low variance in the log-ratio, which means that clusters of rays imply that the parts are correlated. In the protein biplot, the fish and the nut rays are the longest, which also means that all links involving fish and nuts, and in particular the link between fish and nuts, are long. This matches the result we found when we looked at the variation matrix: log-ratios involving fish and nuts have the greatest variance. It should be noted that the length of a ray is actually the link between that part and the origin, which is the

geometric center, so the interpretation of the length of a ray is actually the variance in the log-ratio between the part and the geometric mean. Since the geometric mean will change, in general, when a subcomposition is considered, that log-ratio, and hence the variance in it, will also change. Therefore, drawing conclusions on the length of a single ray is only meaningful when considering the full composition.

Angles between links provide information about the correlation of subcompositions. The cosine of the angle between two links is proportional to the correlation coefficient between the log-ratios. A small angle means a correlation coefficient approaching 1, whereas perpendicular links result in a correlation coefficient of zero, that is, they are completely uncorrelated.

Markers are also plotted in the biplot, and they represent the samples in the data set. Markers are typically drawn as points or symbols, and they can be colored according to a predetermined grouping of the samples. That could, for instance, be the country of sample origin, male or female for human samples, sample year, etc. In cases where there are many samples, it may be useful to plot the group centroid instead (which is possible because the metric properties are preserved under the CLR transformation) as well as covariance error ellipses or other indicators of sample dispersion in the group. Cluster analysis can be performed, e.g., with k-means, in order to detect samples that are more alike and have more similar compositions. Again, due to the conservation of the metric properties, markers that are closer together are more similar.

Markers can be projected onto the rays. The closer the point along the ray where the projection falls is to the origin, the closer the part represented by the ray for that particular sample will be to the geometric mean of the set. Marker projections that fall near the end of a ray will have a CLR value for that part, which is about one standard deviation larger than the set average. The further beyond the tip of a ray a projection falls, the more extreme is the CLR value for that part of that sample. The opposite is true if the projection falls on the reverse side of the origin.

Likewise, markers can be projected onto links. If the projection of the marker onto a link coincides with the projection of the origin onto that link, then the sample has a log-ratio between those two parts that is equal to the sample set average. If the projection of the marker falls one link length away from the point of the projection of the origin, then that sample has a log-ratio of those two parts that is one standard deviation larger than the set average (or smaller, depending on which part is the denominator in the log-ratio).

6.2.3 Subcompositional analysis

We can explore the data further by looking at subcompositions and looking for three-part subcompositions with a one-dimensional pattern, that is, constant log-ratios. We are looking for two aligned vertices and one that is perpendicular within the biplot. Once a promising three-part subcomposition has been identified, we re-close it, re-center it, CLR transform it, and calculate eigenvectors and values.

If, for instance, we pick the three-part subcomposition white meat, cereal, and nuts, we can close it to 100, center it by perturbing by the inverse mean, do a CLR transform, and calculate the loadings and eigenvalues using SVD. By doing so, we get,

	Cereals	Nuts	White meat	λ_i
PC.1	-0.554	-3.076	3.630	4.79
PC.2	1.158	-0.723	-0.436	1.43

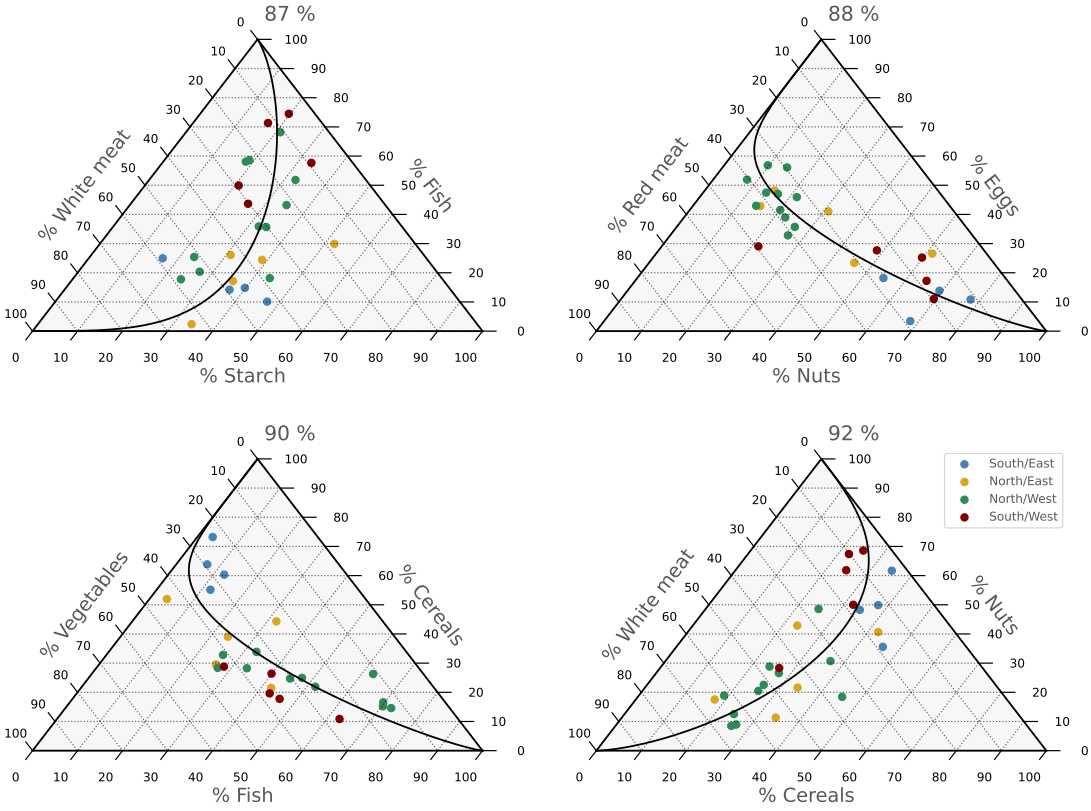

Figure 6.2: Ternary diagrams of various 3-part sub-compositions. The percentage at the top of each diagram is the explained variance along the first principal component.

The explained variance is $\lambda_i^2 / \sum \lambda_i^2 = (0.92, 0.08)$, that is, 92% of the explained variance along the first principal component. We can proceed to plot the subcomposition in a ternary diagram, along with the compositional line $\mathbf{y} = (\alpha \odot e^{\text{PC.1}}) \oplus \mathbf{g}_m$, where α is a scaling parameter, PC.1 is the first principal component, and $\mathbf{g}_m$ is the geometrical mean of the samples. The result of four selected subcompositions is plotted in Fig. 6.2.

In each of the four cases, around 90% of the variation is explained by the first principal component. This is obvious when looking at the ternary plots, where it is seen that the points are distributed rather tightly along the black lines. The deviation from the black lines is the remaining 10% of the variance, and because we have relatively little variation along the second principal component, the *balance* which makes up PC.2, given by,

$$\begin{aligned} PC.2 &= 1.158 \log(\text{Cereals}) - 0.723 \log(\text{Nuts}) - 0.436 \log(\text{White meat}) \\ &= \log \frac{\text{Cereals}^{1.158}}{\text{Nuts}^{0.723} \text{White meat}^{0.436}}, \end{aligned}$$

is close to constant across the samples. This means that cereals to a certain power are highly correlated to the product of nuts to a certain power and white meat to a certain power. The interpretation is that cereal consumption is balanced by nut and white meat consumption; some countries will consume more nuts, others more white meat, and always more white meat than nuts, but the product will be more or less constant relative to cereals.

Likewise, vegetable consumption is balanced by varying proportions of fish and cereals. In the vegetable-cereals-fish ternary plot, we see that the blue points (south/east) and some of the green points (north/west) falls above the PC.1 line, which means that they all consume a little less vegetables relative to fish and cereals, but the blue countries balance the ratio with more cereals (or less fish) and the green countries with more fish (or less cereals).

This type of analysis can be used to break down high-dimensional and complex data sets to look for trends within subcompositions, which may then be pieced together to obtain an understanding of the full data set. Thus, we can conclude that the western Mediterranean countries have a high relative consumption of fish, nuts, vegetables, and cereals, but mainly fish. Scandinavian countries are characterized by a high relative consumption of starch, eggs, and milk. Eastern block countries are high in nuts and vegetables and low in meat and fish. The USSR, France, and UK are very close to the European average, probably mainly because of their large population sizes.

6.3 Exercises

For the exercises in this chapter, we will once again make use of the set of three-part compositions that we also used for the Exercise 2.3 in Chapter 2. This time, we have added more samples. For convenience, the data can also be found in the file `06_exercises_data.csv`.

	1	2	3	4	5	6	7	8	9	10
x_1	79.07	31.74	18.61	49.51	29.22	21.99	11.74	24.47	5.14	15.54
x_2	12.83	56.69	72.05	15.11	52.36	59.91	65.04	52.53	38.39	57.34
x_3	8.10	11.57	9.34	35.38	18.42	18.10	23.22	23.00	56.47	27.11
	11	12	13	14	15	16	17	18	19	20
x_1	57.17	52.25	77.40	10.54	46.14	16.29	32.27	40.73	49.29	61.49
x_2	3.81	23.73	9.13	20.34	15.97	69.18	36.20	47.41	42.74	7.63
x_3	39.02	24.02	13.47	69.12	37.89	14.53	31.53	11.86	7.97	30.88

Exercise 6.1 Compute the geometric center, the variation matrix, and the total variance of the data set. ■

Exercise 6.2 Perturb the data with the inverse of the geometric center. Compute the center, variation matrix, and total variation of the perturbed data. ■

Exercise 6.3 Make a biplot of the perturbed (centered) data by following these steps:

- Calculate the CLR transform.
- Calculate eigenvectors and eigenvalues of the CLR transformed data[a].
- Plot the first 2 principal components of the 3 loadings in a Cartesian coordinate system as arrows.
- Plot the first 2 principal components of the 20 scores in the same coordinate system as points.

■

[a] Eigenvectors and eigenvalues can be calculated using Singular Value Decomposition (SVD), which is a standard package in most programming languages. In Python, SVD can be done using numpy.linalg.svd(). This function returns 3 matrices: the scores (row eigenvectors), eigenvalues, and loadings (column eigenvectors). To bring scores and loadings onto the same scale, scale the loadings by `np.inner(eigvalues*np.identity,loadings.T)`.

Country	Red meat	White meat	Eggs	Milk	Fish	Cereals	Starch	Nuts	Vegetables	EW	NS
Albania	10.1	1.4	0.5	8.9	0.2	42.3	0.6	5.5	1.7	1	2
Austria	8.9	14.0	4.3	19.9	2.1	28.0	3.6	1.3	4.3	2	1
Belgium	13.5	9.3	4.1	17.5	4.5	26.6	5.7	2.1	4.0	2	1
Bulgaria	7.8	6.0	1.6	8.3	1.2	56.7	1.1	3.7	4.2	1	2
Czechoslovakia	9.7	11.4	2.8	12.5	2.0	34.3	5.0	1.1	4.0	1	1
Denmark	10.6	10.8	3.7	25.0	9.9	21.9	4.8	0.7	2.4	2	1
West Germany	8.4	11.6	3.7	11.1	5.4	24.6	6.5	0.8	3.6	2	1
Finland	9.5	4.9	2.7	33.7	5.8	26.3	5.1	1.0	1.4	2	1
France	18.0	9.9	3.3	19.5	5.7	28.1	4.8	2.4	6.5	2	2
Greece	10.2	3.0	2.8	17.6	5.9	41.7	2.2	7.8	6.5	2	2
Hungary	5.3	12.4	2.9	9.7	0.3	40.1	4.0	5.4	4.2	1	1
Ireland	13.9	10.0	4.7	25.8	2.2	24.0	6.2	1.6	2.9	2	1
Italy	9.0	5.1	2.9	13.7	3.4	36.8	2.1	4.3	6.7	2	2
Netherlands	9.5	13.6	3.6	23.4	2.5	22.4	4.2	1.8	3.7	2	1
Norway	9.4	4.7	2.7	23.3	9.7	23.0	4.6	1.6	2.7	2	1
Poland	6.9	10.2	2.7	19.3	3.0	36.1	5.9	2.0	6.6	1	1
Portugal	6.2	3.7	1.1	4.9	14.2	27.0	5.9	4.7	7.9	2	2
Romania	6.2	6.3	1.5	11.1	1.0	49.6	3.1	5.3	2.8	1	2
Spain	7.1	3.4	3.1	8.6	7.0	29.2	5.7	5.9	7.2	2	2
Sweden	9.9	7.8	3.5	24.7	7.5	19.5	3.7	1.4	2.0	2	1
Switzerland	13.1	10.1	3.1	23.8	2.3	25.6	2.8	2.4	4.9	2	1
United Kingdom	17.4	5.7	4.7	20.6	4.3	24.3	4.7	3.4	3.3	2	1
USSR	9.3	4.6	2.1	16.6	3.0	43.6	6.4	3.4	2.9	1	1
East Germany	11.4	12.5	4.1	18.8	3.4	18.6	5.2	1.5	3.8	1	1
Yugoslavia	4.4	5.0	1.2	9.5	0.6	55.9	3.0	5.7	3.2	1	2

Table 6.1: Protein consumption in Europe

Statistics	Red meat	White meat	Eggs	Milk	Fish	Cereals	Starch	Nuts	Vegetables
Sample center	11.93	8.85	3.4	19.94	3.71	39.28	4.89	3.17	4.82
Variation									
Red meat		0.3651	0.2064	0.1354	0.9395	0.3208	0.3851	0.7333	0.3346
White meat			0.1700	0.3198	1.2631	0.5586	0.3001	1.2555	0.4559
Eggs				0.1684	0.8052	0.5679	0.2156	1.1074	0.3936
Milk					0.9611	0.4691	0.3702	1.0476	0.5627
Fish						1.5603	0.6945	2.0827	1.0471
Cereals							0.5937	0.2759	0.2611
Starch								1.1220	0.4313
Nuts									0.4327

Table 6.2: Sample center and off-diagonal compositional covariance matrix.

7. Exploratory data analysis II

7.1 Exploratory analysis of coordinates

In certain cases, a sequential binary partition can be quite informative in itself. Sometimes it is possible to form a meaningful partition based on knowledge of the data, while sometimes such partitions arise from the type of data exploration presented in the previous chapter. In either case, analyzing the coordinates may provide further insights.

7.1.1 Correlation analysis

Because the coordinates of the basis are orthogonal and real vectors, we can apply normal real multivariate analysis to the ILR transformed data as discussed in Chapter 3. For instance, given a composition $\mathbf{x}$ and an orthonormal basis $\mathbf{\Psi}$,

$$\text{ilr}(\mathbf{x}) = \text{clr}(\mathbf{x}) \cdot \mathbf{\Psi}^T, \tag{7.1}$$

gives the logratio-coordinates in the basis $\mathbf{\Psi}$. Recall from Chapter 1 how the standard Pearson correlation coefficients,

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sqrt{\text{var}(X)\text{var}(Y)}}, \tag{7.2}$$

gives a negative correlation bias when applied to compositional data. This bias can be expressed by the relation,

$$\text{cov}(x_1,x_2) + \text{cov}(x_1,x_3) + \ldots + \text{cov}(x_1,x_D) = -\text{var}(x_1), \tag{7.3}$$

where $\text{cov}(\cdot,\cdot)$ is the covariance and $\text{var}(\cdot)$ is the variance. The meaning of Eq. 7.3 is that when the covariance between a part and another increases, the variance within the part itself must decrease by the same amount.

We can obtain proper correlations by either using ILR-coordinates directly in Eq. 7.2 or from the logratio variation matrix **T**, defined in def. 6.1.2 as $t_{ij} = \mathrm{var}(\ln[x_i/x_j])$. If we already have **T**, we can obtain the covariance matrix **S** *in the basis* $\mathbf{\Psi}$ from,

$$\mathbf{S} = -\frac{1}{2}\mathbf{\Psi} \times \mathbf{T} \times \mathbf{\Psi}^T. \tag{7.4}$$

From this, we can calculate the correlation coefficient matrix **R** using Eq. 7.2,

$$r_{ij} = \frac{s_{ij}}{\sqrt{s_i s_j}}. \tag{7.5}$$

The interpretation of **R** is straight forward: r_{ij} close to 0 means little or no correlation, while r_{ij} close to ± 1 means a linear association between the i'th and the j'th balance coordinate.

It should be noted here that even though it isn't possible to straight up calculate correlations between parts in compositional data, there are algorithms that attempt to estimate correlated parts. SparCC (Friedmann & Alm, PLOS Computational Biology, 2012) is a reasonably successful method that works on large compositions with relatively few correlations.

7.1.2 Balance dendrogram

A useful way to visualize a sequential binary partition basis is by using a balance dendrogram. A balance dendrogram shows the basis in a familiar tree structure. In addition to showing the partitioning, it also shows the sample variance and the geometric mean of each balance, represented by branch length and branch split inception point.

As an example, we can construct a sequential binary partition basis for the protein data set from Chapter 6. One such partition could be,

	Red meat	White meat	Eggs	Milk	Fish	Cereals	Starch	Nuts	Vegetables
1	1	1	1	1	1	-1	-1	-1	-1
2	1	1	-1	-1	1	0	0	0	0
3	1	1	0	0	-1	0	0	0	0
4	1	-1	0	0	0	0	0	0	0
5	0	0	1	-1	0	0	0	0	0
6	0	0	0	0	0	1	-1	1	-1
7	0	0	0	0	0	1	0	-1	0
8	0	0	0	0	0	0	1	0	-1

The choice here is to separate vegan from non-vegan in the first row, vegetarian from non-vegetarian in the second row, fish from meat in the third row, and finally meat in the fourth row. In the sixth row, we rather arbitrarily split cereals and nuts from starch and vegetables, both of which are split in rows seven and eight. Using this basis, we can calculate ILR coordinates, from which we can calculate the coordinate means and variances. The result is displayed in a balance dendrogram in Fig. 7.1. Long vertical lines (blue) mean large variance, whereas short lines mean low variance, i.e., that the coordinate is close to constant across the data set.

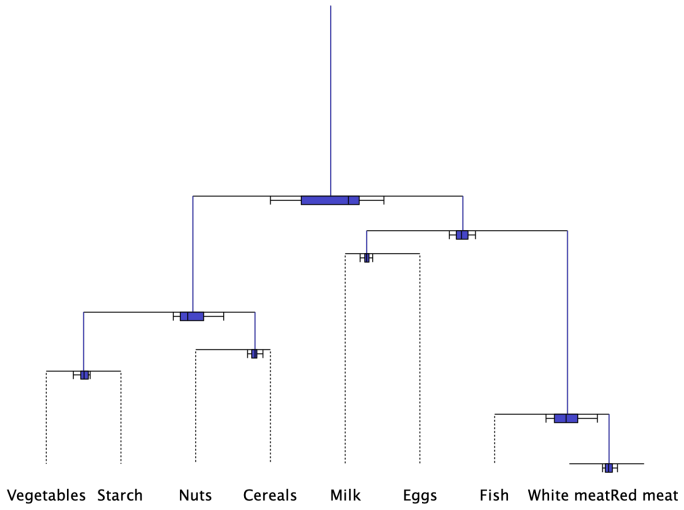

Figure 7.1: A balance dendrogram for a specific sequential binary partition of the protein data set.

7.2 Principal component analysis revisited

Principal Component Analysis, which was introduced in the last chapter, is an example of linear dimensionality reduction. Let us briefly consider the mathematics behind PCA. The PCA is a visualization of the eigenvectors belonging to a matrix that represents a set of compositions. Eigenvectors and the corresponding eigenvalues are defined as

$$A\mathbf{x} = \lambda\mathbf{x}, \tag{7.6}$$

where A is the matrix representation of the data set, $\mathbf{x}$ are the eigenvectors, and λ are the eigenvalues. The interpretation of this equation is that an eigenvector is a vector that, when multiplied by a matrix, does not change direction – only length, and the change in length, the scaling, is determined by the corresponding eigenvalue. The larger the eigenvalue, the more the corresponding eigenvector will be scaled, and we call the two (or sometimes three) eigenvectors with the largest eigenvalues the principal components. A PCA bi-plot is simply the samples and the parts projected onto a plane (or 3-space) spanned by the principal components.

In general, eigenvectors are a linear combination of the parts and therefore can not uniquely be associated with a single part, but we can construct an example where the eigenvectors exactly line up with the basis vectors of the data. Consider a 2D data set of 100 points sampled from the function $f(x) = 5\sin(3x),\ x \in \{-10, 10\}$. The samples are shown in Fig. 7.2. The resulting data set is not compositional (the sum of x- and y coordinates has no meaning), but it nonetheless serves our purpose. By construction, the eigenvectors of this data set coincide with the x- and y-axes, which can easily be seen from the fact that if the frequency (x-axis) or amplitude (y-axis) is changed, points will move only along those two directions. A PCA plot should therefore have the feature vectors along the principal components, which indeed it has, as can be seen in Fig. 7.2. Because the data is two-dimensional, we can describe 100% of the variation using just two principal components. However, if we allowed a non-linear principal component, particularly a

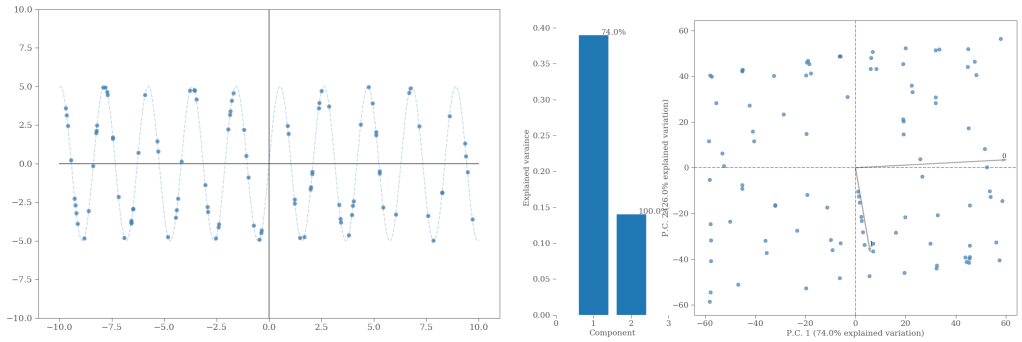

Figure 7.2: Left panel shows 100 random points drawn from a sine function. Right panel shows the PCA of the x and y coordinates of those points.

principal component along the direction of $5\sin(3x)$, the we could describe 100% of the variation using only 1 principal component. In other words, if we use a nonlinear dimensional reduction method, we can describe more variation using the same number of components or the same variation using fewer components. A number of non-linear dimension reduction methods exist, many of which can be seen as generalizations of PCA. In this chapter, we will learn about one of these methods, namely the self-organizing map.

7.3 Self-organizing maps

A self-organizing map (SOM) is a type of artificial neural network (ANN), trained using unsupervised learning, to produce a two-dimensional discrete approximation of the input space from which the training samples are drawn. The purpose is to create views of high-dimensional data that preserve as much of the high-dimensional structure as possible, and it is primarily a visualization technique. The number of neurons in the ANN determines the type of map that is produced. If the number n of neurons is much smaller than the number M of samples (n$\ll$m), the result is akin to k-means clustering (see Sect. 7.4) with k=n, whereas if n$\gg$m, the result is more akin to a topological map. In the special case of a one-dimensional SOM, the output can be interpreted as a non-linear PCA, as illustrated in Fig. 7.3. In this figure, the first (linear) principal component is shown in blue, and it falls along the direction of the largest variance in the data (grey points). The red squares are a one-dimensional neuron grid, which approximates the data much better than the linear component.

The neurons of the ANN are arranged in a two-dimensional grid, either as a rectangular grid or a hexagonal grid. The grid can either be finite or it can have periodic boundaries so that it describes the surface of a torus. Each cell in the grid represents a neuron, which again is described by a vector with the same length as the input vectors (the samples). These neuron vectors (sometimes called weights) are initialized with random values.

The ANN needs to be trained, which is done by feeding it a set of training samples – usually the entire data set that we wish to visualize. In an iterative manner, going through each sample, the ANN identifies the neuron that is most similar to the sample and moves it as well as the adjacent neurons towards the sample. The distance we allow the neurons to move decreases at each iteration, and the training continues until either after a predefined

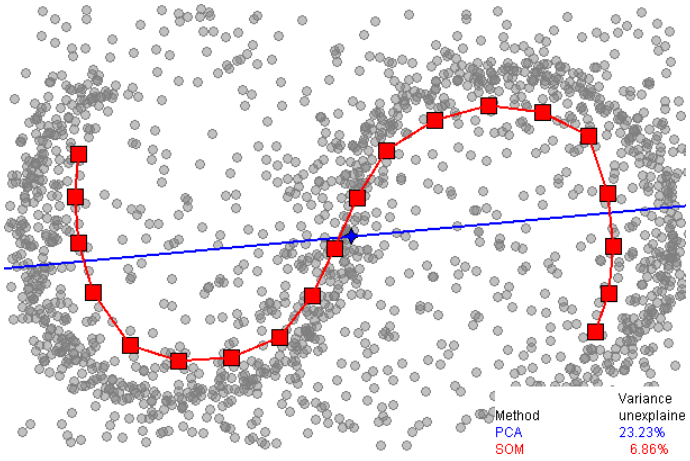

Figure 7.3: Comparison of PCA to SOM: blue shows the first principal component of the data, which is a linear approximation along the direction of largest varaince. Red is the 1D SOM, which can be interpreted as a non-linear principal component. Source: wikipedia

set of iterations or until the map has converged, i.e., the neurons don't move anymore. The result is that the network of neurons gets stretched and bends into the shape of the data, but its topology is preserved, so there is no twisting or rearranging of the neurons. Because the ANN determines the distance between neurons and samples using the Euclidean metric, we need to provide CLR-transformed values as input when mapping compositional data.

Once the training has converged and the network is molded into a shape that approximates the data, we visualize the network by coloring the map according to the distance from the neurons to their neighbors. We can then map the samples onto this map by plotting a sample label next to the neuron that is closest to the sample. If the number of neurons is small, many samples will map to the same neurons, and we will have a clustering algorithm, while a large number of neurons will provide a topological map that shows the samples locations relative to each other.

Figure 7.4 shows the European protein consumption data on a SOM. In this case, the SOM was made using a 100×100 rectangular map with finite boundaries. The dark ridges indicate greater neuronal distances, while the light patches describe neurons that are closer together. The SOM clearly separates the southern European countries from the northern, and it separates the western Mediterranean countries from the southern East Block countries. France, which we by random definition label as southern, is placed among the northern countries, while the opposite is true for Hungary. The four Scandinavian countries are grouped together inside their own light patch while still being close to Germany and the UK. The SOM tends to place countries closer that are physically close to each other (France and Belgium, Denmark and Germany, Portugal and Spain, etc.), almost preserving physical topology. Albania is clearly seen to be an outlier but is still placed among the other south-eastern countries. We can also see from the SOM that the central European countries are quite similar, regardless of whether they belong to the eastern or western block.

We can use such a map to predict the country of origin for a person if we know the protein consumption composition of that person. We simply map the CLR-transformed

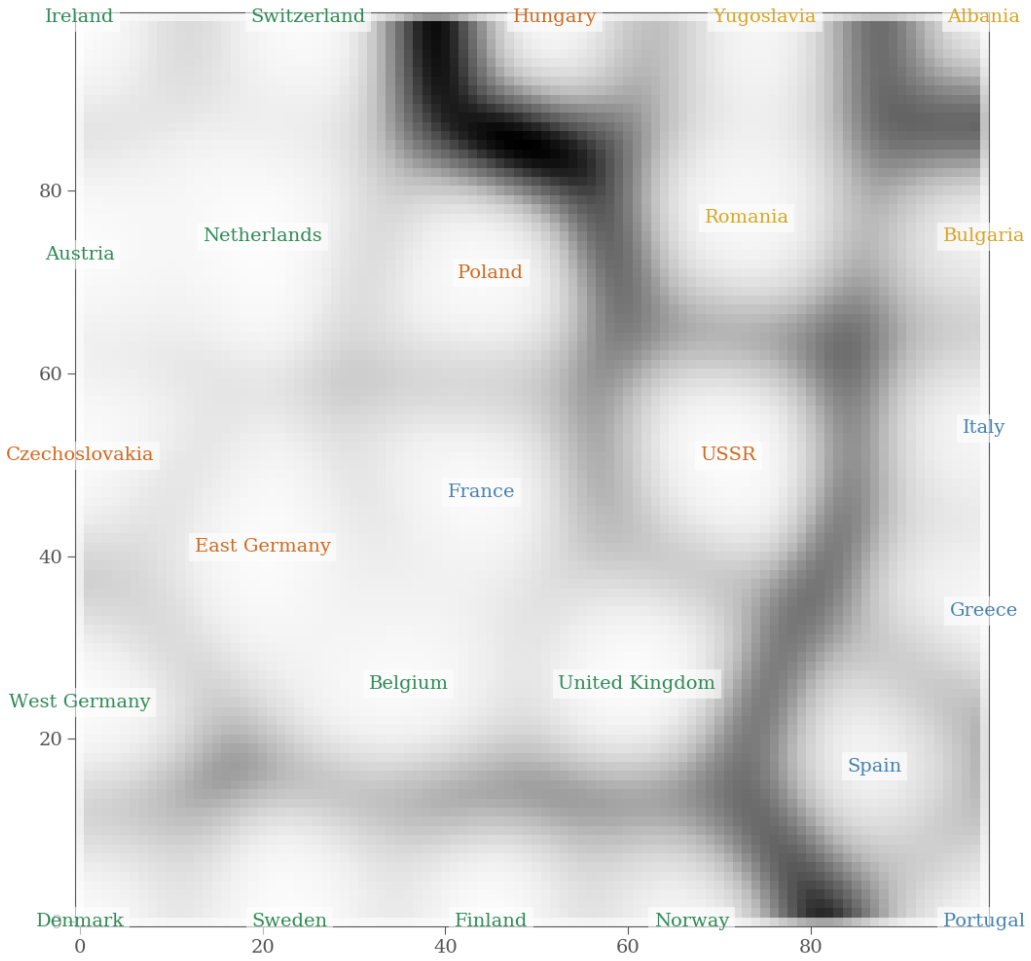

Figure 7.4: Self-organizing map of the protein consumption data. Black ridges correspond to large neuron distances.

composition to the SOM, which gives us the neuron that most resembles the person. By identifying that neuron on the map, we can predict which part of Europe and possibly which country the person lives in.

There are a few downsides to SOMs, though. Due to the fact that they need to be trained and converge, they can be quite computationally expensive to make. But a more problematic aspect is that the algorithm is stochastic by nature, and SOMs are therefore difficult to reproduce. This can, to some extent, be remedied by initializing the weights, not with random numbers, but by sampling from the subspace spanned by the first two principal components. This provides a more robust starting point, which already traces the data, and the resulting map will be more consistent with consecutive runs, but there is still some randomness involved. Moreover, a large number of parameters can be adjusted, for instance the learning rate, which makes the resulting maps a bit ambiguous.

We can try and produce a SOM in the regime where the number of neurons is small compared to the number of samples, so that the SOM resembles a k-means clustering of the data. If we chose 4 neurons, corresponding to 4 clusters, we find that the SOM divides

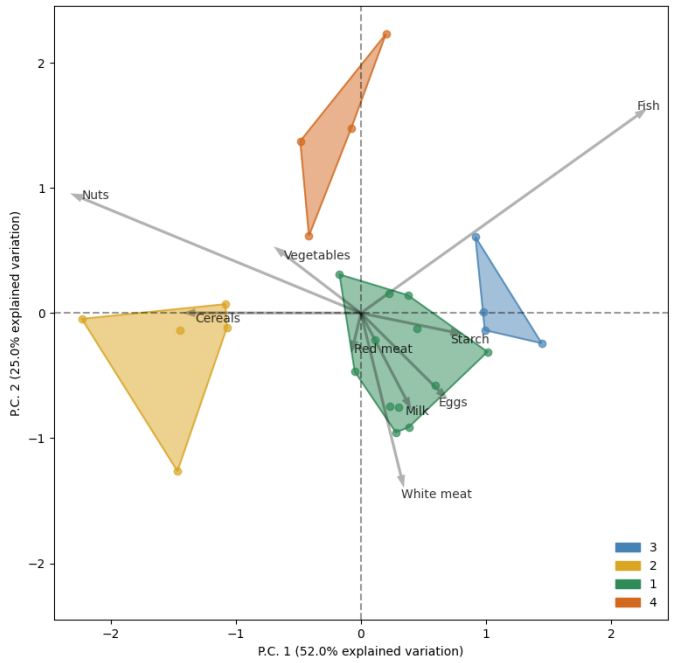

Figure 7.5: A PCA biplot of the protein data with samples colored according to the 4 clusters defined by the SOM. The convex hull has been drawn for each group to show that there is no overlap.

the data set into the following partition:

1	2	3	4
Austria	Albania	Denmark	Greece
Belgium	Bulgaria	Finland	Italy
Czechoslovakia	Hungary	Norway	Portugal
East Germany	Romania	Sweeden	Spain
France	Yugoslavia		
Ireland			
Netherlands			
Poland			
Switzerland			
USSR			
United Kingdom			
West Germany			

Here, cluster 2 consists of the south-eastern countries, cluster 3 is Scandinavia, cluster 4 is the Mediterranean countries, and cluster 1 is all the rest. If we chose five clusters, we would split the USSR off of cluster 1 into its own cluster with Romania. If we recreate the PCA biplot from Chapter 6 and color our samples according to the 4 clusters, we see that the SOM splits the samples nicely into 4 distinct groups, which is shown in Fig. 7.5.

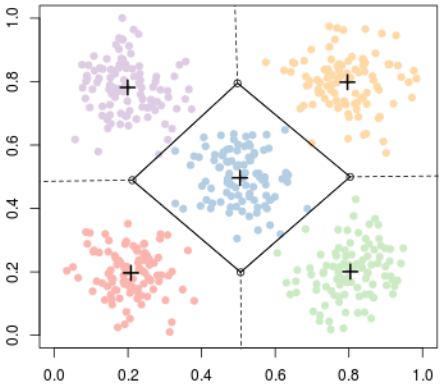

Figure 7.6: The result of k-means clustering is a Voronoi tessellation.

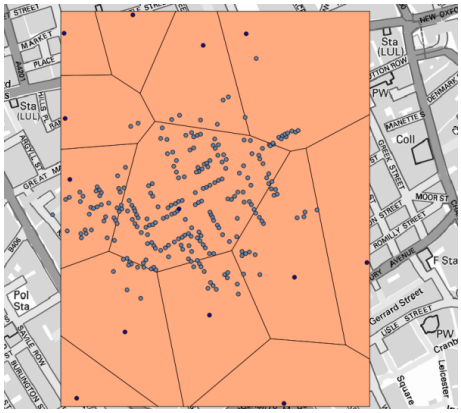

Figure 7.7: John Snow's map of 1854 cholera cases in London, with a Voronoi diagram superposed.

7.4 k-means clustering

One commonly used technique to identify clusters and particularly to define the membership of clusters is k-means clustering. This technique partitions n samples into k clusters so that each sample belongs to the cluster with the nearest mean. The result is a so-called Voronoi tessellation with the cluster means as generators.

One of the earliest and most famous, while also coincidental, uses of k-means clustering and Voronoi diagrams was when the British doctor John Snow discovered that the source of a London cholera outbreak was the water pump in Broad Street. He realized that the vast majority of cases lived closer to that pump than to any other water pump in London, and therefore that the one thing that all cases had in common was that they all collected their water in the same place. A modern rendering of the Voronoi diagram with the location of the water pumps as generators is shown in Fig. 7.7, where it is easily seen that almost all cases lived within the same Voronoi cell.

We can use the same approach when trying to identify samples with similar properties in a data set. Sometimes we know how many clusters we expect (like John Snow knew the number of water pumps), but most often we do not know how many clusters we have. In that case, if we cannot visually determine the appropriate number of clusters, we must repeat the clustering algorithm for an increasing number of clusters and determine when the explained variance no longer increases, at which point we have the most probable number of clusters. How to determine the variance between clusters is the topic of the next chapter.

8. Linear models

"Linear model" is an umbrella term for models that relate two sets of random variables with linear relations. One set of variables is called *response variables* and they are to be predicted by the model from the second set of variables, called *predictor variables* or sometimes *covariates*. The standard approach is to find linear combinations of predictors that result in a response. With very many variables in either group, this becomes a typical task for various machine learning algorithms, which is beyond the scope of this course.

Both the predictors and the covariates can be compositional, in which case compositional analysis must be applied rather than standard methods. A special case is of particular relevance for metagenomic data analysis, namely when the response is a composition and the covariate is a categorical variable. This situation arises, for instance, if we take a number of blood samples from a group of people and, after sequencing, extract the metagenomes. These are the compositional responses. The categorical variable used as a covariate could be the sex of the persons. The linear model would try to predict which genetic elements could be predicted by the covariate. In this case, the answer would be the abundance of the Y chromosome. Inthis situation, ANOVA is a popular choice for a linear model. In this chapter, we will deal with the simple case of a single predictor yielding a response. In reality, samples are described by a large number of covariates, and multivariate methods need to be applied.

8.1 Linear regression with compositional response

A common type of linear model is regression. However, as we have seen many times so far, we cannot do linear regression directly on compositional data because the method of least squares, which is the typical way to fit the regressions, is based implicitly on the Euclidean metric. The assumption of linear regression is that the data deviates randomly from a mean model, and the method of least squares seeks to determine the mean model by minimizing the residuals between the model and the data. These residuals, the sum of squares, are the

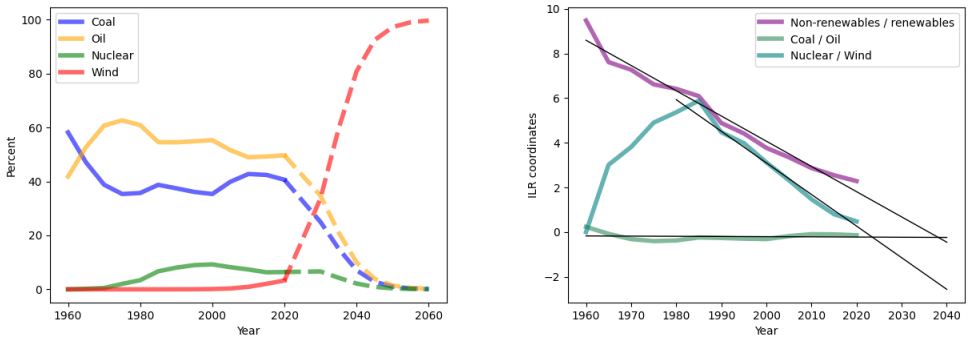

Figure 8.1: The left panel shows a 4-part sub-composition of the world's energy production between 1960 and 2020. The dashed lines extending beyond 2020 are linear extrapolations. The right panel shows the ILR coordinates on a basis where oil and gas (non-renewables) are split from nuclear and wind (renewables). The black lines are linear regression models of the ILR coordinates.

(squared) distances between the model and the data.

As usual, we can choose to implement a least squares algorithm using the Aitchison distance, or we can choose to transform our data into $\mathbb{R}^{D-1}$ and perform the model fitting using the normal least squares method.

For a practical application of compositional linear regression, we consider the world's energy production from four sources, coal, oil, nuclear, and wind, between 1960 and 2020. We close the 4-part sub-composition to 100, and the result is shown in the left panel of Fig. 8.1 as solid colored lines. It is difficult to tell by looking at the proportions whether the parts evolve linearly in time. It is also clear that if we tried to do linear regression directly on the proportions, this would lead to a nonsensical (and non-closed) result.

Instead we build a binary partition and use it to form an orthonormal basis to be used for ILR transforming the data. The partition we chose is the following,

	Coal	Oil	Nuclear	Wind
v_1	1	1	-1	-1
v_2	1	-1	0	0
v_3	0	0	1	-1

(8.1)

where we have chosen to split coal and oil from nuclear and wind (non-renewables from renewables; it is obviously debatable whether nuclear can be considered a renewable), and then split coal and oil and finally nuclear and wind. With this partition, we ILR transform the data, and the resulting coordinates are plotted in the right panel in Fig. 8.1. It can clearly be seen that the first two ILR coordinates are well approximated by linear functions over the entire period. The third coordinate (nuclear over wind), however, is clearly not linear over the entire period. We therefore only use the data from 1980 to 2020 to do the regression on that coordinate.

A regular least squares fit can be done on the ILR coordinates, and the resulting models are shown as thin black lines on top of the data. Notice how the fit to the third coordinate only covers the data from 1980 and onward. The regression models can be

extrapolated into the future, and we can do an inverse ILR transformation of these future values back to the simplex to obtain predicted proportions in the future. This is shown as dashed line segments in the left panel of Fig. 8.1. There are a few caveats, though. This extrapolation predicts that wind will reach 100 percent by 2060, but remember that this is in the sub-composition of only these four energy sources and the fact that nuclear has actually been declining since the 1990s while wind has grown tremendously over the same period, something which is unlikely to continue at the same rate over the next 40 years.

It should also be noted that this extrapolation depends on our choice of basis. If we had split our energy sources differently, our predicted energy budget would be different.

8.2 Analysis of Variance (ANOVA)

Statistical models that are used to analyze the difference between variation within a group and variation between groups are collectively known as ANOVA. ANOVA originated in evolutionary biology, where it was used to test whether variation between two groups of animals were larger than the variation within each group and use the result to identify new species. Testing two groups of responses that are separated by a categorical covariate is one obvious application of ANOVA, making it highly applicable for discovering effects within a set of metagenomic samples.

8.2.1 Hypothesis testing

The core of ANOVA is the concept of hypothesis testing. The idea behind hypothesis testing is that an expectation is made, typically expecting no effect, in which case it is called a null hypothesis, and a statistical test is conducted to see if a sample deviates significantly from the null hypothesis. If it does, the null hypothesis is rejected, and if it is not, the premise of the null hypothesis is true (from a statistical point of view).

In applications of ANOVA that are typically relevant for genomics, two sets of samples are tested against each other, and the null hypothesis is that the two sets of samples are identical. For example, consider a number of patients who are infected with a bacterial pathogen. The patients are divided into two groups, and one group is given an antibiotic agent. Afterwards, samples are taken from all patients, and they are analyzed for the presence of the pathogen. Our null hypothesis is that the drug has no effect, and that the pathogen will be present at equal levels in both groups. However, after a statistical test, we may find that the patient group that received the drug has a significantly reduced level of infection, in which case we reject the null hypothesis and conclude that the drug has an effect.

In order to make the decision to reject or accept the null hypothesis, we need to establish a significance level α. The significance level determines how certain we are of our decision. The statistical test returns a so-called p-value, and if $p < \alpha$ we reject the null hypothesis. The default value for α is 5%, which means that out of 100 tests, only 95 will result in a correct decision, or in other words, we can be 95% certain that our decision is correct.

Two types of errors occur in hypothesis testing: (I) rejecting a true hypothesis (true negative) and (II) accepting a false hypothesis (false positive). The rate at which these errors occur depends on the value of α. If set too high, we get many type II errors, and if set too low, we get many type I errors. Finally, just because a hypothesis is rejected,

doesn't mean that the opposite is the only possible hypothesis or even the best, and that is why we reject hypotheses rather than accept them.

8.2.2 Student's t-test

The most common form of testing differences between two normal sample distributions is the Student's t-test. This test determines if there is a significant shift in the sample means, given the variance of the two sets of samples. If the variances of the two distributions differ, the test is referred to as Welch's t-test or unpaired t-tests.

In its simplest form, the test between two distributions, $\mathbf{x}_1$ and $\mathbf{x}_2$ is given by

$$t = \sqrt{n}\frac{\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2}{\sqrt{\mathsf{var}(\mathbf{x}_1) + \mathsf{var}(\mathbf{x}_2)}}, \tag{8.2}$$

where the bars indicate the distribution's (arithmetic) mean and n is the total number of samples in the two distributions. The t value can be converted into a p value (probability of the null hypothesis being wrong) using the t-distribution,

$$f(t,\nu) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\nu/2)}\left(1+\frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}, \tag{8.3}$$

where ν are the degrees of freedom, $\nu = n_1 + n_2 - 2$. t tests are implemented in most programming languages, and they provide both the t and the p values.

8.2.3 F-test

Sometimes it may be useful to test whether the variances between two parts are significantly different (obviously before scaling with the total variance). The procedure is similar, but in this case we use the F-distribution and the corresponding F-test. The F-test has the form

$$F = \frac{\sum_{i=1}^{K} n_i(\bar{Y}_i - \bar{Y})^2/(K-1)}{\sum_{i=1}^{K}\sum_{j=1}^{n_i}(Y_{ij} - \bar{Y}_i)^2/(N-K)}, \tag{8.4}$$

where the nominator is the between-group variability and the denominator is the within-group variability. $\bar{Y}_i$ is the mean of the i'th group with n_i samples, and K is the number of groups. N is the total number of samples.

When there are only two groups ($K = 2$), $F = t^2$ where t is the t-test statistic.

8.3 ANOVA with compositional response

In the following, we will consider a model where the response is compositional and the covariates are categorical variables. The goal of the ANOVA is, in this case, to test if there is a significant difference in the compositional centers between different categories of the covariates.

A compositional ANOVA model can be written as

$$\hat{\mathbf{x}} = \beta_1 \oplus (I(z=2)) \odot \beta_2 \oplus \ldots \oplus (I(z=K) \odot \beta_K), \qquad \mathbf{x} \ominus \hat{\mathbf{x}} = \varepsilon, \tag{8.5}$$

for K different categories. The indicator function $I(z=k)$ equals 1 when the condition is true and 0 otherwise. ε is the compositional residual of the model.

ANOVA methods are typically implemented to work on real coordinates, so the compositional approach is to select a basis, ilr transform the compositions, and apply ANOVA on the resulting coordinates. Afterwards, coefficients can be inversely transformed to return to the compositional coefficients.

In chapter 6, we worked through an example based on protein consumption in the early 1980's in Europe. We will now revisit that data set and carry out an ANOVA to look for food sources that contribute significantly different amounts of protein consumption between eastern and western Europe. The first step is to ILR transform the data, and therefore to choose a basis. In this case, it is important how that basis is chosen because it may affect the outcome of the ANOVA. Therefore, if we have previous knowledge of associations between food sources and country location, for instance, from the PCA or from sub-compositional analysis, our basis should be informed by this. Prior knowledge, however, is not always available, so we will proceed as if no such information were available.

In this case, we start by picking an arbitrary hierarchical binary partition, which we normalize to form an orthonormal basis. From this basis, we can calculate ILR coordinates.

	RM	WM	E	M	F	C	S	N	V
v_1	-1	+1	0	0	0	0	0	0	0
v_2	-1	-1	+1	0	0	0	0	0	0
v_3	-1	-1	-1	+1	0	0	0	0	0
v_4	-1	-1	-1	-1	+1	0	0	0	0
v_5	-1	-1	-1	-1	-1	+1	0	0	0
v_6	-1	-1	-1	-1	-1	-1	+1	0	0
v_7	-1	-1	-1	-1	-1	-1	-1	+1	0
v_8	-1	-1	-1	-1	-1	-1	-1	-1	+1

It is now up to the analyst to pick an ANOVA model and fit the two coefficients that correspond to the two categorical variables, north-south and east-west. In this example, we chose a double-sided Student's t test with equal variance in the two groups. This is appropriate if we power the data by the inverse of the total variance, like we did in Chapter 6.

	v_1	v_1	v_3	v_4	v_5	v_6	v_7	v_8
β_1	-0.40	-0.13	-0.06	0.31	0.73	-0.25	1.28	0.51
β_2	-0.22	0.26	0.05	1.07	-0.81	-0.09	-0.59	-0.10

The coefficients come out in ILR coordinates, which may be slightly inconvenient, but, just like we can go from CLR to ILR by taking the dot product with the transposed basis (eq. 3.3.4), we can go from ILR to CLR by taking the dot product with the un-transposed basis. Thus, we can easily express our fitted coefficient in CLR values.

	RM	WM	E	M	F	C	S	N	V
clrβ_1	-0.04	-0.61	-0.48	-0.45	-0.05	0.47	-0.46	1.14	0.48
clrβ_2	0.05	-0.26	0.21	0.06	1.21	-0.63	0.01	-0.54	-0.09

These values are easily interpreted from their signs. The coefficient clrβ_2 corresponds to the east-west category, and food items with a positive sign are consumed more in the

west, while coefficients with a negative sign describe food that is consumed more in the east.

The problem with this result is that conclusions based on CLR values are not sub-compositionally coherent because the geometric mean moves, in general, when a sub-composition is considered, and therefore, this result is only valid for the full composition (which is a sub-composition in itself). Another problem is that we can't tell how significant this result is because we can't provide a p-value on individual items against each other. We must use this result as a guide to form another basis on which we can do statistical testing. We do this by grouping together components with similar CLR values. If we sort $\text{clr}\beta_2$ we get

	C	N	WM	V	S	RM	M	E	F
$\text{clr}\beta_2$	-0.63	-0.54	-0.26	-0.09	0.01	0.05	0.06	0.21	1.21

We group the items two by two, so that cereals and nuts, white meat and vegetables, red meat and starch, milk and eggs, and finally fish on its own. Our new tailored basis looks like this

	RM	WM	E	M	F	C	S	N	V
v_1	0	-1	0	0	0	0	0	0	+1
v_2	-1	0	0	0	0	0	+1	0	0
v_3	0	0	-1	+1	0	0	0	0	0
v_4	0	0	0	0	0	-1	0	+1	0
v_5	-1	0	+1	+1	0	0	-1	0	0
v_6	+1	-1	+1	+1	0	0	+1	0	-1
v_7	+1	+1	+1	+1	0	-1	+1	-1	+1
v_8	+1	+1	+1	+1	-1	+1	+1	+1	+1

We proceed with a t test based ANOVA to get the statistics of the coordinates in the tailored basis.

	v_1	v_1	v_3	v_4	v_5	v_6	v_7	v_8
β	0.02	0.01	-0.11	<0.01	0.14	0.35	0.88	-1.29
t value	0.10	0.06	-0.92	0.03	1.05	1.90	2.66	-4.36
$P(>\|t\|)$	0.92	0.96	0.37	0.98	0.31	0.07	0.01	<0.001

If we decide on a confidence interval of 95%, we can remove all coefficients with p-values greater than 0.05. This leaves us with only two balances: fish versus everything else and cereal-nuts versus everything else but fish. The sign of the t-values determines the direction of the overconsumption, which means that fish is consumed more in the west than in the east, while cereals and nuts are consumed more in the east. There is no significant difference in the consumption of meat, dairy, starch, and vegetables between the east and west.

8.3.1 Effect plot

Effect plots are a way to visualize the result of an ANOVA. In essence, an effect plot is the variance between groups plotted against the variance within the groups. Parts (or groups) are plotted as points and can be colored by p-value to show which ones are significant. Figure 8.2 shows the protein data in an effect plot, where we have simply taken the variances on the CLR values.

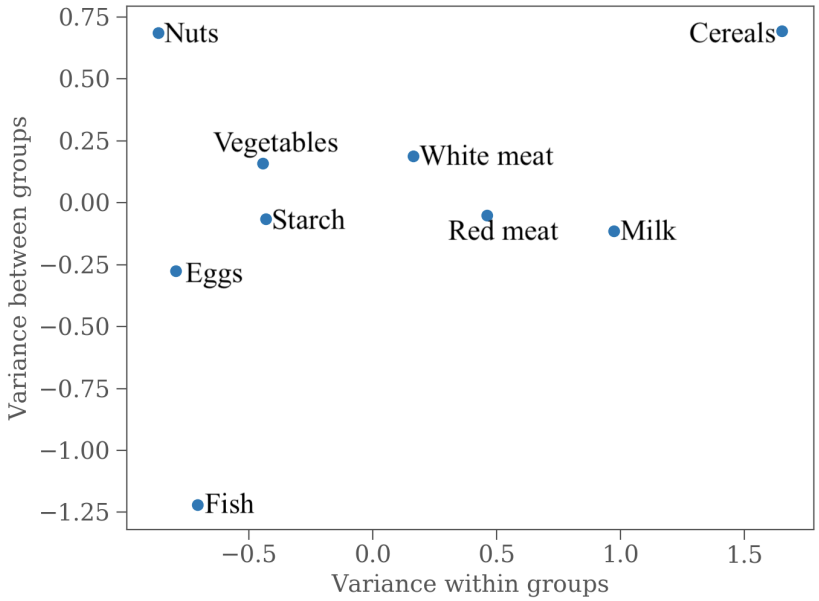

Figure 8.2: Effect plot of the protein data.

8.4 Exercises

Exercise 8.1 Repeat the example in Sect. 8.1 of linear extrapolation of energy production using a different sequential binary partition, specifically the one where nuclear is considered non-renewable:

	Coal	Oil	Nuclear	Wind
v_1	+1	+1	+1	-1
v_2	+1	+1	-1	0
v_3	+1	-1	0	0

The data can be found in the file `world_energy.csv`. ■

Exercise 8.2 Calculate the sample centers for the nutrition data separately for eastern and western countries. Calculate the perturbation difference between the two sample center compositions. Do an inverse CLR transform of the first set of $clr\beta_2$ coefficients in Sect. 8.3. Compare the resulting β_2s to the perturbation difference. The data can be found in the file `protein.csv`. ■

Exercise 8.3 Repeat the ANOVA example for the nutrition data, but check for differences between north/south instead of east/west.

Hint: start from the clrβ table. Sort the parts by clrβ_1 and build a basis on similar parts. ■

Exercise 8.4 Redefine "northern" and "southern" countries based on significant differences only within animal products (meats, eggs, milk, and fish). This exercise has no unique solution. ■

9. Compositional processes

So far in this course, we have seen compositions as samples drawn from an underlying distribution. However, just like real vectors, compositions can be parameterized, either by a continuous or a discrete parameter, typically representing time or space. One such example is the data set with the world's energy production between 1960 and 2020 used in the previous chapter.

In some cases, the process is known and can be used to predict or model the evolution of a system, while in other cases, the process is unknown, but obtainable from existing compositional data.

9.1 Time-dependent compositions

A D-part composition $\mathbf{z}$ that evolves in time t is said to be time-dependent and it can be described as

$$\mathbf{x}(t) = \mathscr{C}[\mathbf{z}(t)], \qquad z_i \in \mathbb{R}_+^D \tag{9.1}$$

The dependency can be arbitrarily complex, but among the simplest cases is proportional growth (or decay), where the change in some property z depends proportionally on z itself,

$$\frac{dz}{dt} = \lambda z \tag{9.2}$$

where the sign of the rate parameter λ determines whether we have growth or decay and the magnitude of λ determines how fast it goes. Well-known examples of such processes are radioactive decay and bacterial growth. The solution to Eq. 9.2 is an exponential function,

$$z(t) = z_0 \cdot \exp(\lambda t), \tag{9.3}$$

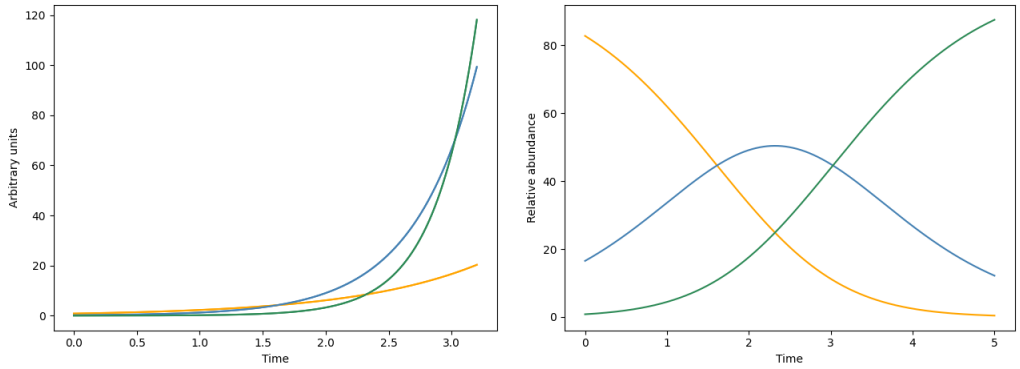

Figure 9.1: The growth in time of three different species of bacteria. The left panel shows the absolute abundances, while the right panel shows the relative abundances.

where z_0 is known as the initial condition or "starting amount". In the case of proportional growth, growth happens exponentially.

If we consider radioactive decay and we have a medium containing multiple radioactive isotopes, we can describe the system by letting z and λ in Eq. 9.2 be vectors with as many entries as there are different isotopes. We can then measure the amount of each isotope z_i, for instance using a mass spectrometer, at $t = 0$ and again at some later time $t > 0$ and solve for each λ_i to obtain their decay rates.

But we can also choose to take a compositional approach to the problem by determining relative abundances rather than absolute abundances. In this case, $\mathbf{z}$ is a composition, which may be closed to 100 for convenience. When $\mathbf{z}$ is a composition, the solution Eq. 9.3 is no longer valid. Recall from Chapter 3 that multiplication (and addition) are not valid operations on the simplex, so we must substitute the operations with their Aichison equivalents (Def. 3.2.1). Doing so gives us the compositional solution,

$$\mathbf{x}(t) = \mathbf{x}_0 \oplus t \odot \mathbf{p}, \qquad \mathbf{p} = \exp(\lambda). \tag{9.4}$$

This solution is recognized as a straight line (in the simplex), with $\mathbf{x}_0$ as the intercept, $\mathbf{p}$ as the slope, and t as the variable. From this, we can go to coordinate space using the ILR transformation, defining $\mathbf{u} = \text{ilr}(\mathbf{x})$ and $\mathbf{v} = \text{ilr}(\mathbf{p})$, so that,

$$\mathbf{u}(t) = \mathbf{u}_0 + t \cdot \mathbf{v}, \tag{9.5}$$

a straight line in coordinate space $\mathbb{R}^{D-1}$.

▪ **Example 9.1 — Bacterial growth.** Three species of bacteria (x_1, x_2, x_3) grow at rates $\lambda = (1, 2, 3)$. The initial relative abundances of the three species are $\mathbf{z}_0 = (82.7\%, 16.5\%, 0.8\%)$. We can easily graph the growth of each species using the solution Eq. 9.3, which is shown in the left panel of Fig. 9.1. The right panel in that figure shows the relative abundance in time, where the solution has been closed to 100 at each point in time.

The growth process can also be shown in a ternary diagram using the compositional solution Eq. 9.4. In the ternary diagram we can track the evolution of the composition along the line. This is shown in Fig. 9.2, where the starting point $t = 0$ and the end point $t = 5$ have been labeled. Equivalently, we can plot this in a coordinate plot, like the ones

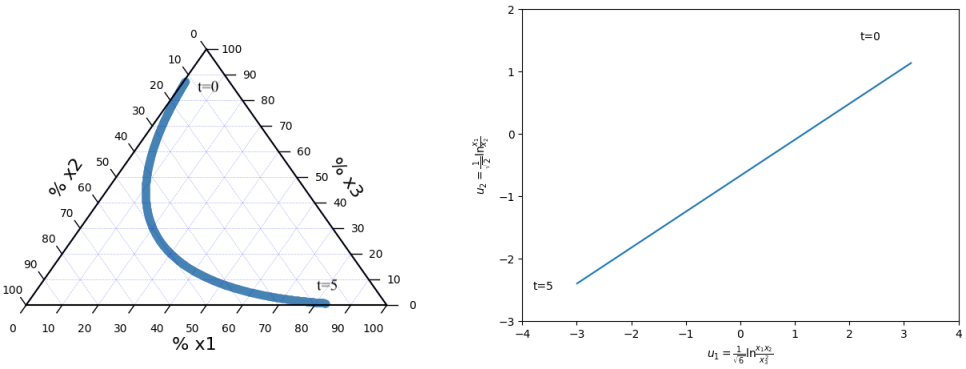

Figure 9.2: The growth in time of three different species of bacteria in a ternary diagram (left) and in coordinate space (right).

we introduced in Sect. 5.3.2. In order to obtain that plot, we need a basis, which we have chosen to be the normalized partition $((+1,+1,-1),(+1,-1,0))$. From the coordinate plot, it is very clear that we are dealing with a linear compositional process.

■

9.2 Compositional derivatives

An important part of the field of calculus, and in particular differential equations, is the derivative, which describes the rate of change of a function with respect to a variable. The derivative is defined as the difference of two slightly offset function values in the limit where the offset approaches zero. This is written as

$$\frac{df(t)}{dt} = \lim_{h\to 0} \frac{1}{h}\left(f(t+h) - f(t)\right) \tag{9.6}$$

This usual form is defined for a real function f and the derivative of f will likewise take real values. If we want to define derivatives for compositions, we need to recall that the sample space for compositions is the simplex, and thus we have to use the homologous algebraic operations, which we introduced in chapter 3. We can see that the expression above contains the inverse sum (minus) of two function values as well as an inverse scale multiplication (divide by h). Defining derivatives for composition thus comes down to replacing those operations by (inverse) perturbation and (inverse) powering,

$$\frac{d_{\oplus}\mathbf{x}(t)}{dt} = \lim_{h\to 0} \frac{1}{h} \odot \left(\mathbf{x}(t+h) \ominus \mathbf{x}(t)\right), \tag{9.7}$$

where $\mathbf{x}$ is a compositional process. We use the subscript $\oplus$ to denote the compositional derivative. However, this definition is not practical for calculating actual derivatives, but we can apply our usual strategy of transforming the compositions into coordinate space, carry out regular real calculus, and back-transform into the simplex,

$$\frac{d_{\oplus}\mathbf{x}(t)}{dt} = ilr^{-1}\left[\frac{d\ ilr(\mathbf{x}(t))}{dt}\right] \tag{9.8}$$

The CLR transformation can be used likewise, since CLR is also isometric. However, we can not use ALR for doing derivatives because it is not isometric. The CLR version is similar to the ILR version, but due to the definition of CLR, the expression can be simplified to,

$$\frac{d_{\oplus}\mathbf{x}(t)}{dt} = clr^{-1}\left[\frac{d\ clr(\mathbf{x}(t))}{dt}\right] = \mathscr{C}\left[\exp\left(\frac{d\ \ln(\mathbf{x}(t))}{dt}\right)\right]. \tag{9.9}$$

The compositional derivative has analogue qualities to what we are used to from real calculus. It is well known that if we differentiate a constant using the normal derivative, we get zero. Likewise, if we take the compositional derivative of a constant composition, i.e., a composition that is not part of a process, we get $\mathbf{n} = (1/D, 1/D, ...)$, the center of the simplex. Recall from Sect. 5.3.1 how the center of the simplex acts as a neutral element like 0 does in real algebra.

The above result is a special case of the well-known rule for differentiating polynomials. The standard rule in normal calculus is,

$$\frac{dP_m(x)}{dx} = \sum_{k=1}^{m} k a_k x^{k-1} \tag{9.10}$$

for an m-degree polynomial with coefficient vector $\mathbf{a} = (a_1, a_2, ..., a_m)$. The compositional version applies when the vector $\mathbf{a}$ is a composition, in which case, after replacing the operators by the compositional homologous, we get,

$$\frac{d_{\oplus}P_m(x)}{dx} = \bigoplus_{k=1}^{m} k x^{k-1} \odot \mathbf{a}. \tag{9.11}$$

9.3 Compositional differential equations

Above, in Sect. 9.1, we encountered the simplest possible differential equation. The general form of these first-order differential equations is

$$\frac{d_{\oplus}\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}(t), t), \tag{9.12}$$

that is, the compositional derivative of a compositional process equals a simplex-valued function of the process itself and, in general, time. When $\mathbf{f}$ is not explicitly dependent on time t, the equation describes an isolated system, free of external forcing. Examples of external forcing would be a damped oscillator or, in the case of bacterial growth, the influence of anti-microbials.

In general, we can solve Eq. 9.12 by ILR transforming both sides of the equation. Using Eq. 9.8 and denoting the ILR-corrdinates by $\mathbf{x}^*$, we can write Eq. 9.12 as,

$$\frac{d\mathbf{x}^*(t)}{dt} = \mathbf{f}^*(\mathbf{x}^*(t), t). \tag{9.13}$$

This equation describes a system of $D-1$ ordinary differential equations, which can be solved using standard techniques. Equation 9.13 will in general have a different expression depending on the ILR basis used for the transformation, but every solution to Eq. 9.13 will inversely ILR transform back to the same process $\mathbf{x}(t)$.

When the right-hand side of the differential equation is a constant composition, $\mathbf{z}$, with $z_i = exp(\lambda_i)$, we saw above that the solution is a straight line in the simplex, in which case the differential equation can be written as

$$\frac{d_{\oplus}\mathbf{x}(t)}{dt} = \mathscr{C}exp(\lambda) \tag{9.14}$$

$$\mathbf{x}(t) = \mathbf{a} \oplus t \odot \mathscr{C}exp(\lambda) = \mathbf{a} \oplus \mathscr{C}exp[\lambda_1 t, \lambda_2 t, ..., \lambda_n t]. \tag{9.15}$$

The following example is an application of this equation.

9.3.1 Population dynamics

Early in the 19th century, a discussion was raised between two mathematicians, the Englishman Thomas Malthus and the Belgian Pierre Verhulst, about the future of the human population. Given the exponential increase in population at the time, Malthus proposed that the human population would continue to grow exponentially. Obviously, the Earth is not capable of sustaining an exponentially growing population forever, so Verhulst counter proposed that the population growth would follow the logistic curve, that is, flatten out at a certain constant population. The controversy at the time centered around which differential equation would rightly describe the growth of the human population. Malthus argued that population growth was governed by the equation,

$$\frac{dN}{dt} = N, \qquad N = N_0 exp(t), \tag{9.16}$$

whereas Verhulst proposed the logistic equation and its solution,

$$\frac{dN}{dt} = \alpha N - \beta N^2, \qquad N = \frac{K_1}{1 + K_2 exp(-\alpha t)}, \tag{9.17}$$

as the best way to describe population growth.

As it turns out, when considering the problem from a compositional perspective, both scenarios are solutions to the same compositional differential equation, and the only difference between the solutions is the assumption on the availability of resources. The total amount of available resources M at a given time t can be described as the sum of consumed C and remaining R resources,

$$M(t) = C(t) + R(t), \tag{9.18}$$

and the idea is that consumed resourced is a proxy for the number of people alive. Because consumed and remaining resources sum up to a constant, they can be regarded as parts of a two-part composition. The resource compositional process can thus be written as,

$$\mathbf{x}(t) = M(t)[C(t), R(t)]. \tag{9.19}$$

If we consider the simplicial differential equation and its solution, Eq. 9.14,

$$\frac{d_{\oplus}\mathbf{x}(t)}{dt} = \mathscr{C}[exp(\lambda_1), exp(\lambda_2)] \tag{9.20}$$

$$\mathbf{x}(t) = \mathbf{a} \oplus \mathscr{C}[exp(\lambda_1 t), exp(\lambda_2 t)] \tag{9.21}$$

$$= \left[\frac{a_1 exp(\lambda_1 t)}{a_1 exp(\lambda_1 t) + a_2 exp(\lambda_2 t)}, \frac{a_2 exp(\lambda_2 t)}{a_1 exp(\lambda_1 t) + a_2 exp(\lambda_2 t)}\right] \tag{9.22}$$

we can identify both the Malthus and the Verhulst solutions by a proper choice of λ_2 and $M(t)$. If we let $\lambda_2 = 0$ and substitute the solution into eq. 9.19, we get,

$$\mathbf{x}(t) = M(t)\left[\frac{a_1 exp(\lambda_1 t)}{a_1 exp(\lambda_1 t) + a_2}, \frac{a_2}{a_1 exp(\lambda_1 t) + a_2}\right] \tag{9.23}$$

By letting $M(t) = (a_1 exp(\lambda_1 t) + a_2)/a_2$, the solution reduces to,

$$\mathbf{x}_{Malthus}(t) = [(a_1/a_2)exp(\lambda_1 t), 1], \tag{9.24}$$

which is exactly exponential growth in the consumed resources (the population), while the remaining resources stay constant. On the other hand, we can also just let $M(t)$ be a constant (equal to 1 for simplicity), in which case, after dividing the first part by $exp(\lambda_1 t)$, the solution becomes,

$$\mathbf{x}_{Verhulst}(t) = \left[\frac{a_1}{a_1 + a_2 exp(-\lambda_1 t)}, \frac{a_2}{a_1 exp(\lambda_1 t) + a_2}\right] \tag{9.25}$$

The first part of this solution, which describes the population growth, is clearly recognized as having the same algebraic form as the solution in Eq. 9.17, the Verhulst solution.

So both scenarios can be derived from the same underlying compositional process. The difference between them is that Malthus assumed that there would always be a constant amount of remaining resources or an infinite amount of available resources, while Verhulst assumed that the sum of remaining and consumed resources would be constant, corresponding to a finite amount of available resources.

9.3.2 Epidemics

One of the standard tools used by epidemiologists when facing a new epidemic is the compartment model, also known as SIR, where S, I, and R stand for susceptible, infected, and recovered. SIR is the simplest compartment model, and very often, additional compartments are introduced, such as exposed, contagious, susceptible-again, and so on. Here we shall consider the SIR model for an epidemic, where people are contagious while they are infected and recovered people cannot be infected again.

The basic form of the model is a differential equation, where the sum of changes in the compartment equals 0,

$$\frac{dS(t)}{dt} + \frac{dI(t)}{dt} + \frac{dR(t)}{dt} = 0. \tag{9.26}$$

Since the number of people is assumed to be constant (no deaths or births), the solution to this equation must have the form,

$$S(t) + I(t) + R(t) = N, \tag{9.27}$$

where N is the size of the population. Equation 9.27 shows that the three compartments S, I, and R form a composition (because they sum up to a constant), and because they are all functions of time, the model is a compositional process.

The three derivatives in eq. 9.26 depends on two rates: the rate of infection or contagiousness β and the rate of recovery γ, with the basic reproduction number defined as

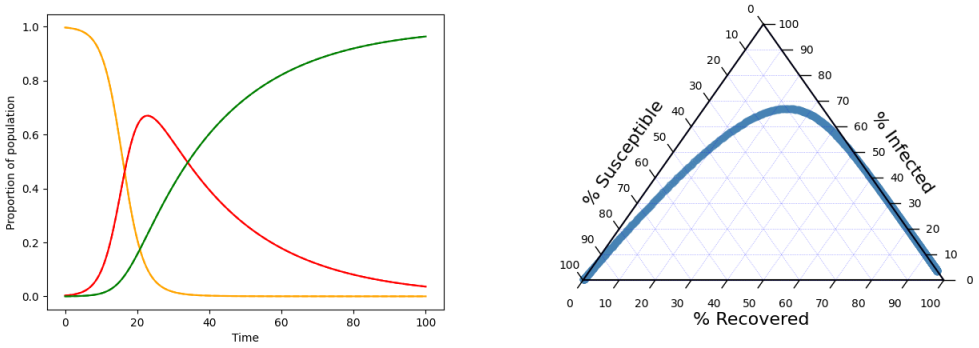

Figure 9.3: The evolution of an epidemic according to the SIR model. Orange shows the susceptible, red are the infected, and green are the recovered population. The right panel shows the process in the simplex.

$R_0 = \beta/\gamma$, the number of new infections caused by one infected individual in a susceptible population. The first rate, β, determines how fast people move from S to I, while the second rate, γ, determines how fast they move from I to R. In the compositional formulation, the three derivatives can be expressed as

$$\frac{dS(t)}{dt} = -\beta S(t)I(t) \qquad \frac{dI(t)}{dt} = \beta S(t)I(t) - \gamma I \qquad \frac{dR(t)}{dt} = \gamma I(t), \tag{9.28}$$

where S, I, and R are the proportion of people in each compartment at any given time t. This is an example of a system of non-linear, first-order, compositional differential equations, that cannot be solved analytically, except for a few trivial cases, such as, when the number of infected is zero and everything stays constant. In this case, the system of equations is also no longer compositional. The non-linearity comes from the mixed term in the first two equations, which involves the product $S \times I$.

Because of the compositional nature of the process, we only need to solve two equations because the third part is determined from the other two through closure, e.g., $R(t) = \kappa - [S(t) + I(t)]$, where κ is the closure constant. Because of this, we can obtain a so-called implicit solution by dividing the equation for dS/dt by the equation for dR/dt,

$$\frac{dS(t)}{dR(t)} = -R_0 S(t), \tag{9.29}$$

which have the implicit solution,

$$S(t) = S(0)e^{-R_0(R(t)-R(0)}. \tag{9.30}$$

A full solution requires numerical integration, and an example solution for a typical value of R_0 of 10, can be seen in Fig. 9.3.

9.4 Exercises

Exercise 9.1 A mineral assemblage contains three radioactive isotopes,

$$[^{238}\mathrm{U},^{232}\mathrm{Th},^{40}\mathrm{K}] = [150, 30, 120] \text{ ppm}$$

at $t = 0$. The half-lives are $4.468 \cdot 10^9, 14.05 \cdot 10^9$, and $1.277 \cdot 10^9$ years, respectively. Plot the evolution of the composition over the course of $50 \cdot 10^9$ years.
HINT: Half-life, $t_{1/2}$, is related to decay rate as $t_{1/2} = \frac{log(2)}{\lambda}$.
HINT: measure time in units of 10^9 years to avoid numeric underflow ■

Exercise 9.2 Consider the compositional differential equation,

$$\frac{d_{\oplus}\mathbf{x}(t)}{dt} = \mathbf{x} \boxdot \mathbf{A} \oplus \mathbf{f}$$

with

$$\mathbf{A} = \begin{bmatrix} 0.56 & 2.55 & -3.11 \\ -1.40 & -1.61 & 3.01 \\ 0.84 & -0.94 & 0.10 \end{bmatrix}, \qquad \mathbf{f}^T = \begin{bmatrix} 0.37 \\ 0.03 \\ 0.60 \end{bmatrix}$$

Using the sequential binary partition,

$$\Psi = \mathrm{norm}\left(\begin{bmatrix} 1 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix}\right)$$

calculate the contrast matrix,

$$\mathbf{A}^* = \Psi \mathbf{A} \Psi^T$$

and its eigenvalues. Obtain the equilibrium solution, $(\partial x = 0)$, for the real-valued equation,

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{x}^* \mathbf{A}^* + \mathbf{f}^*$$

This point is a fixed-point and is transformed back to the simplex, it corresponds to the equilibrium composition of the process. From the eigenvalues, what can you say about the nature of the fixed point? Obtain the full solution (optional).
HINT: Look up "phase portrait behavior" in wikipedia.

■

Exercise 9.3 The SIR model in Sect. 9.3.2 does not take births and deaths into account, and the epidemic therefore dies out when everyone has been infected. We can include births and deaths and thereby replenish the susceptible compartment. For simplicity, we let the birth rate be equal to the death rate so that the population size is constant, i.e., the composition is always closed to the same number. The SIR model with demographics is given by

$$\frac{dS}{dt} = \mu - \beta SI - \mu S \qquad \frac{dI}{dt} = \beta SI - \gamma I - \mu I \qquad \frac{dR}{dt} = \gamma I - \mu R$$

Solve this system using the supplied Python script, and plot the evolution of the model.
■