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 carries 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 in undefined.

In many cases, parts that are zero are simply just 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 an (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, where as 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 which 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 is 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 which kind of zero it is, we need to get rid of them 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 as 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 is the same a detection limit zeros, but not limited to small values. If our measuring equipment saturates at a certain high value, e.g., if you over-expose a photograph, or if your device is limited to a certain max value, for instance, a volt meter which can measure up to 5 volts is used in a 230 volt wall socket, then we get a censored value. The measurement comes out as 5 volts, but we know that the true value is much greater, so the true value is missing.

For these kind 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 sum of the composition 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 increase. 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 increase 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 which is zero due to rounding, we consider the distribution of non-zero values in the remaining samples and make the assumption that the part is 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 among 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 which contains structural zeros in some samples. This is only useful if the number of structural zeros is small and the parts in which they are present a 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 long 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 where 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 the simple situation where a question is simply overlooked and forgotten at random.

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

4.1.4 Amalgamated values

Amalgamated values is a special kind of missing value, where some part in 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 is 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 are 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 a 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 we 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, and 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 of 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, and a wide range of methods exists to deal with latent compositions. The benefit is that we can analyze count data as if they are 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 which most people are familiar with. It has certain limitations however. If we roll a dice 6 times and we 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, however 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 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 one is supposed not to play the lottery. This results in more than 1000 trials for each person with the outcome won/not won. Both persons are asked to record the number of times they 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 then gets 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 prior probability of winning (because that is reasonable in the case where you don't play) you will 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 state 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 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 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 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 know as a flat Dirichlet distribution. α-values above 1 results 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 at an equal abundance. By choosing a sparse distribution, we acknowledge that zero-entries 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 a “correct” number. We can at most hope to obtain a value – given our choice of concentration parameter – which our zero value observation is statistically consistent with. 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 values 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

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

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

$$\mathrm{E}_a[\mathbf{X}] = \mathscr{C}[\exp[\Psi(\alpha'_1)], \exp[\Psi(\alpha'_2)], ..., \exp[\Psi(\alpha'_D)]], \tag{4.6}$$

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 what 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 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:

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

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

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

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 exists 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 from the *k* samples which 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 not 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 on 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, 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 balances. 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 second-most missing values and do the same, and so on until all 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 such iterations, the estimates should converge and now change any more and the we have our model-based replaced zeros after an ILR back 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 square 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 into 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, 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? ■