

DADA2

Anders K. Krabberød

a.k.krabberod@ibv.uio.no



UiO : Universitetet i Oslo

Amplicons, denoising, ASVs, OTUs, cluster... 🤔

- **Amplicons:** products of PCR (typically from a metabarcoding region)
- **Denoising:** cleaning of data. Reduce noise from PCR and sequencing
- **ASV:** Amplicons denoised by DADA2 are often called **ASVs** (amplicon sequence variants)
- **Clustering:** Groups of similar things (in this case amplicons or **ASV**)
- **OTU:** Operational Taxonomic Unit. Proxy for species, ecotypes, populations, or another functional unit
- **Species:** who knows?

Illumina seq. and amplicon data

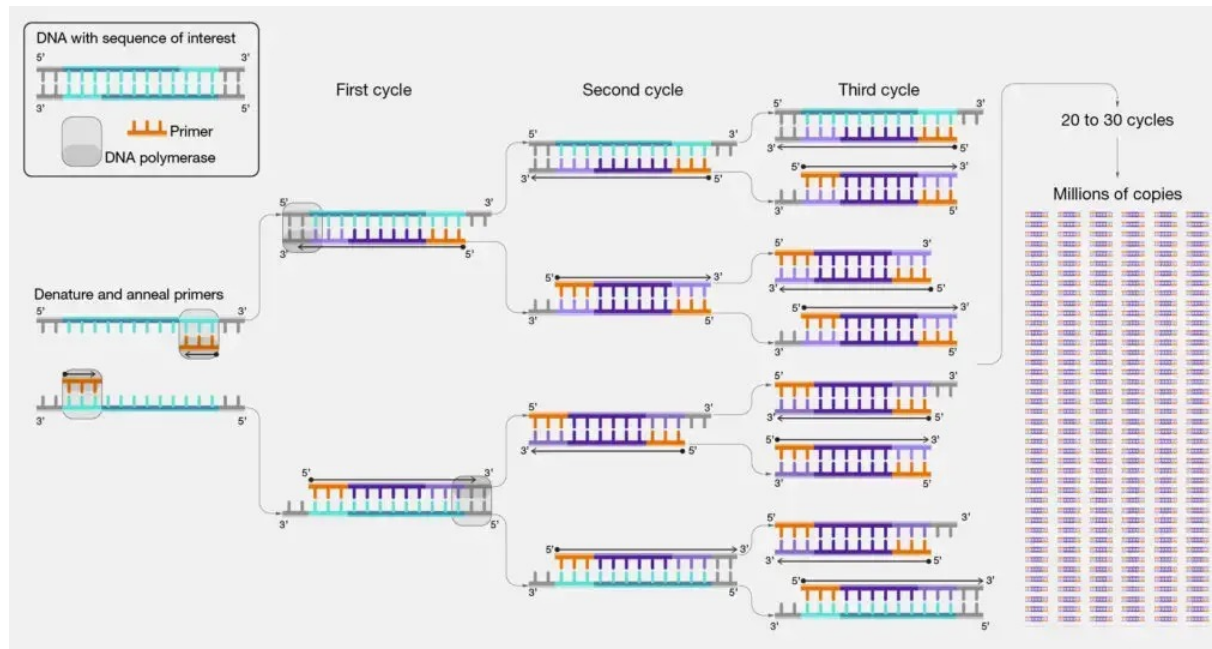
- Illumina HiSeq and MiSeq sequencing have been used extensively for amplicon sequencing in the last years
 - High yields with relatively low cost
 - NovaSeq (and HiSeq) short fragments 2*150bp
 - MiSeq a bit longer reads 2*300bp
- Low error rate, but due to the high number of sequences generated, errors are bound to occur in the library.
 - (About 1% pr 1000 nt, not adjusted for improvements due to overlapping reads.)
- Dealing with errors has been an ongoing discussion for high-throughput data for quite some time.
- But the field is changing rapidly, and new technologies require new methods of dealing with errors.

A small aside... Long-reads are coming fast

- Long read sequencing technology is rapidly evolving:
 - Currently not as high output as short reads (i.e. Illumina), but the taxonomic and phylogenetic resolution gained might outweigh the loss in depth.
 - **PacBio**: Relatively long reads with low error rate (15-20 kbp, 99.9% accuracy).
Oxford Nanopore: Longer than PacBio, but higher error rate and fewer reads as output.
- How to deal with errors for long-reads might be different than shorter reads -> different sequencing platforms have different error profiles.

How to deal with PCR and sequencing errors?

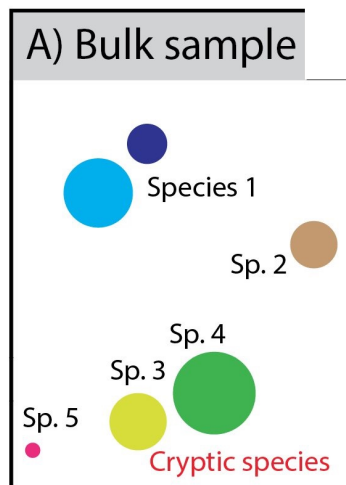
- Errors are introduced both in PCR and in the sequencing
- An increase in the number of cycles in PCR will increase the number of errors introduced by the polymerase



Cycle	Template Copies
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1,024
15	32,768
20	1,048,576
25	33,554,432
30	1,073,741,824

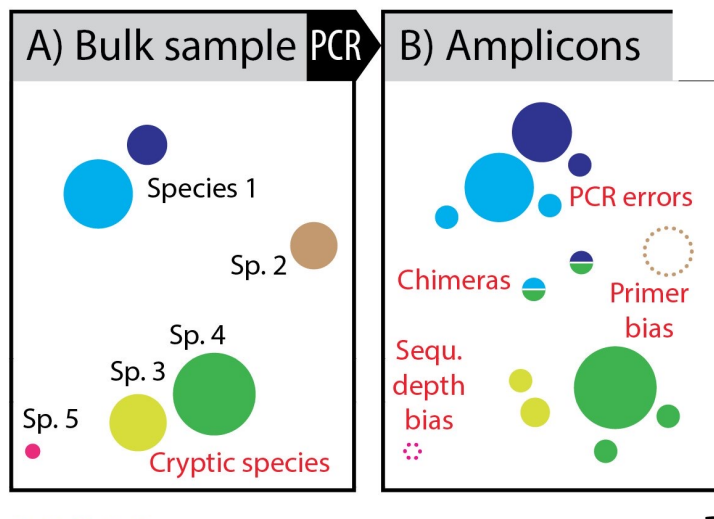
How to deal with PCR and sequencing errors?

- Errors are introduced both in PCR and in the sequencing
- An increase in the number of cycles in PCR will increase the number of errors introduced by the polymerase



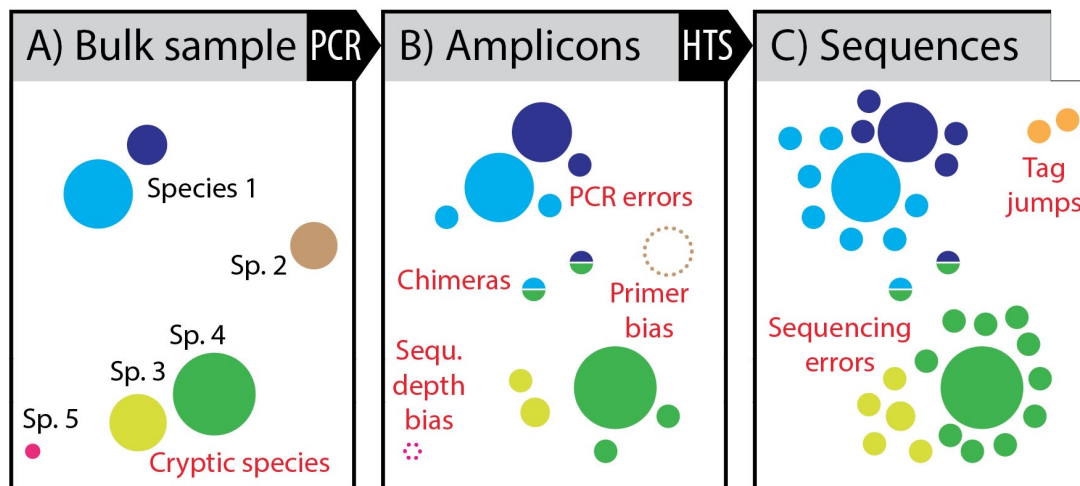
How to deal with PCR and sequencing errors?

- Errors are introduced both in PCR and in the sequencing
- An increase in the number of cycles in PCR will increase the number of errors introduced by the polymerase



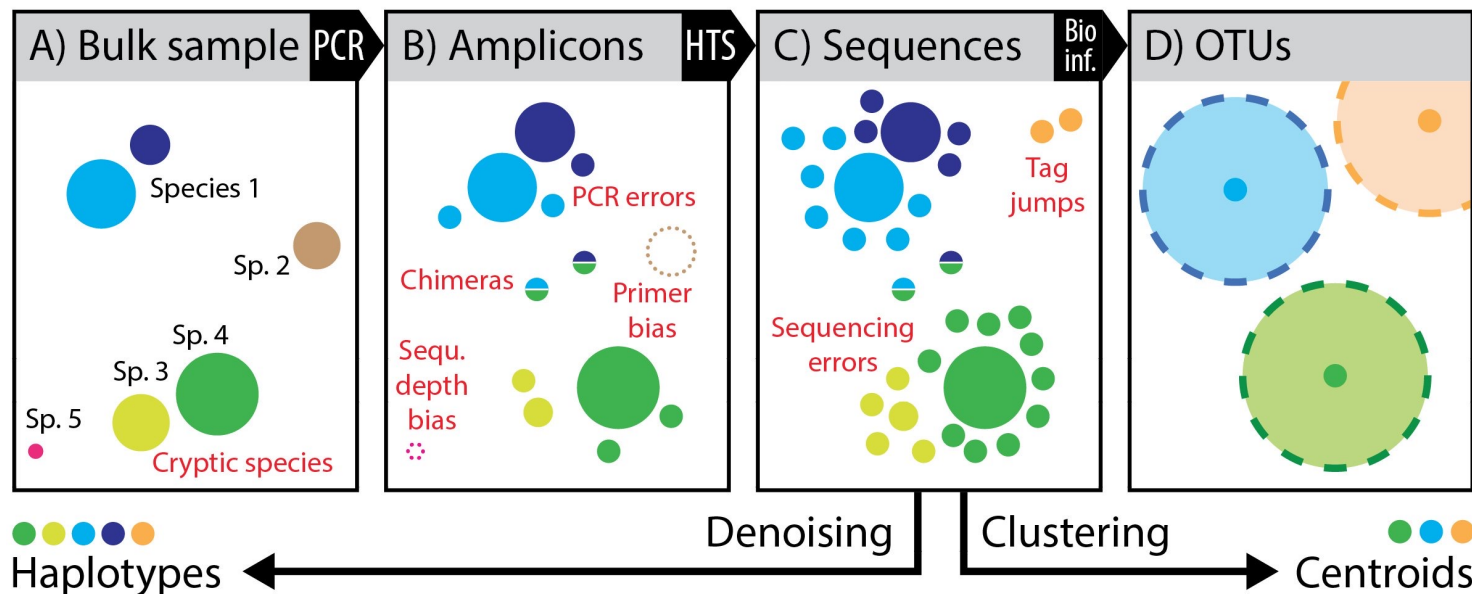
How to deal with PCR and sequencing errors?

- Errors are introduced both in PCR and in the sequencing
- An increase in the number of cycles in PCR will increase the number of errors introduced by the polymerase



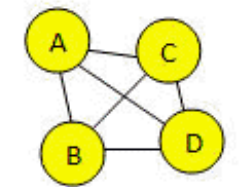
How to deal with PCR and sequencing errors?

- Reasons for clustering of metabarcoding data:
 - Reduce effect of sequencing error
 - Reduce “noise”

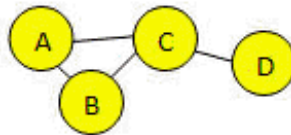


Clustering types

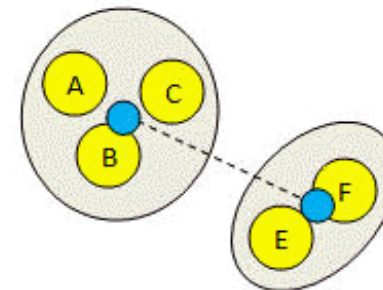
- **Complete linkage** means that *all* pairs of sequences in a cluster must be closer than the threshold.
- **Single linkage** means that a sequence should be included in a cluster if the distance to any other sequence is below the threshold.
- **Average linkage** (similar to UPGMA) distance between the “average sequence” for a cluster with the other cluster (the average is in fact calculated over all pairs).



Maximum distance
Complete linkage



Minimum distance
Single linkage



Average linkage

UPARSE - (U/VSEARCH)

- UPARSE-OTU considers input sequences in order of decreasing abundance. This means that OTU centroids tend to be selected from the more abundant reads, and hence are more likely to be correct biological sequences.
1. All pairs of OTU sequences should have <97% pair-wise sequence identity.
 2. An OTU sequence should be the most abundant within a 97% neighbourhood.
 3. Chimeric sequences should be discarded.
 4. All non-chimeric input sequences should match at least one OTU with $\geq 97\%$ identity.

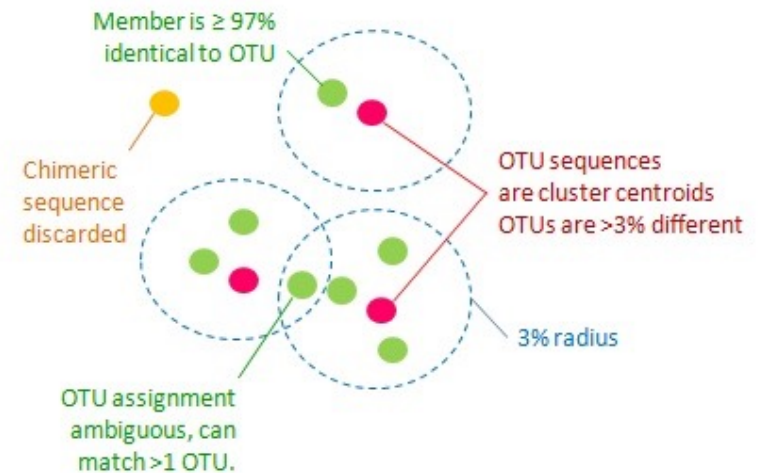
Published: 18 August 2013

UPARSE: highly accurate OTU sequences from microbial amplicon reads

Robert C Edgar ✉

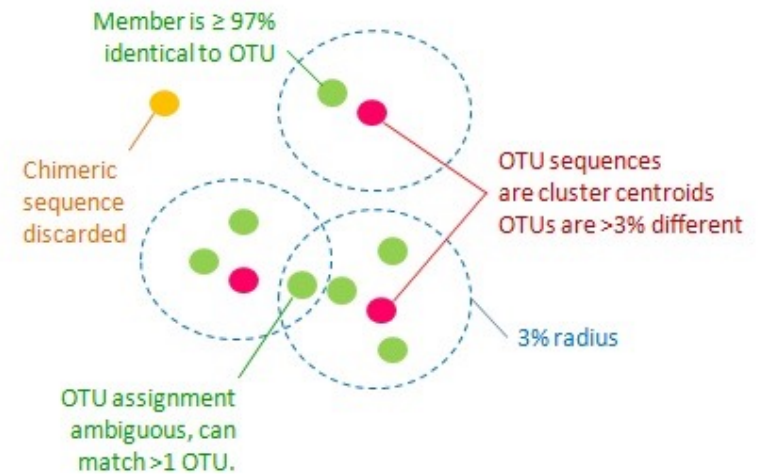
Nature Methods 10, 996–998(2013) | [Cite this article](#)

6924 Accesses | 5903 Citations | 113 Altmetric | [Metrics](#)



UPARSE - (U/VSEARCH)

- Greedy OUT clustering
- Advantages
 - Fast and greedy
 - The similarity threshold is relatively easy to interpret (although not necessarily easy to decide...)
- Disadvantages
 - Arbitrary fixed global clustering threshold. But different lineages evolve at a different rate, which means that there is no single cut-off value for the entire Tree of Life
 - The input order of the amplicons strongly influences the clustering results. Centroid selections are not re-evaluated during the clustering process, this might lead to inaccurately formed OTUs.



SWARM



**Swarm: robust and fast clustering
method for amplicon-based studies**

Frédéric Mahé^{1,2,3}, Torbjørn Rognes^{4,5}, Christopher Quince⁶,
Colomban de Vargas^{1,2} and Micah Dunthorn³

- Fast and exact, two-phased, agglomerative, unsupervised (de novo) single-linkage-clustering algorithm.
- Advantage
 - No global (and arbitrary) clustering threshold
 - The result is not dependent on the input sequence order
- SWARM builds OTUs in two steps
 - An initial set of OTUs is constructed by iteratively agglomerating similar amplicons
 - Amplicon abundances are used to break clusters into sub-OTUs

Greedy cluster vs. SWARM

PeerJ

Swarm: robust and fast clustering method for amplicon-based studies

Frédéric Mahé^{1,2,3}, Torbjørn Rognes^{4,5}, Christopher Quince⁶,
Colomban de Vargas^{1,2} and Micah Dunthorn⁷

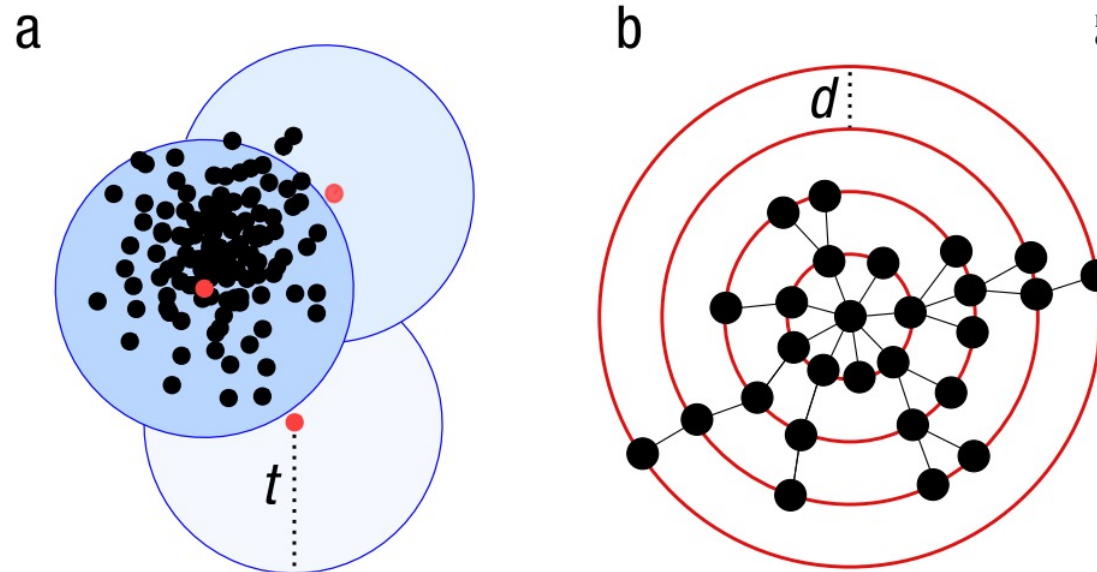


Figure 1 Schematic view of the greedy clustering approach and comparison with swarm. (A) Visualization of the widely used greedy clustering approach based on centroid selection and a global clustering threshold, t , where closely related amplicons can be placed into different OTUs. (B) By contrast, Swarm clusters iteratively by using a small user-chosen local clustering threshold, d , allowing OTUs to reach their natural limits.

DADA2 – Denoising

DADA2: High-resolution sample inference from Illumina amplicon data

Benjamin J Callahan¹, Paul J McMurdie²,
Michael J Rosen³, Andrew W Han², Amy Jo A Johnson² &
Susan P Holmes¹

- An update of DADA (Rosen et al., 2012)
- **Divisive Amplicon Denoising Algorithm**
 - Originally made for 454 data.
- **DADA2** Infers “amplicon sequence variants from Illumina-scale amplicon data data without imposing the arbitrary dissimilarity thresholds that define molecular OTUs”

DADA2 - in R

- Main steps

Step	Function	Explanation
1	<code>filterAndTrim()</code>	Filters and trims an input fastq file(s) (can be compressed) based on several user-definable criteria
2	<code>learnErrors()</code>	Error rates are learned by alternating between sample inference and error rate estimation until convergence.
3	<code>derepFastq()</code>	A custom interface to FastqStreamer for dereplicating amplicon sequences from fastq or compressed fastq files,
4	<code>dada()</code>	The dada function takes as input dereplicated amplicon sequencing reads and returns the inferred composition of the sample (or samples).
5	<code>mergePairs()</code>	This function attempts to merge each denoised pair of forward and reverse reads, rejecting any pairs which do not sufficiently overlap or which contain too many (>0 by default) mismatches in the overlap region.
6	<code>makeSequenceTable()</code>	This function constructs a sequence table (analogous to an OTU table) from the provided list of samples.
7	<code>removeBimeraDenovo()</code>	screen for and remove chimeras
8	<code>assignTaxonomy()</code>	<code>assignTaxonomy</code> implements the RDP Naive Bayesian Classifier algorithm described in Wang et al. Applied and Environmental Microbiology 2007,

DADA2

- Advantages (according to the *developers*, aka *the selling point...*)
 - **Resolution:** DADA2 infers “amplicon sequence variants (ASVs)” from amplicon data
 - **Accuracy:** DADA2 reports fewer false positive sequence variants than other methods report false OTUs.
 - **Comparability:** The ASVs output by DADA2 can be directly compared between studies, without the need to reprocess the pooled data.
 - **Computational Scaling:** The computing time of DADA2 scales linearly with sample number, and memory requirements are essentially flat.

DADA2

- How it works:
 - Infers an error model from the sequencing data.
 - The error model incorporates **(qualitative) abundance**
 - Takes **fastq** as input
 - These are used to divide the amplicon reads into partitions, and calculate the expected error rate for the partition(s) given that these are representatives of true sample
 - An iterative process that runs until convergence (i.e. until the cut-off parameters for expected errors are met, to put it another way: that it is unlikely that a smaller partition would represent true sequences)

DADA2

- The error model incorporates quality information, which is usually ignored by other methods (after filtering).
- The error model incorporates quantitative abundances, many other methods use abundance ranks (if they use abundance at all).
- The error model have different error rates for different substitutions,
 - eg. A->C, is different from A->G, which is different from A->T etc. whereas other methods merely count the mismatches.
- DADA2 can parameterize its error model from the data itself, rather than relying on previous datasets that may or may not reflect the PCR and sequencing protocols used in your study.

Callahan et al., Nat.Meth. (2016); Callahan et al., ISMEj (2017)

sample
sequences



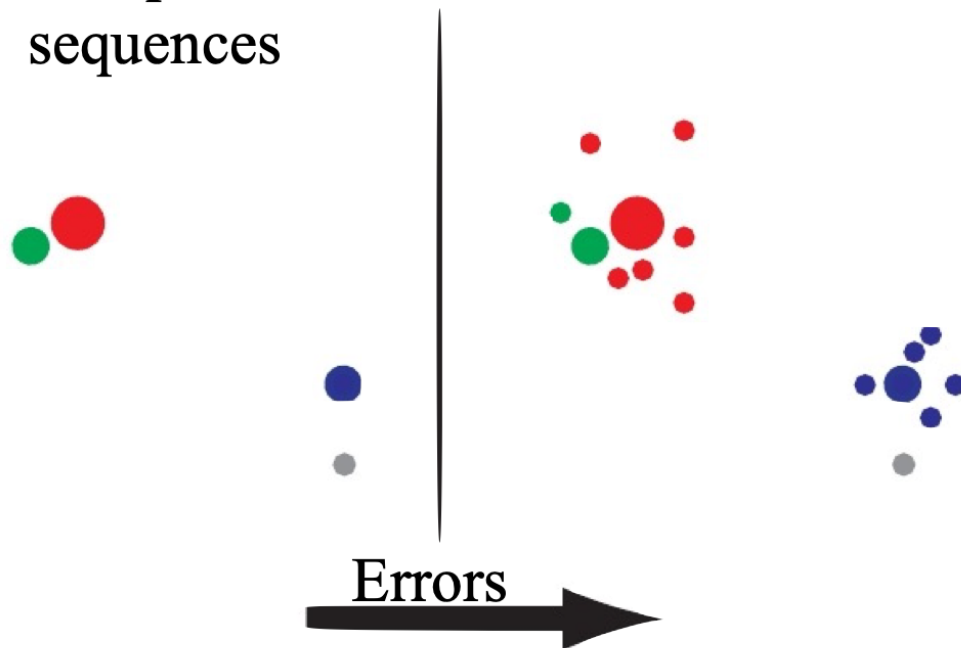
sample
sequences

PCR → amplicon reads



sample
sequences

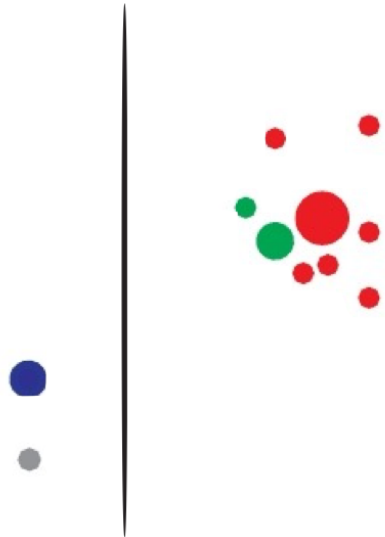
PCR → amplicon reads



sample
sequences



PCR → amplicon reads



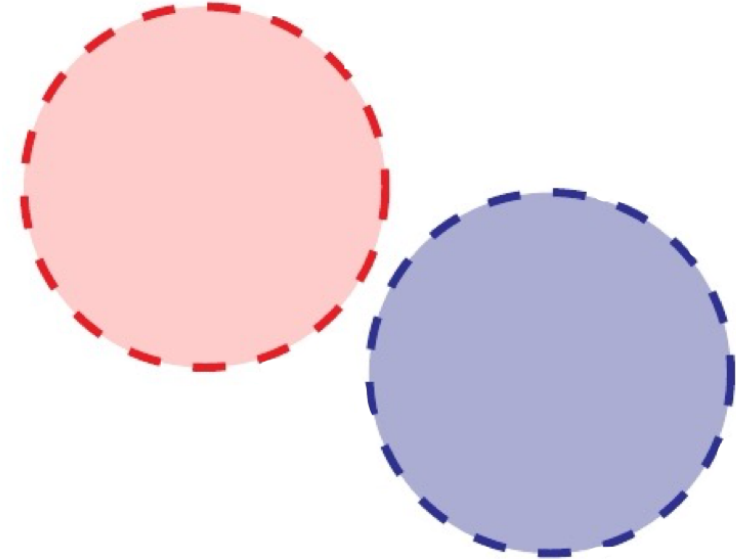
Errors



DADA2



OTUs



Make OTUs



DADA2

- The core denoising algorithm in the DADA2 is built on a model of the errors in Illumina-sequenced amplicon reads.
 - A function for PacBio has been implemented
- These can be modelled based on the data
- The error model quantifies the rate λ_{ji} at which an amplicon read with sequence i is produced from sample sequence j as a function of sequence composition and quality.

DADA2

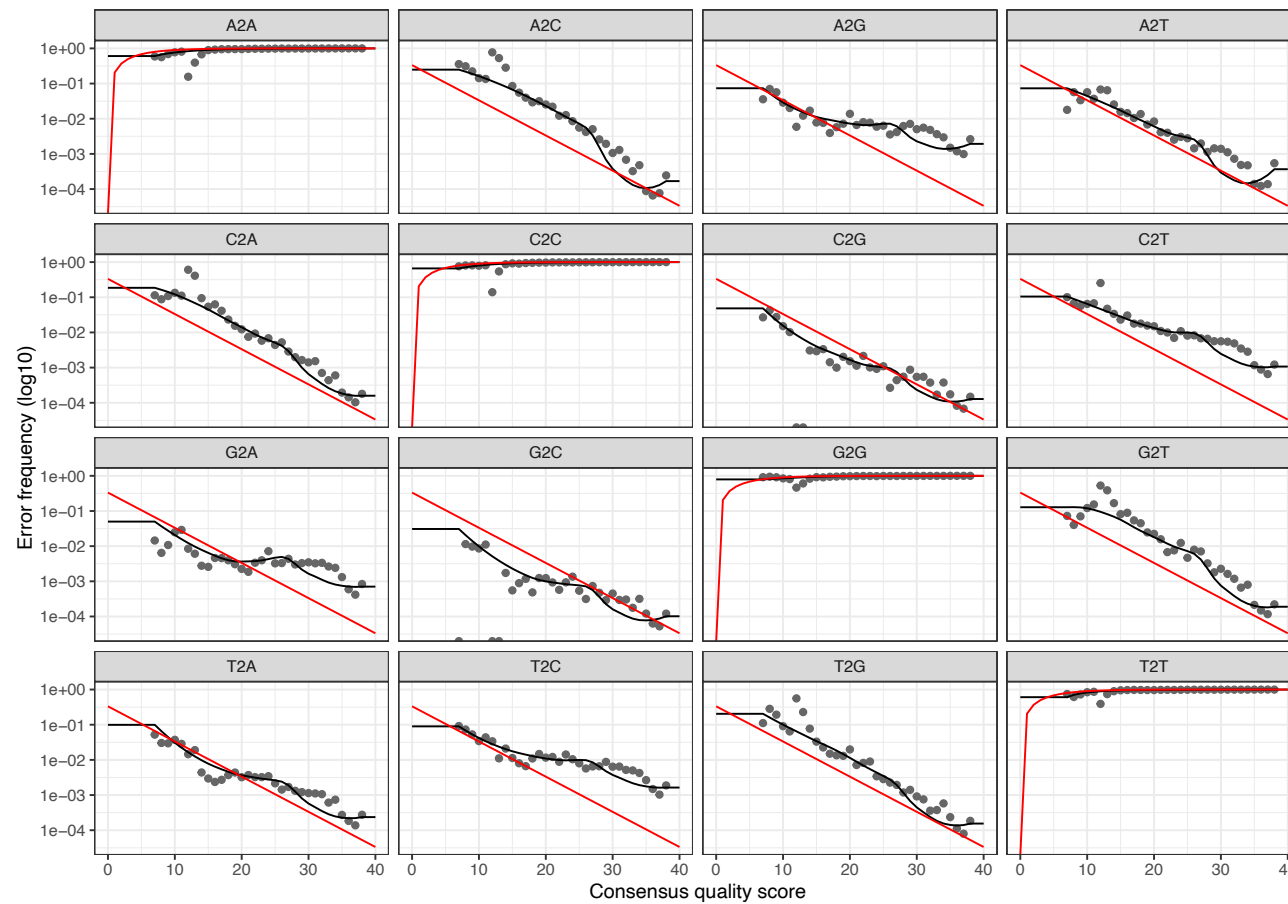
- **Build error model**

- Assumption: Errors occurs independently within a read
- Assumption: Errors occurs independently between reads
- The rate at which an amplicon read with sequence i is produced from sample sequence j is the product of the transition probabilities of the aligned nucleotides:

$$\lambda_{j \rightarrow i} = \prod_{l=0}^L p(j(l) \rightarrow i(l), q(l))$$

- The transition probability between aligned nucleotides depend on the original nucleotide, substituting nucleotide, and associated quality score, for example, $p(A \rightarrow C, 35)$.
- After sequence alignment, the error rate λ_{ji} is calculated and stored.

DADA2 - Error model



Bin Emoji

N 🚫

2-9 🦴

10-19 🗿

20-24 ⚠️

25-29 😊

30-34 😏

35-39 😎

≥ 40 🤖

<https://fastq.com/>

DADA2

- **The abundance p-value.**

- The number of amplicon reads with sequence i that will be produced from sample sequence j is Poisson distributed with expectation equal to an error rate λ_{ji} multiplied by the expected reads of sample sequence j

$$p_A(j \rightarrow i) = \frac{1}{1 - \rho_{\text{pois}}(n_j \lambda_{ji}, 0)} \sum_{a=a_i}^{\infty} \rho_{\text{pois}}(n_j \lambda_{ji}, a)$$

- A low pA indicates that there are more reads of sequence i than can be explained by errors introduced during the amplification and sequencing of n_j copies of sample sequence j .

DADA2

- **The divisive partitioning algorithm.**
 - Amplicon reads with the same sequence are grouped into unique sequences with an associated abundance and consensus quality profiles (aka dereplicated).
 - The algorithm is initiated by placing all sequences in a single partition with the most abundant as the centre.
 - All unique sequences are compared to the center
 - Calculate error rates
 - Calculate abundance p-value

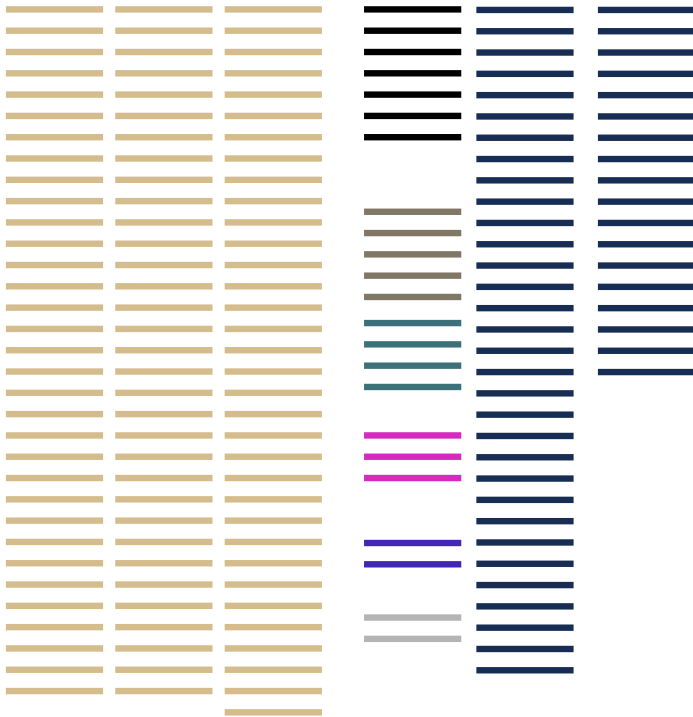
DADA2

- **The divisive partitioning algorithm.**









- If the smallest p-value falls below a threshold $OMEGA_A = 1e-40$ (default) a new partition is formed
- After the new partition is formed, every unique sequence is allowed to join the partition most likely to have produced it
- Repeat until all unique sequences are consistent with being produced by the amplicon sequence at the center of their partition.

DADA2 - dereplication

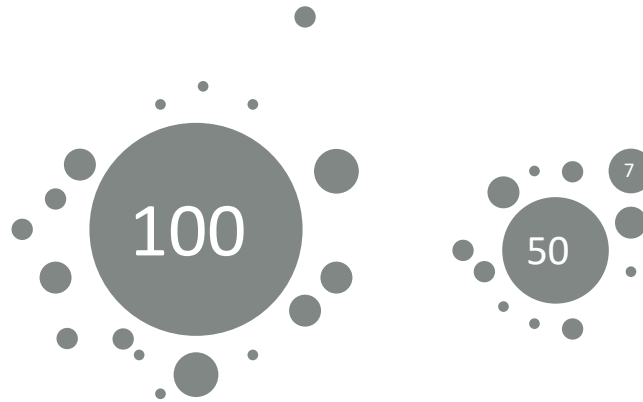
"raw" reads



dereplicate

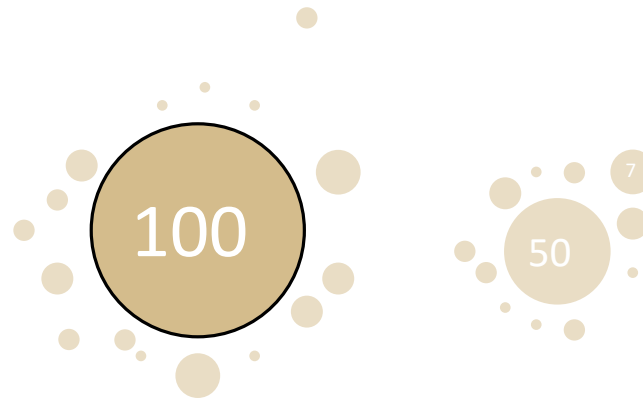
unique sequences	abundance	mean-Q
	100	32
	50	32
	7	20
	5	...
	4	...
	3	...
	2	...
	2	...

DADA2



Initial guess: one real sequence + errors

DADA2



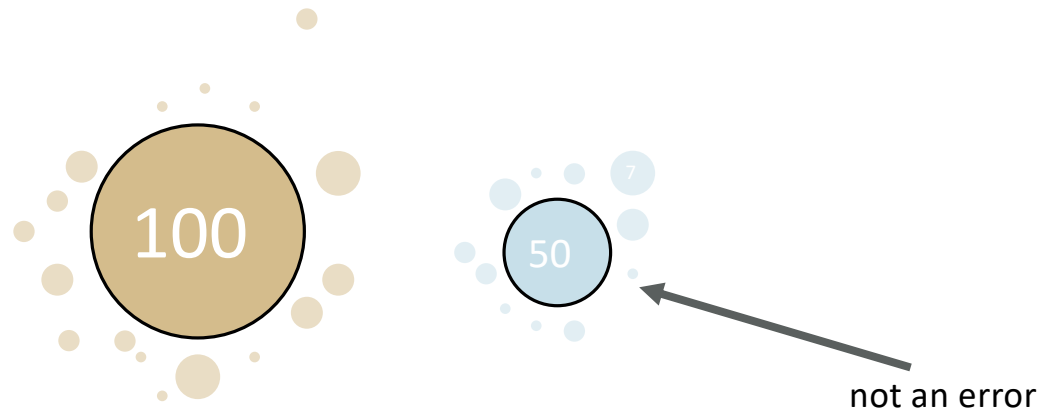
Infer initial error model under this assumption

$\Pr(i \rightarrow j, q) =$

	A	C	G	T
A	0.97	10^{-2}	10^{-2}	10^{-2}
C	10^{-2}	0.97	10^{-2}	10^{-2}
G	10^{-2}	10^{-2}	0.97	10^{-2}
T	10^{-2}	10^{-2}	10^{-2}	0.97

DADA2

“divide the amplicon reads into partitions, and calculate the expected error rate for the partition(s) given that these are representatives of true sample-sequence+errors”



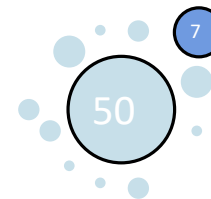
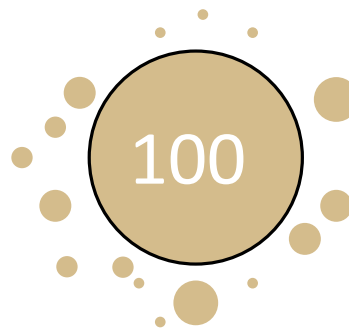
Update the model.

$\text{Pr}(i \rightarrow j) =$

	A	C	G	T
A	0.997	10^{-3}	10^{-3}	10^{-3}
C	10^{-3}	0.997	10^{-3}	10^{-3}
G	10^{-3}	10^{-3}	0.997	10^{-3}
T	10^{-3}	10^{-3}	10^{-3}	0.997

DADA2

not an error



not an error

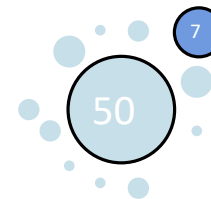
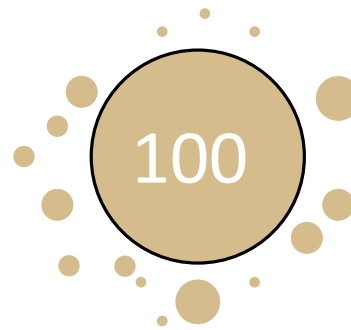
Update model again

$\text{Pr}(i \rightarrow j) =$

	A	C	G	T
A	0.998	1×10^{-4}	2×10^{-3}	2×10^{-4}
C	6×10^{-5}	0.998	3×10^{-4}	1×10^{-3}
G	1×10^{-4}	1×10^{-4}	0.998	6×10^{-5}
T	2×10^{-4}	2×10^{-3}	1×10^{-4}	0.998

DADA2

not an error



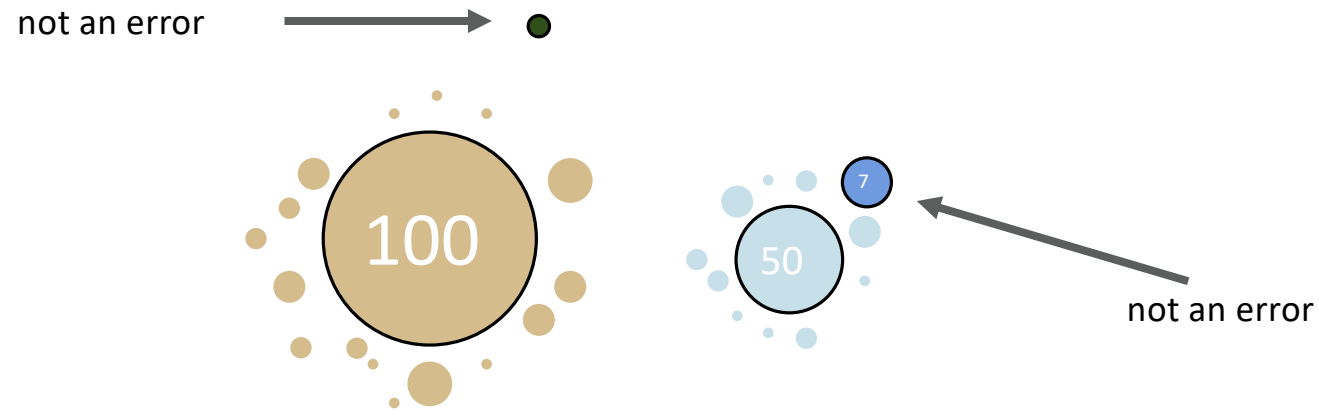
not an error

Repeat until all unique sequences are consistent with being produced by the amplicon sequence at the center of their partition

$\text{Pr}(i \rightarrow j) =$

	A	C	G	T
A	0.998	1×10^{-4}	2×10^{-3}	2×10^{-4}
C	6×10^{-5}	0.998	3×10^{-4}	1×10^{-3}
G	1×10^{-4}	1×10^{-4}	0.998	6×10^{-5}
T	2×10^{-4}	2×10^{-3}	1×10^{-4}	0.998

DADA2



Repeat until all unique sequences are consistent with being produced by the amplicon sequence at the center of their partition

- all abundance p-values are greater than OMEGA_A;
- i.e., all unique sequences are consistent with being produced by amplicon sequencing at the center of their partition.
- The inferred composition of the sample is then the set of central sequences and the corresponding total abundances of those partitions

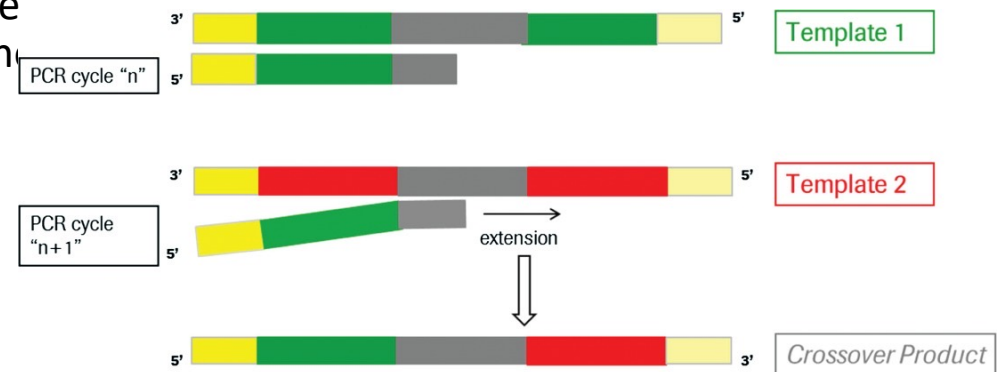
Create sequence table

- Merge forward/reverse reads
 - Illumina data is often pair-ended,
 - Forward and reverse reads (R1 and R2) are processed independently until this point.
 - R1 and R2 have (very) different error-profiles
- Count the abundance of merged pair reads per sample.

Seq.	Sample 1	Sample 2	Sample 3	...	Sample n
Seq 1	20	5463	7763	..	1
Seq 2	0	0	176	..	667
Seq 3	343	12	256	..	57
Seq n	136	673	937	..	79

Remove chimeras

- After denoising it is advisable to check for chimeras.
- Chimeras are formed when an incomplete sequence functions a primer in the next amplification step. The resulting read contains half of one sample sequence and half of another.
- The function *removeBimeraDenovo* will remove chimeric sequences.
- Chimeric sequences are identified if they can be reconstructed by combining a left-segment and a right-segment from two more abundant “parent” sequences



Huh? No clustering?

- You might still need to cluster after denosing with DADA2
- This depends on
 - the barcoding region,
 - the organisms,
 - the systems under study
 - the research question
- More on this on this in the coming days...

DADA2 - in R

- Main steps

Step	Function	Explanation
1	<code>filterAndTrim()</code>	Filters and trims an input fastq file(s) (can be compressed) based on several user-definable criteria
2	<code>learnErrors()</code>	Error rates are learned by alternating between sample inference and error rate estimation until convergence.
3	<code>derepFastq()</code>	A custom interface to FastqStreamer for dereplicating amplicon sequences from fastq or compressed fastq files,
4	<code>dada()</code>	The dada function takes as input dereplicated amplicon sequencing reads and returns the inferred composition of the sample (or samples).
5	<code>mergePairs()</code>	This function attempts to merge each denoised pair of forward and reverse reads, rejecting any pairs which do not sufficiently overlap or which contain too many (>0 by default) mismatches in the overlap region.
6	<code>makeSequenceTable()</code>	This function constructs a sequence table (analogous to an OTU table) from the provided list of samples.
7	<code>removeBimeraDenovo()</code>	screen for and remove chimeras
8	<code>assignTaxonomy()</code>	<code>assignTaxonomy</code> implements the RDP Naive Bayesian Classifier algorithm described in Wang et al. Applied and Environmental Microbiology 2007,

DADA2 - hands on session

- The goal is to get from a set of fastq-files (trimmed) to an OTU-table that can be used for downstream analysis.
- Scripts and setup on GitHub
- https://github.com/krabberod/BIO9905MERG1_V25
- Dataset for the run-through:
 - Selected samples from Blanes Bay Marine Observatory (BBMO) near Barcelona
 - Mini-time series: January, April, July and October for 2004 and 2005.
 - Subsample of a larger dataset: doi [10.1186/s40793-022-00417-1](https://doi.org/10.1186/s40793-022-00417-1)

Research article | [Open Access](#) | Published: 07 May 2022

Long-term patterns of an interconnected core marine microbiota

[Anders K. Krabberød](#) , [Ina M. Deutschmann](#), [Marit F. M. Bjorbækmo](#), [Vanessa Balagué](#), [Caterina R. Giner](#), [Isabel Ferrera](#), [Esther Garcés](#), [Ramon Massana](#), [Josep M. Gasol](#) & [Ramiro Logares](#) 

Environmental Microbiome 17, Article number: 22 (2022) | [Cite this article](#)

2206 Accesses | 5 Citations | 22 Altmetric | [Metrics](#)

DADA2

- Resources:
- The developer has a nice tutorial
 - <https://benjjneb.github.io/dada2/>
 - <https://benjjneb.github.io/dada2/tutorial.html>
- DADA2 for ITS
 - https://benjjneb.github.io/dada2/ITS_workflow.html