

DADA2

(with comparison to some other methods)

Anders K. Krabberød

a.k.krabberod@ibv.uio.no

Illumina seq. and amplicon data

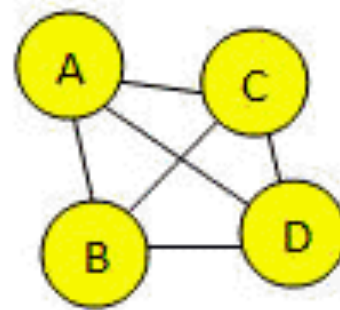
- Illumina HiSeq and MiSeq sequencing has been used extensively for amplicons sequencing in the last years
 - High yields with relative low cost
 - HiSeq (and NovaSeq) short fragments 2*150bp
 - MiSeq a bit longer reads 2*300bp
- Low error rate, but due to the high number of sequences generated errors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

OTUs

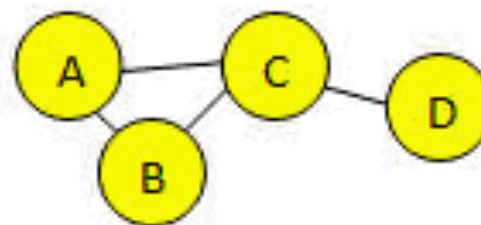
- Clusters of similar amplicons is called OTUs (operational taxonomic units)
 - Proxies for species, ecotypes or another functional unit
 - Clustered based on dissimilarity cut-off
 - But what cut-off to choose?
 - And how to calculate the similarity/dissimilarity?
- One solution is UPGMA-style clustering.
 - I.e. clustering at a fixed sequence distance
 - For instance 97%, which was often used for 18S V4

Clustering types

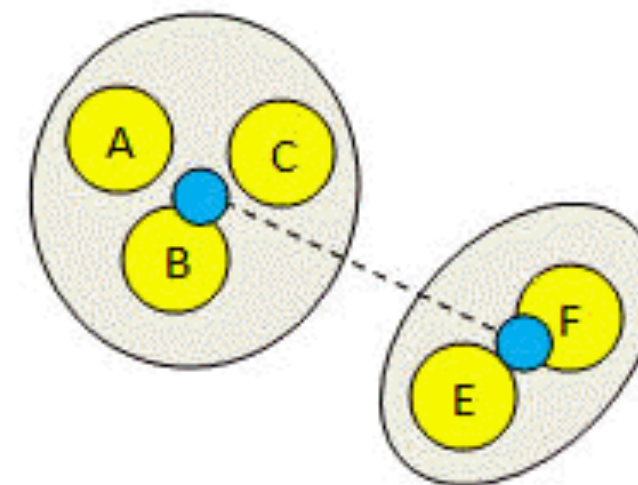
- **Average linkage** (similar to UPGMA) distance between the “average sequence” for a cluster with the other cluster (the average is in fact calculated over all pairs).
- **Complete linkage** means that *all* pairs of sequences in a cluster must be closer than the threshold.
- **Single linkage** means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



Maximum distance
Complete linkage



Minimum distance
Single linkage

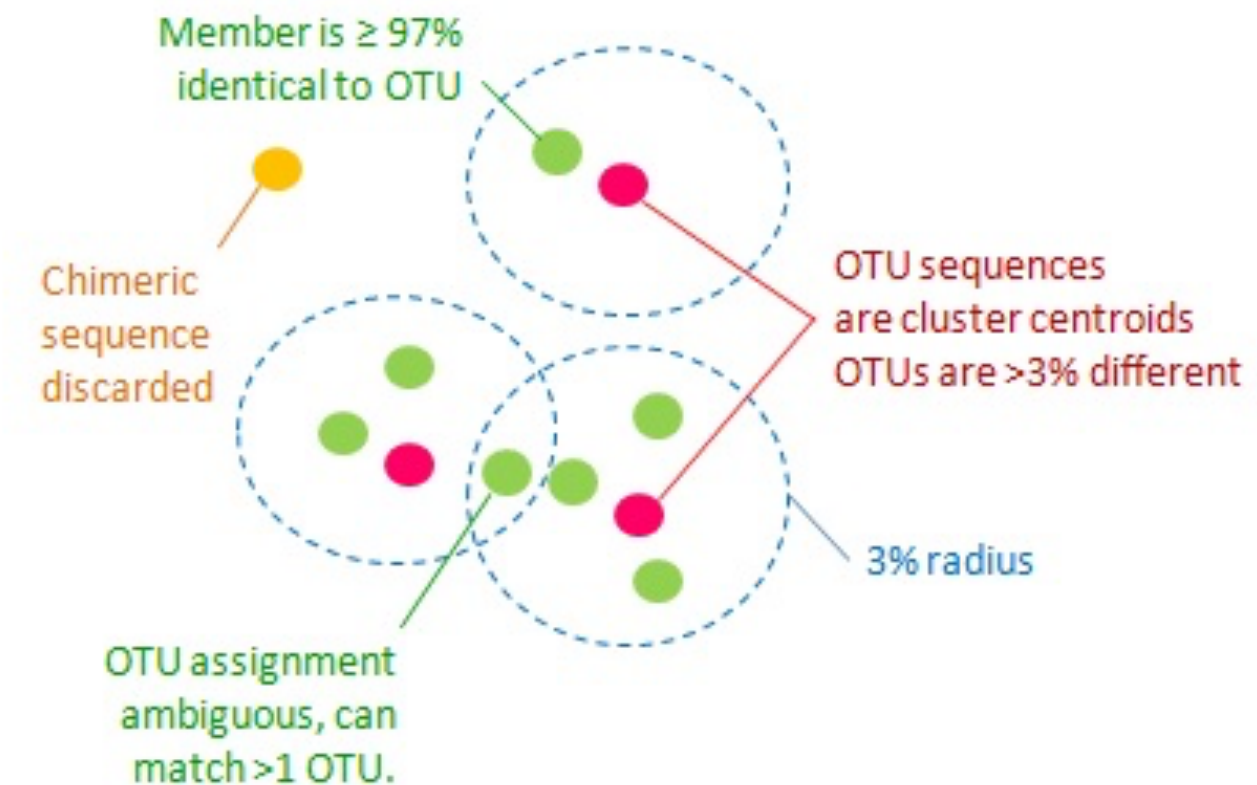


Average linkage

UPARSE - (U/VSEARCH)

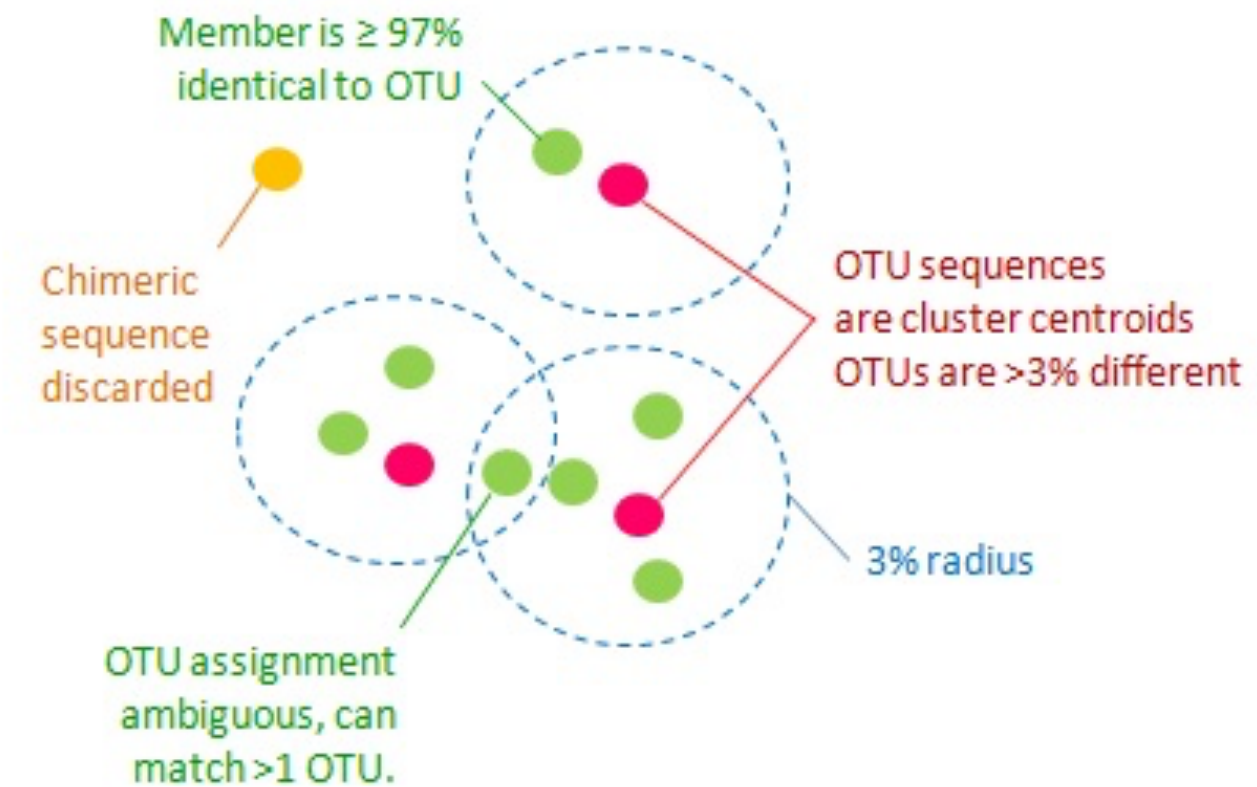
- UPARSE-OTU considers input sequences in order of decreasing abundance. This means that OTU centroids tend to be selected from the more abundant reads, and hence are more likely to be correct biological sequences.

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.



UPARSE

- UPARSE implements a greedy algorithm that performs OTU clustering
- Advantages
 - Fast and greedy
 - Similarity threshold is relatively easy to interpret (although not necessarily easy to decide...)
- Disadvantages
 - Arbitrary fixed global clustering threshold. But different lineages evolve at 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 Dunthorn³

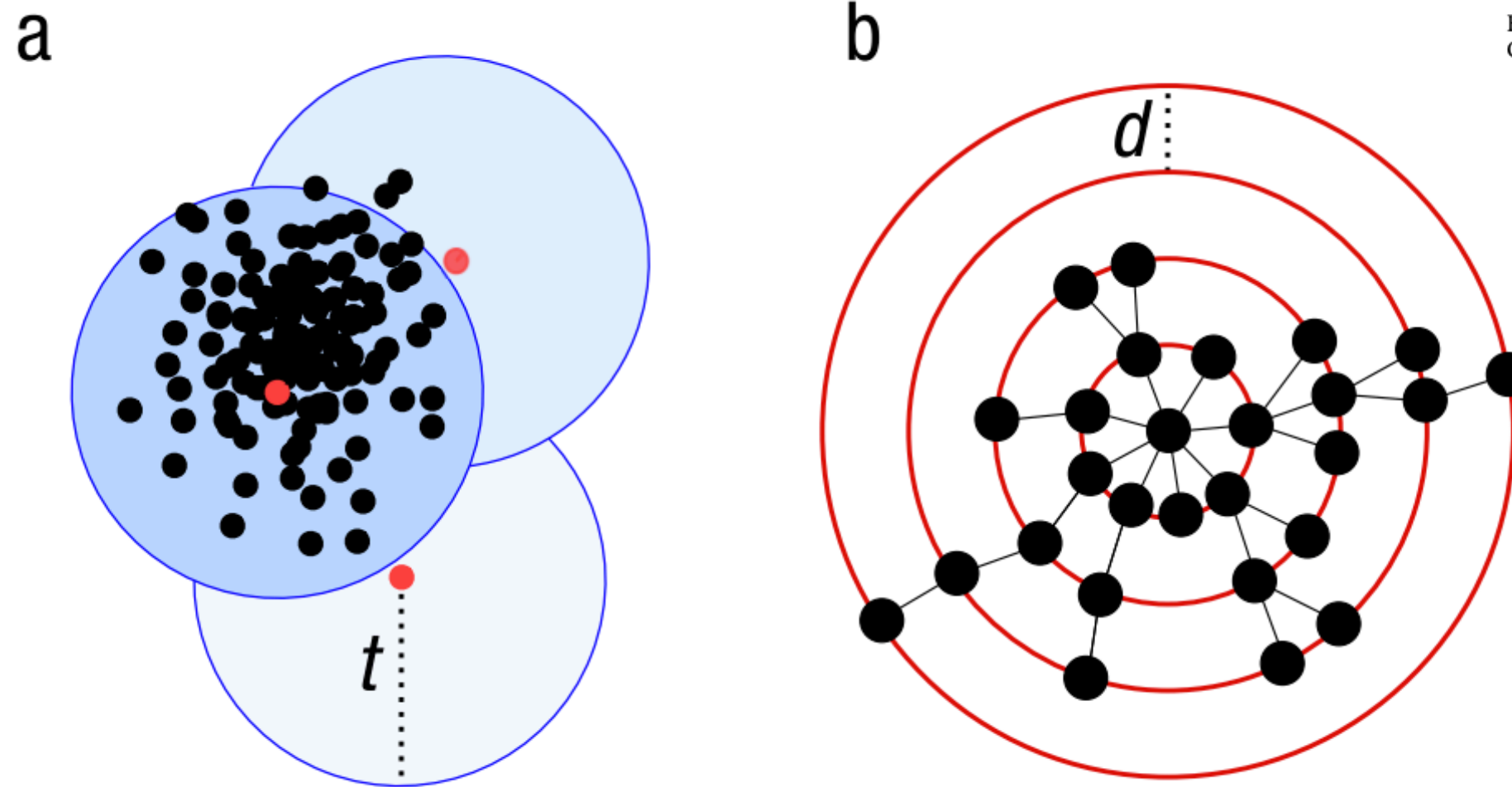
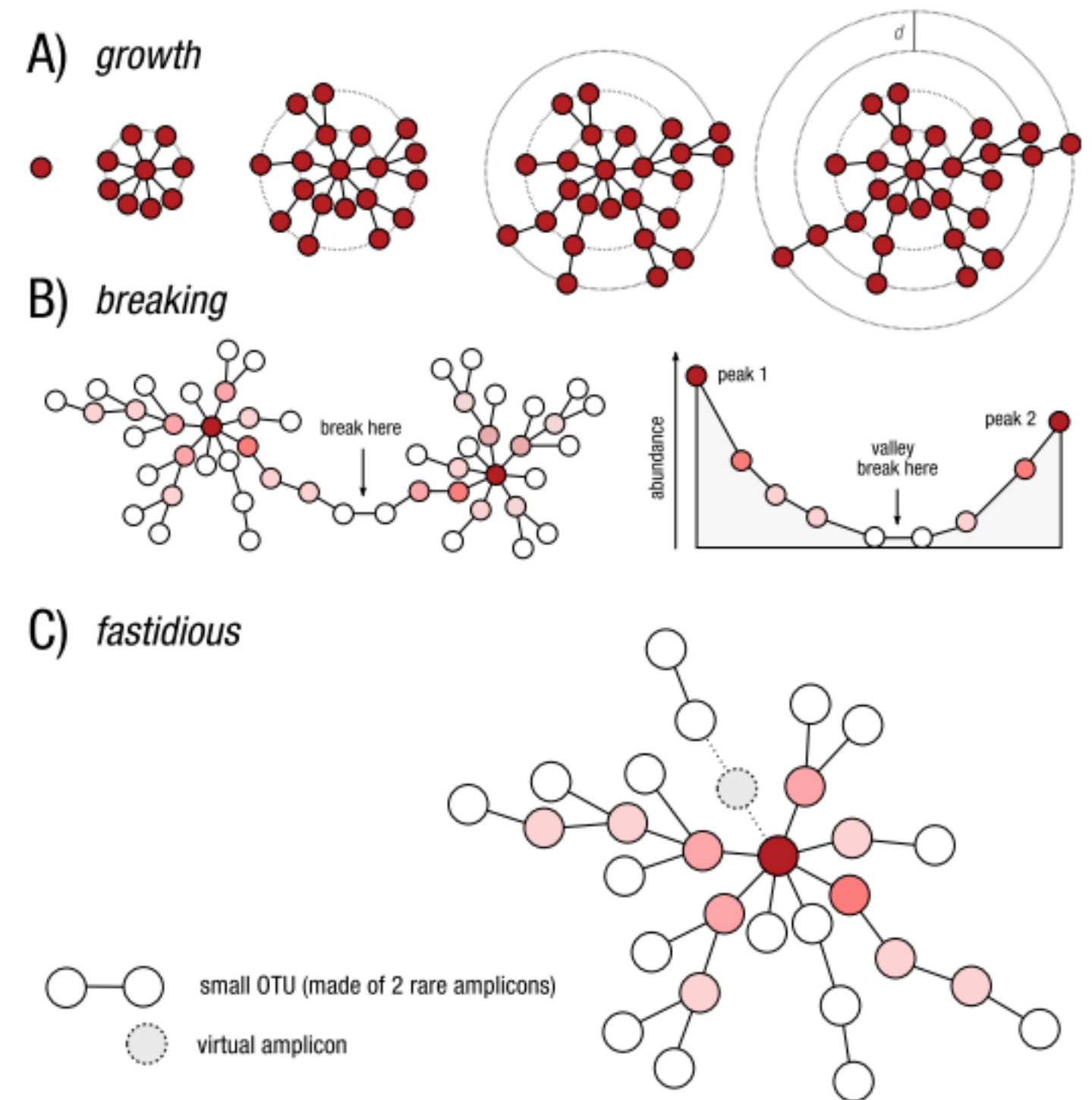


Figure 1 Schematic view of the greedy clustering approach and comparison with swarm. (A) Visualization of the widely used greedy clustering approach based on centroid selection and a global clustering threshold, t , where closely related amplicons can be placed into different OTUs. (B) By contrast, Swarm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

SWARM

- (A) Swarm clusters amplicons iteratively by using a small user-chosen local threshold, d , allowing OTUs to grow to their natural limits, where no other amplicons can be added.
- (B) Swarm takes into account the abundance of each amplicon to produce higher resolution clusters, by not allowing the formation of amplicon chains. The darker the red, the higher the abundance.
- (C) The fastidious option avoids under-grouping (e.g., the production of small OTUs such as singletons and doubletons) by postulating the existence of virtual linking amplicons to graft smaller OTUs onto larger ones.



Swarm v2: highly-scalable and high-resolution amplicon clustering

Frédéric Mahé¹, Torbjørn Rognes^{2,3}, Christopher Quince⁴, Colomán de Vargas^{5,6} and Micah Dunthorn¹

¹ Department of Ecology, Technische Universität Kaiserslautern, Kaiserslautern, Germany

² Department of Informatics, University of Oslo, Oslo, Norway

³ Department of Microbiology, Oslo University Hospital, Rikshospitalet, Oslo, Norway

⁴ Warwick Medical School, University of Warwick, Warwick, United Kingdom

⁵ UMR 7144, EPEP-Evolution des Protistes et des Écosystèmes Pélagiques, Station Biologique de Roscoff, CNRS, Roscoff, France

⁶ UMR7144 Station Biologique de Roscoff, Sorbonne Universités, UPMC Univ Paris 06, Roscoff, France

DADA2

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

- Advantages (according to themselves, aka the selling point...)
 - **Resolution:** DADA2 infers exact amplicon sequence variants (ASVs) from amplicon data, resolving biological differences of even 1 or 2 nucleotides.
 - **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 compute time of DADA2 scales linearly 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) abundances
 - Which 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-sequence+errors
 - Iterative process that runs until convergence (i.e. the until the cut-off parameters for expected errors are met)

DADA2

- The error model incorporates quality information, which is usually ignored by other methods (after filtering).
- The error model incorporates quantitative abundances, whereas most 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 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.

sample
sequences

amplicon reads



Errors



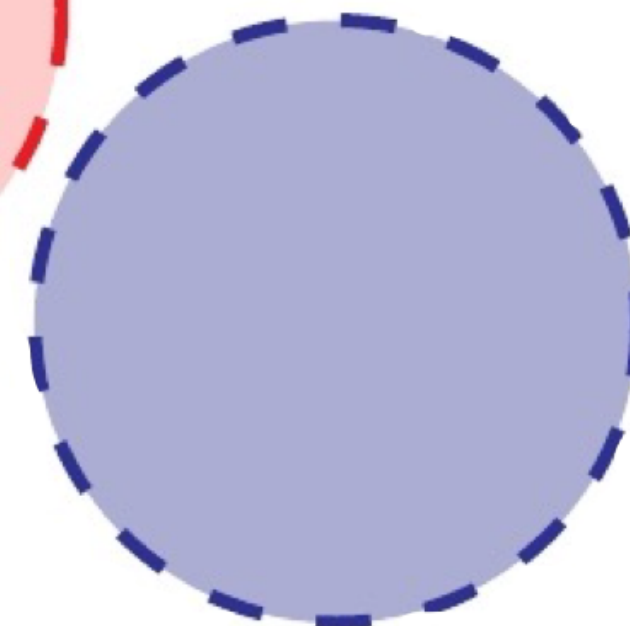
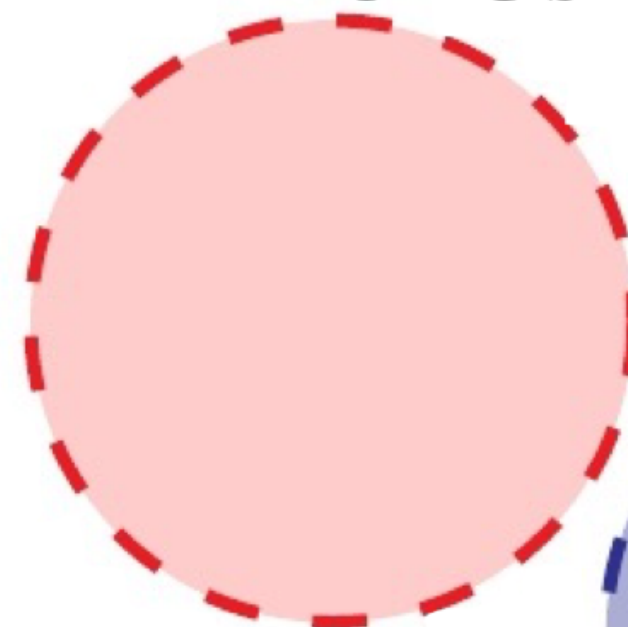
sample
sequences



amplicon reads



OTUs



Errors



Make OTUs



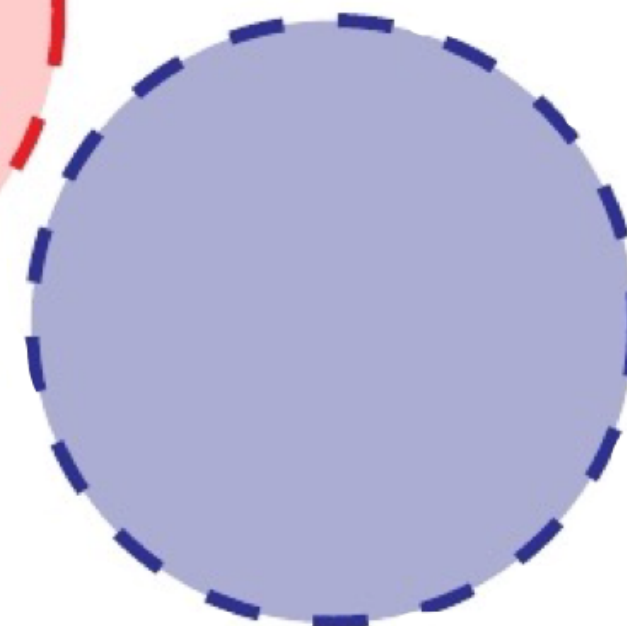
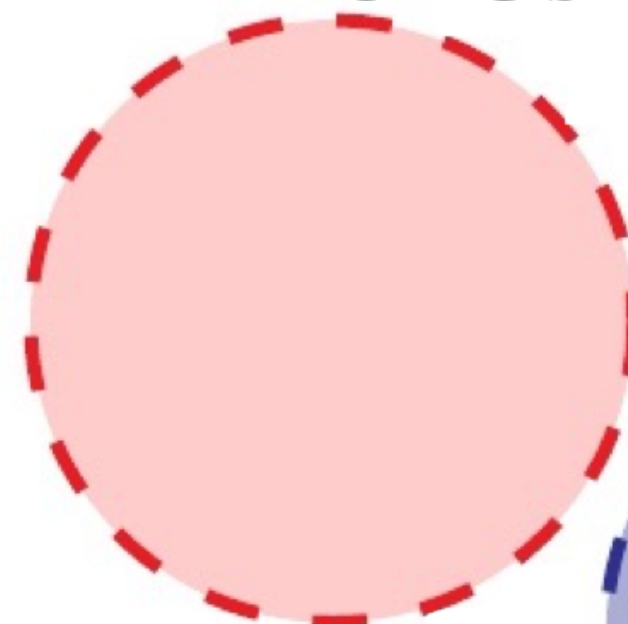
sample
sequences



amplicon reads



OTUs



Errors



Make OTUs



DADA2



DADA2

- The core denoising algorithm in the DADA2 is built on a model of the errors in Illumina-sequenced amplicon reads.
- 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

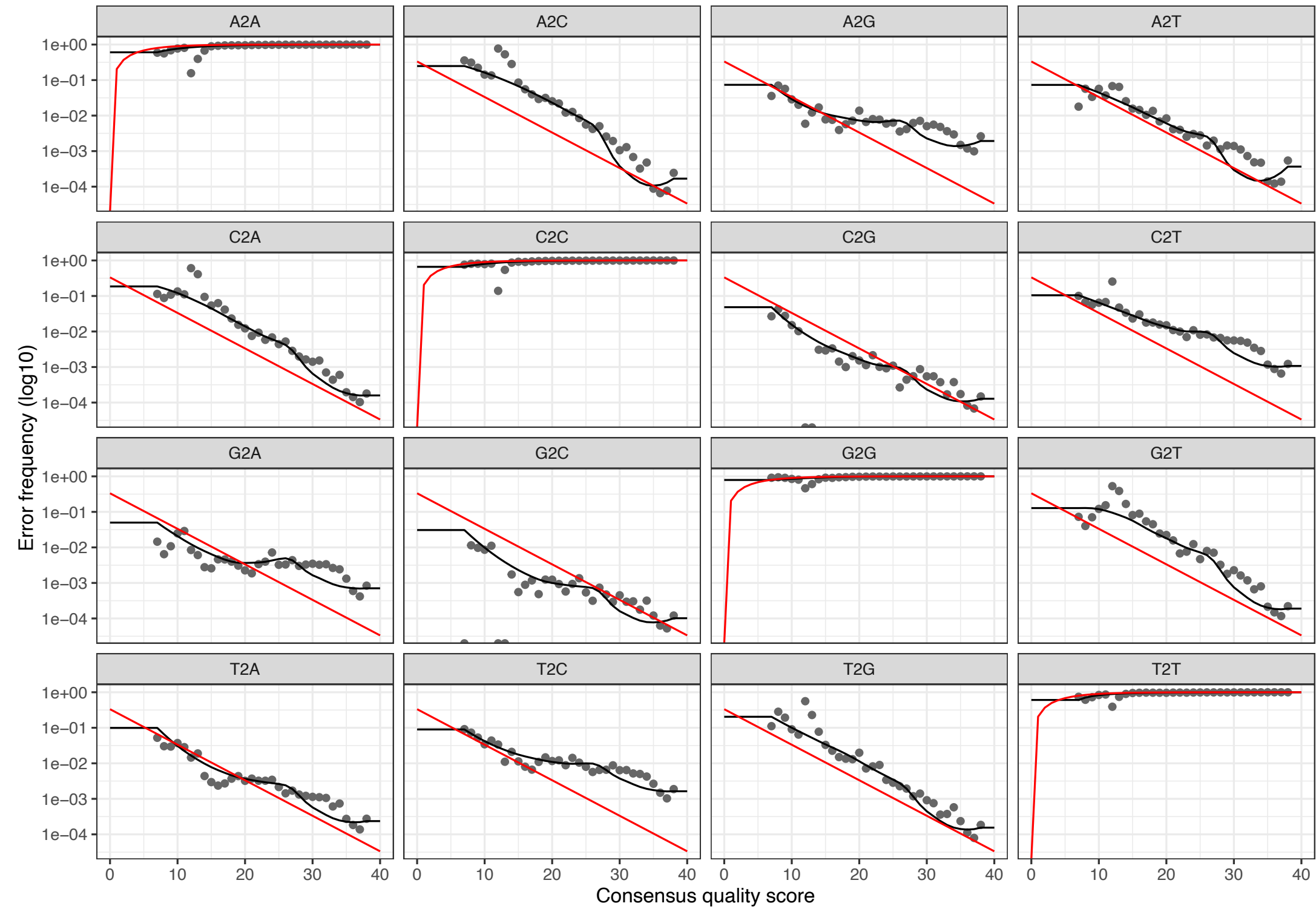
- **Build error model**

- 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:

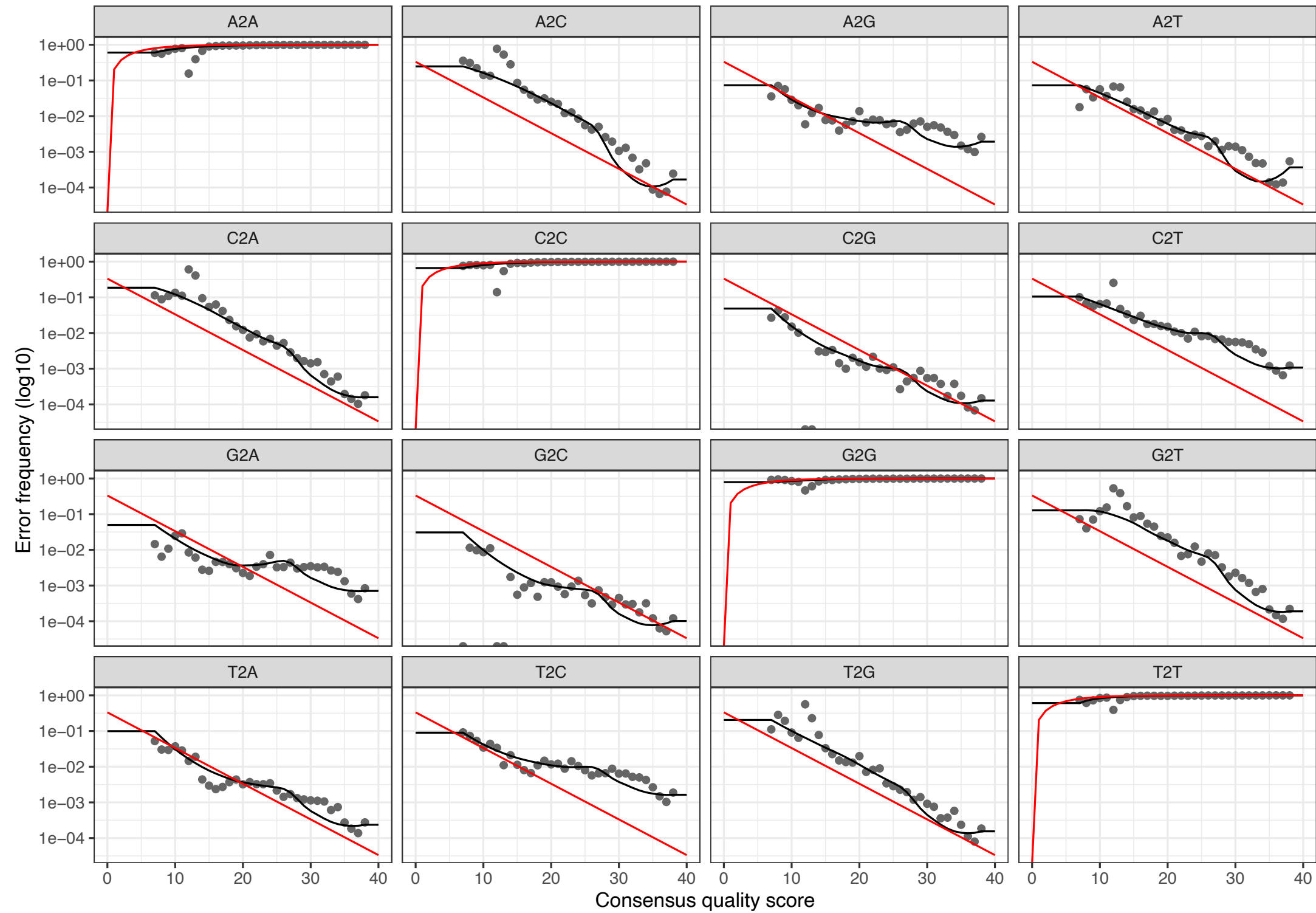
$$\lambda_{j \rightarrow i} = \prod_{l=0}^L p(j(l) \rightarrow i(l), q(l)))$$

- 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



DADA2 - Error model



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 compare to the centre
 - Calculate error rates
 - Calculate abundance p-value

DADA2

- **The divisive partitioning algorithm.**
 - If the smallest p-value falls below a threshold ($\text{OMEGA_A} = 1e-40$ (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 amplicon sequencing the center of their partition.

DADA2 - dereplication

“raw” reads











dereplicate

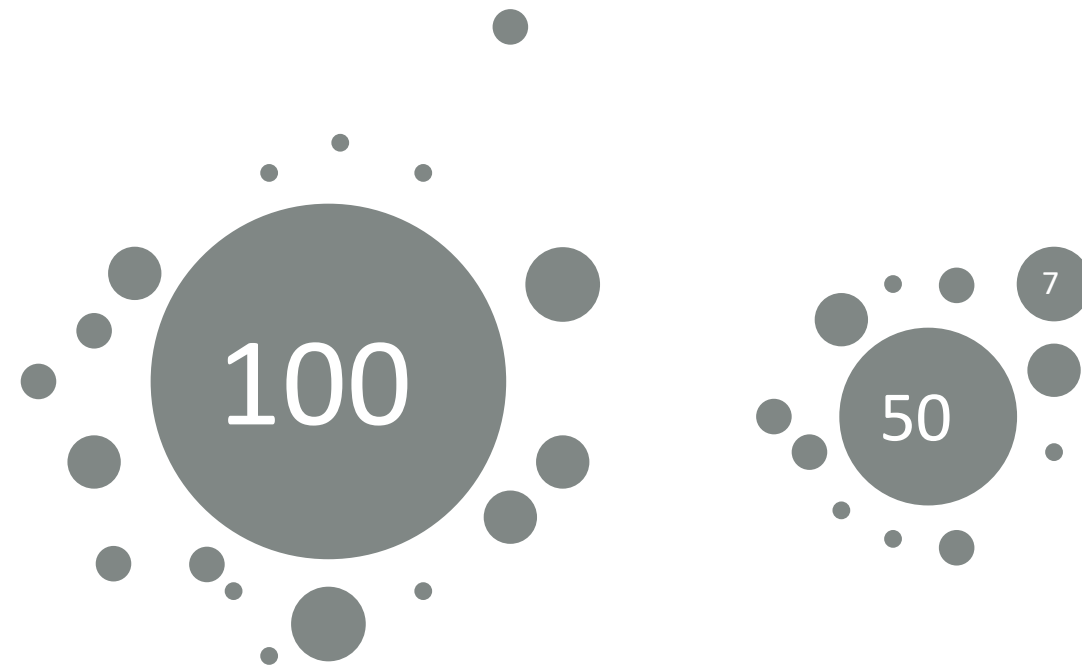
unique
sequences

abundance

mean-Q

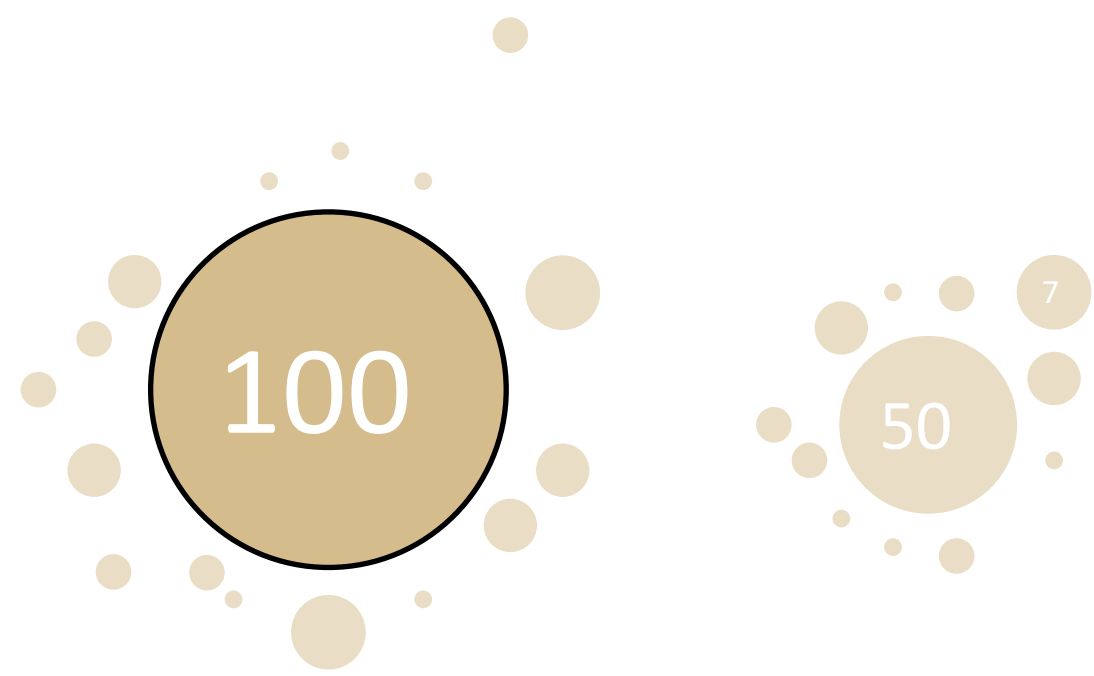
	100	32
	50	32
	7	20
	5	...
	4	...
	3	...
	2	...
	2	...

DADA2 - error model



Initial guess: one real sequence + errors

DADA2

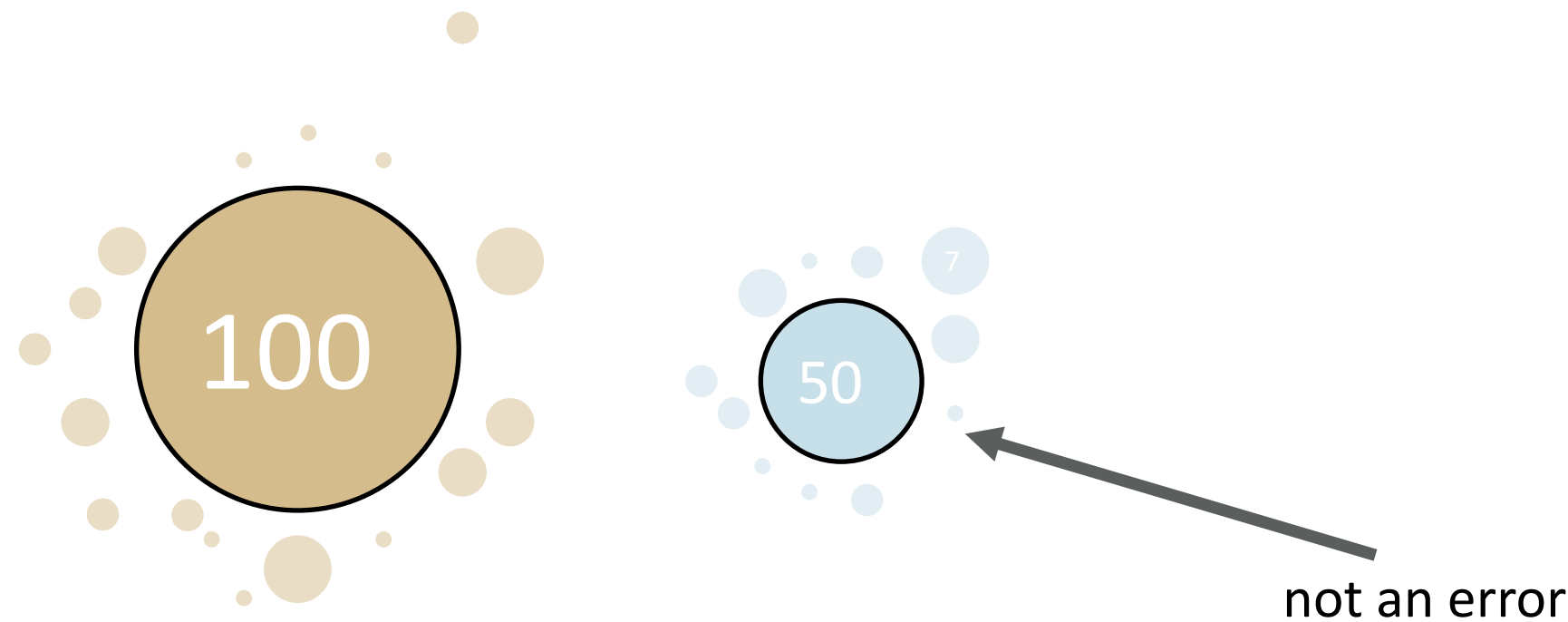


Infer initial error model under this assumption

Pr(i → j) =

	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

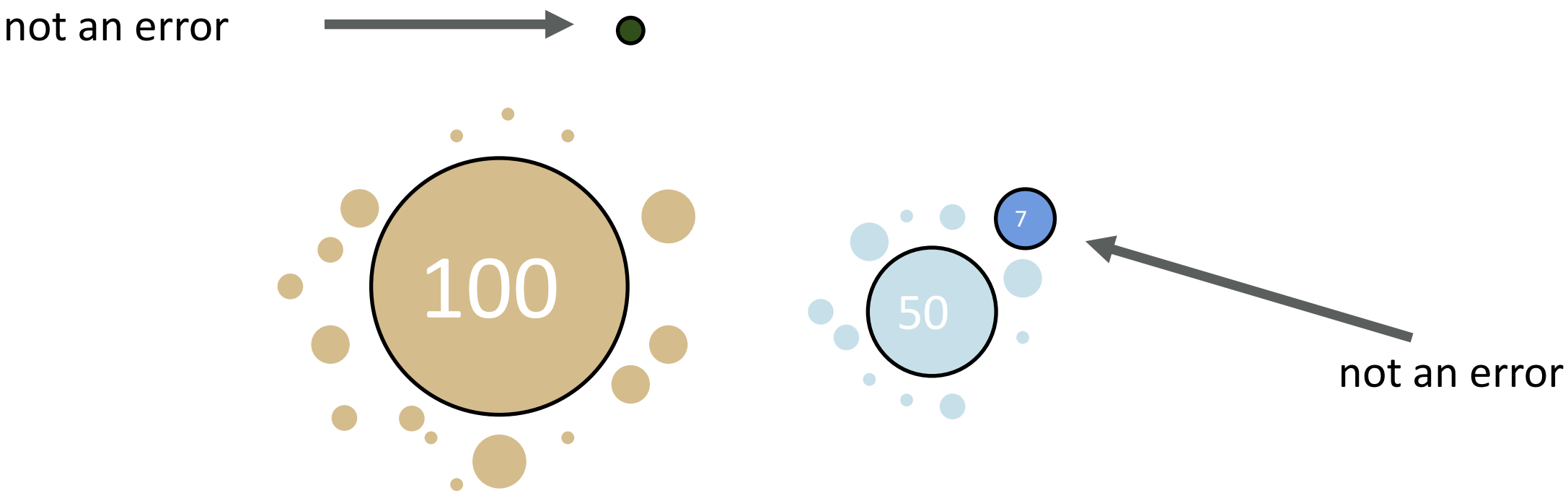


Update the model.

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

	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



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 - 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_V21
- 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: Preprint on BioRxiv:
 - doi.org/10.1101/2021.03.18.435965

Long-term patterns of an interconnected core marine microbiota

[Comment on this paper](#)

 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

doi: <https://doi.org/10.1101/2021.03.18.435965>

This article is a preprint and has not been certified by peer review [what does this mean?].

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