Algorithms in Computational Biology - Scribe: Dissimilarity measures in metagenomics

Netta Barak

February 2023

1 Introduction

Metagenomics is an approach to studying microbial populations. While traditional microbiology techniques are usually based on isolation and culturing, in metagenomics the entire DNA content of a sample is sequenced and analyzed. While enabling the study of microbial communities as a whole and investigating bacteria that are not easily cultured, metagenomics also introduces many computational challenges. Among these challenges, we can find the visualization of cohorts of dozens or hundreds of metagenomic samples, the mathematical representation of metagenomic data, and the application of clustering and classification techniques. Dissimilarity measures play a crucial role in the common approaches for tackling these challenges.

1.1 Metrics and dissimilarity measures

A metric (also known as distance function) is a function $d : X \times X \to \mathbb{R}$ that defines the distance between two elements, and is required to satisfy the following conditions for every $x, y, z \in X$:

- Non-negativity: $d(x, y) \geq 0$ and $d(x, y) = 0$ if and only if $x = y$
- Symmetry: $d(x, y) = d(y, x)$
- Triangle inequality: $d(x, z) \leq d(x, y) + d(y, z)$

Dissimilarity measures, on the other hand, are functions used for describing the relationships between two elements, returning values in the ranges $[0, 1]$ or $[0, \infty]$ - 0 if the two objects are very similar and 1 (or ∞) when they are very dissimilar. However, these functions are not necessarily metrics as they do not satisfy the aforementioned conditions (commonly the Triangle inequality). Dissimilarity measures are often confused with metrics, although the latter's definition is stricter. In this text, we will refer to functions measuring dissimilarity as dissimilarity measures, although some of them might also be metrics.

2 Dissimilarity measures for composition vectors

When analyzing metagenomic data, it is common to process the fastq files of raw sequencing data into composition vectors. In this approach, millions of sequencing reads are mapped to different databases creating a vector of hundreds of entries (commonly normalized based on the total number of reads and

the total length of the genomes found in the sample so the vector will sum to 1) representing the abundance of different species, genes, or metabolic pathways in the sample [9][6][1]. For simplicity, we will refer to them as vectors of species). Formally, we can annotate two such composition vectors as $A = [a_1, a_2, ..., a_n]$ and $B = [b_1, b_2, ..., b_n]$, where a_i and b_i stand for the relative abundance of the i-th species in samples A and B respectively. It is then common to calculate the dissimilarity between samples by calculating the dissimilarity based on the composition vectors. Many dissimilarity measures can be applied to composition vectors representing metagenomic samples, here we will describe two common ones - the Bray-Curtis and Aitchison Dissimilarity.

2.1 Bray-Curtis Dissimilarity

The Bray-Curtis dissimilarity[3] is a very popular dissimilarity measure for metagenomic composition vectors. For two samples A and B as defined above, the Bray-Curtis dissimilarity measure is defined as follows:

$$Bray - Curtis(A, B) = 1 - \frac{2\sum_{i=1}^{n} min(a_i, b_i)}{\sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i}$$

Note that the dissimilarity will be 0 if and only if the composition vectors are identical, and 1 if and only if the two composition vectors have no shared species, Namely, that $a_i > 0 \Rightarrow b_i = 0$ and $b_i > 0 \Rightarrow a_i = 0$. Intuitively, Bray-Curtis can be thought of as a weighted version of the Jaccard index, assigning different weights to species according to their abundance in the given samples.

2.2 Aitchinson and Robust Aitchison dissimilarity

In recent years, there are voices calling to replace the Bray-Curtis with a dissimilarity measure that is better suited to compositional data [5]. Aitchison similarity, which is simply the Euclidean distance between the centered log-ratio (CLR) transformed composition vectors, is considered to be less affected by highly abundant species, less confounded by the sequencing depth, and robust across data sub-setting and aggregation [7]. Formally, the dissimilarity between two samples A and B is defined as follows:

$$Aithinson(A, B) = \sqrt{\sum_{i=1}^{n} (clr(a)_i - clr(b)_i)^2}$$

Where the CLR transformation is defined as follows

$$clr(A) = [log(\frac{a_1}{G(A)}), log(\frac{a_2}{G(A)}), ..., log(\frac{a_n}{G(A)})]$$

And G is the geometric mean:

$$G(A) = \sqrt[n]{\prod_{i=1}^{n} a_i}$$

The CLR transformation places the metagenomic data in a log-ratio coordinates space while making it symmetric and linearly related [5].

However, a careful look reveals that the standard CLR transformation will almost always result in undefined representations for metagenomic data due to zero-count entries in the composition vectors. If the

counts' vector contains a 0, $G(x) = \sqrt[n]{0}$ and is not defined. Therefore, in order to adapt Aitchinson distance to metagenomic data, the zero counts must be replaced with a non-zero value. In Robust Aitchison (RA), CLR is replaced with Robust CLR (RCLR), simply applying the CLR transformation while taking into account only the non-zero value counts when normalizing by the geometric mean [7]:

$$rclr(A) = [log(\frac{a_1}{G_r(A)}), log(\frac{a_2}{G_r(A)}), ..., log(\frac{a_n}{G_r(A)})]$$

$$G_r(A) = \sqrt[n]{\prod_{i \in [1,n], a_i \neq 0} a_i}$$

3 Dissimilarity measures for raw sequencing data

Another approach for measuring the dissimilarity between metagenomic samples is based on using the raw sequencing files (fastq files) without first representing them as composition vectors, often referred to as *De Novo* comparative metagenomics. This is commonly done by representing the samples as vectors of k-mer counts and applying dissimilarity measures to these count vectors, similarly to the ones applied to composition vectors [2][4][8].

The major advantage of this approach is that it can grasp the sample as a whole, while not taking into account only the species or genes or metabolic pathways found in the sample, but rather the entire genomic content. In addition, it is reference-free, which is very beneficial when studying samples from less explored environments such as ocean microbiome, soil microbiome, and gut microbiome of communities living non-western lifestyles.

However, using a representation based on k-mer counts raises computational challenges due to the high dimension of count vectors - For example for a rather small k like 10, each sample would be represented by a vector of counts 1,048,576 k-mers. It is also important to note that dissimilarity is often calculated between all samples in the dataset, so assuming a relatively small cohort of 100 samples, the k-mers count matrix C would be of size $1,048,576 \times 100$. In order to reduce the dimension of count vectors and allow strand-neutral comparisons, it is common to use only the canonical k-mers by maintaining counts only for the lexicographically smaller of the forward and reverse complement representation of the k-mer [8]. Although this reduces the number of k-mer counts by half, the k-mer counts matrix is still very large. This can result in very long run time and limit the possible k-mer size due to memory limitations. Here bellow we will describe some approaches for coping with these challenges.

3.1 Using parallel computing to calculate dissimilarity measures on k-mer count vectors

Many dissimilarity measures share a very useful feature - the can be calculated by aggregating calculations preformed one coordinate (k-mer, in our case) at the time(ref). Namely, assuming we have N samples that we wouldn't like to keep the entire k-mer count matrix $C \in \mathbb{R}^{4^k \times N}$, we can obtain counts ($C_i \in \mathbb{R}^N$) and calculate intermediate results for each k-mer at the time and discard the k-mer counts. The intermediate results can later be easily aggregated to obtain the dissimilarity measure. This observation holds many dissimilarity measures such as Jaccard, Jensen-Shannon and many more (see).

Here we will demonstrate it on the Bray-Curtis dissimilarity measure described above. We will define the k-mer counts of the samples A and B as $A = [C_{1,A}, C_{2,A}, ..., C_{4^k,A}]$ and $B = [C_{1,B}, C_{2,B}, ..., C_{4^k,B}]$.

Now, let us define a function $f : \mathbb{R}^N \to \mathbb{R}^{N \times N}$. For $C_t \in \mathbb{R}^N$ the counts of the t-th k-mer in all N samples we will get:

$$f(C_t)_{A,B} = min(C_{t,A}, C_{t,B})$$

Let us define another function $g : \mathbb{R}^N \to \mathbb{R}^{N \times N}$. For the same input vector we will define g to be:

$$g(C_t)_{A,B} = C_{t,A} + C_{t,B}$$

We will now show that these functions, calculated for one k-mer counts vector at the time can be aggregated to obtain the Bray-Curtis dissimilarity measure described above. We will do so by defining matrices F and G such that:

$$F_{A,B} = \sum_{i=1}^{4^k} f(C_i)_{A,B}$$

and

$$G_{A,B} = \sum_{i=1}^{4^k} g(C_i)_{A,B}.$$

Note that although these matrices incorporate information from multiple k-mer count vectors, they can be calculated by summing multiple matrices that each one of them is calculated based on the counts of a single k-mer among all samples:

$$Bray - Curtis(A,B) = 1 - \frac{2\sum_{i=1}^{4^k} min(C_{i,A}, C_{i,B})}{\sum_{i=1}^{4^k} C_{i,A} + \sum_{i=1}^{4^k} C_{i,B}} = 1 - 2\frac{\sum_{i=1}^{4^k} f(C_i)_{A,B}}{\sum_{i=1}^{4^k} g(C_i)_{A,B}} = 1 - 2\frac{F_{A,B}}{G_{A,B}}$$

Simka [2] utilizes this feature to distribute the calculation between multiple cores, without holding the full k-mers count matrix in memory, to calculate the Bray-Curtis and other dissimilarity measures.

3.2 Utilizing Min-Hashing to reduce the dimension of k-mer count vectors

Another approach for coping with the high dimension of the k-mer count is based on Min-Hashing, a method for efficiently calculating an unbiased estimation of the Jaccard index of two large sets of objects X and Y defined as:

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}.$$

The idea behind it is that if we could sample objects from both sets in a random-like but yet consistent manner (namely that if an object appears in both sets and was sampled from one of them, it should also be sampled from the other set), we could use these samples to calculate and estimation of the Jaccard index without calculating the intersection and union of the two full sets. This aim can be achieved by applying a hash function on all the objects in the two sets, and keeping for each set only s objects with the smallest hash value. Here bellow, we will describe the adaptation of this method for calculating the dissimilarity between fastq files of metagenomic sequencing data.

For each sample, Mash [8] iterates over the sequencing reads in the fastq file using a sliding window of size k and applies a hash function $h : \{A,C,T,G\}^k \to \{0,1\}^{32}$ (or $h : \{A,C,T,G\}^k \to \{0,1\}^{64}$, depends on the requested size of hash values) on each k-mer. While iterating over the fastq file, it maintains for each sample a sorted list of s minimal hash values by comparing the hash value of the current k-mer to the objects in the list (using binary search) and updating the list if the current k-mer's hash value is smaller than a hash

value in the list. For a given sample A, we will mark the sorted list of s minimal hash values as $S(A)$. The following formula defines the estimator of the Jaccard index of two samples A and B:

$$J'(A,B) = \frac{|(S(A \cup B) \cap S(A) \cap S(B)|}{|S(A \cup B)|} \approx \frac{|A \cap B|}{|A \cup B|} = J(A,B)$$

Note that $S(A \cup B)$ that can be calculated as $\{$the s smallest hash values of $S(A) \cup S(B)\}$. As $S(A \cup B)$ is a pseudo-random sample of $A \cup B$, the fraction of elements of $(S(A \cup B)$ that appear in both $S(A)$ and $S(B)$ are an estimator of $J(A,B)$. This estimator is expected to be more accurate as we increase s. Note that the Jaccard index is not a dissimilarity measure as it equals 1 if the sets are identical and 0 if the sets have no elements in common. To convert it to a dissimilarity measure, $1 - J(A,B)$ or $1 - J'(A,B)$ in the case of approximation using min-hashing is used.

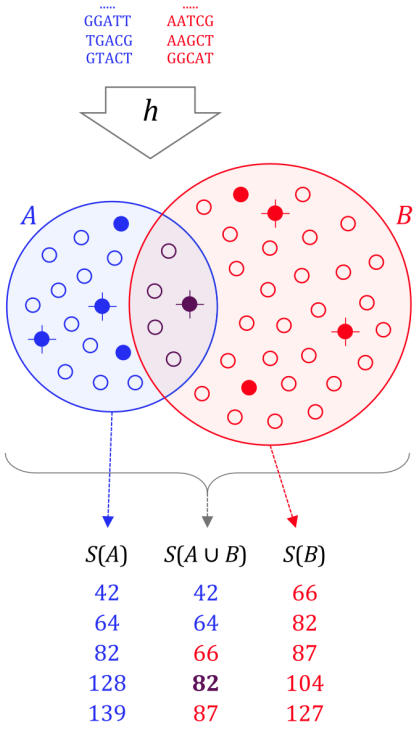

Figure 1: A schematic description of the Jaccard similarity estimation as performed by Mash. First, a hash function is applied on all k-mers from each one of the fastq files. The resulting hash sets, marked as large in blue and red circles, contain $|A|$ and $|B|$ distinct hashes each (marked using small circles). The Jaccard index is the fraction of shared hashes (purple) out of all distinct hashes in A and B. This is approximated by considering a smaller sample sampled using the $s = 5$ minimal hash values from A and B (filled circles). [8]

4 Summary and conclusion

- Dissimilarity measures are used for visualization, clustering, and classification of metagenomic samples.
- Such dissimilarity measures can be calculated based on count vectors, normalized count vectors, or k-mer counts obtained from the raw sequencing data. There is evidence for the correspondence between the distances calculated based on species count vectors and raw sequencing data. While calculating dissimilarity based on raw sequencing data can take into account more aspects of the microbial community and support reference-free analysis, it also introduces significant computational challenges. These challenges can be mitigated using parallel computation and min-hashing.
- Different dissimilarity measures emphasize different aspects of the data - for example, the simple $1-$ Jaccard index weights all species equally while the Bray-Curtis dissimilarity is dominated by highly abundant species. When choosing a dissimilarity measure, one needs to consider the characteristics of the different measures and choose the one most appropriate for the task.

References

[1] Francesco Beghini et al. "Integrating taxonomic, functional, and strain-level profiling of diverse microbial communities with bioBakery 3". en. In: *Elife* 10 (May 2021).

[2] Gaëtan Benoit et al. "Multiple comparative metagenomics using multiset k-mer counting". en. In: *PeerJ Comput. Sci.* 2 (Nov. 2016), e94.

[3] J Roger Bray and J T Curtis. "An ordination of the upland forest communities of southern Wisconsin". In: *Ecol. Monogr.* 27.4 (Feb. 1957), pp. 325–349.

[4] Veronika B Dubinkina et al. "Assessment of k-mer spectrum applicability for metagenomic dissimilarity analysis". en. In: *BMC Bioinformatics* 17 (Jan. 2016), p. 38.

[5] Gregory B Gloor et al. "Microbiome Datasets Are Compositional: And This Is Not Optional". en. In: *Front. Microbiol.* 8 (Nov. 2017), p. 2224.

[6] James Kaminski et al. "High-Specificity Targeted Functional Profiling in Microbial Communities with ShortBRED". en. In: *PLoS Comput. Biol.* 11.12 (Dec. 2015), e1004557.

[7] Cameron Martino et al. "A Novel Sparse Compositional Technique Reveals Microbial Perturbations". en. In: *mSystems* 4.1 (Feb. 2019).

[8] Brian D Ondov et al. "Mash: fast genome and metagenome distance estimation using MinHash". en. In: *Genome Biol.* 17.1 (June 2016), p. 132.

[9] Duy Tin Truong et al. "MetaPhlAn2 for enhanced metagenomic taxonomic profiling". en. In: *Nat. Methods* 12.10 (Oct. 2015), pp. 902–903.