ETH zürich

551-1119-00L Microbial Community Genomics

16S rRNA amplicon data analysis using DADA2

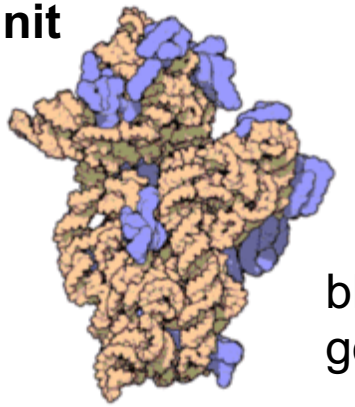
The 16S rRNA gene

- Part of prokaryotic ribosomes

50S large subunit (33 proteins)	5S: 120 nt 23S : 2906 nt
30S small subunit (22 proteins)	**16S: 1542 nt**

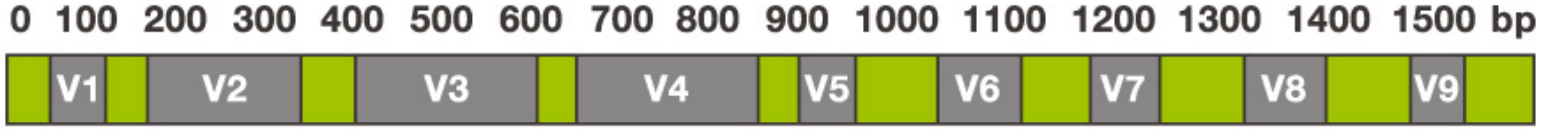
- 16S rRNA present in all prokaryotes

30S small subunit

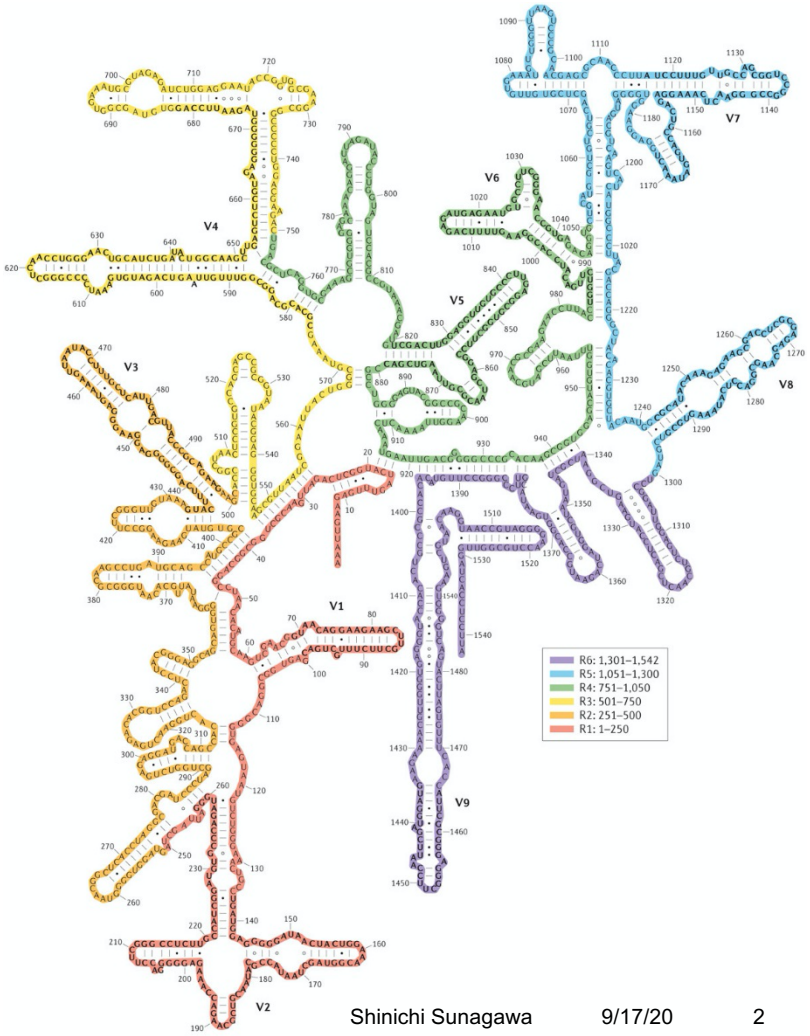


blue: ribosomal proteins
gold: 16S rRNA

- conserved regions and variable regions

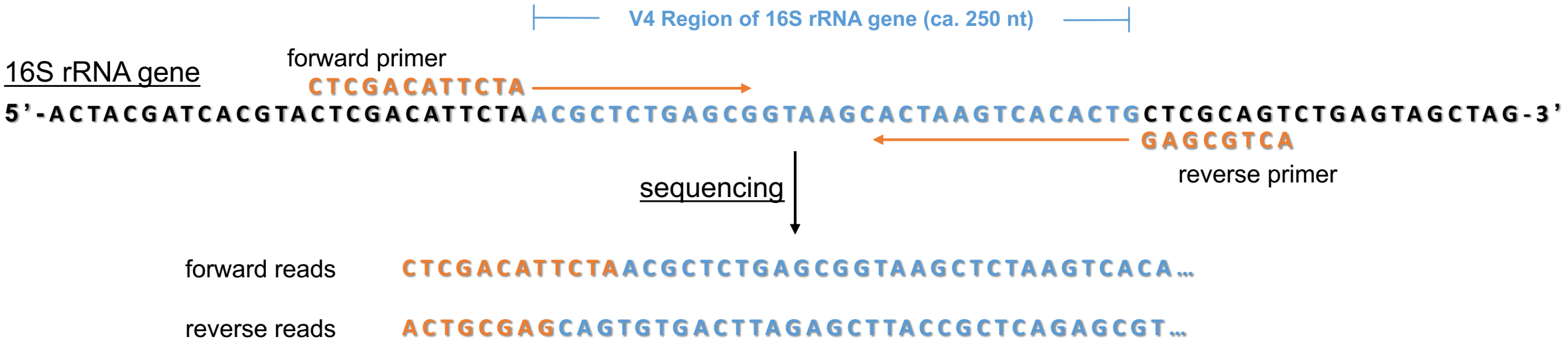


Secondary structure of 16S rRNA



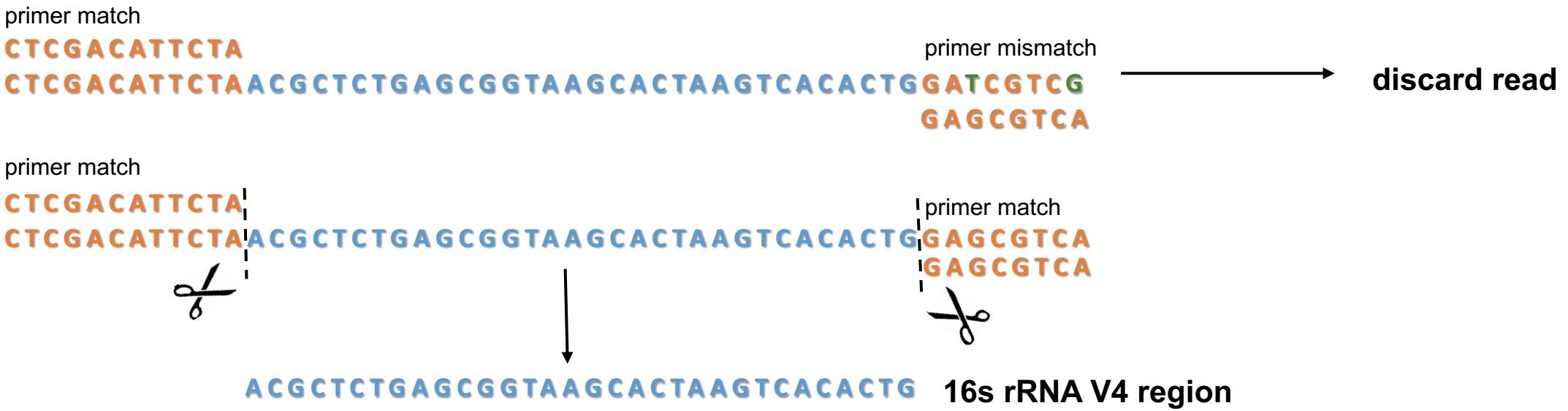
Nature Reviews | Microbiology

Generation of 16S rRNA gene PCR amplicons



- PCR products (amplicons) are used for sequencing
- The sequencing output are forward reads and reverse reads
- Sequencing reads are saved as plain (or compressed) text files in the fastq format
  - see: https://en.wikipedia.org/wiki/FASTQ_format

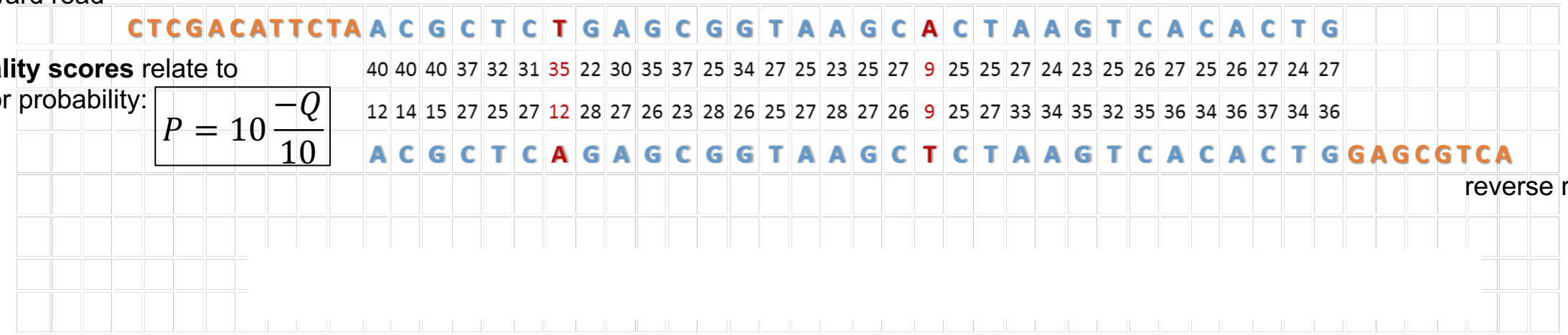
Quality control reads: primer match filtering



- Forward and reverse primer sequences are aligned to the read.
- If both primers perfectly match, the read is used for further steps, otherwise the read is discarded.
- This assures that all reads start at the same 16S position, which is mandatory for the pipeline to work.

Quality scores of amplicon reads

forward read



quality scores relate to error probability: $P = 10^{\frac{-Q}{10}}$

- Each base of the forward and reverse have a quality (Q) score:
  - the higher the Q score the higher the probability that the base call is correct

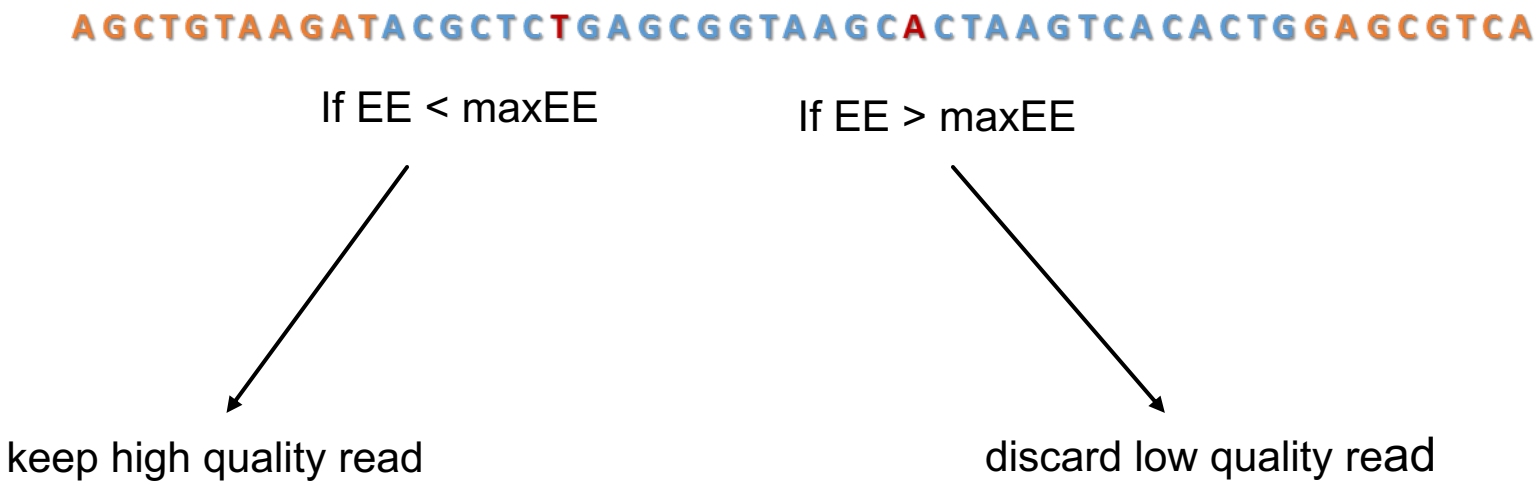
Quality control of amplicon reads: error filtering

- Quality filtering by maximum expected errors (maxEE) should be performed as a first processing step

EE = expected errors = sum of error probability (sum of P)

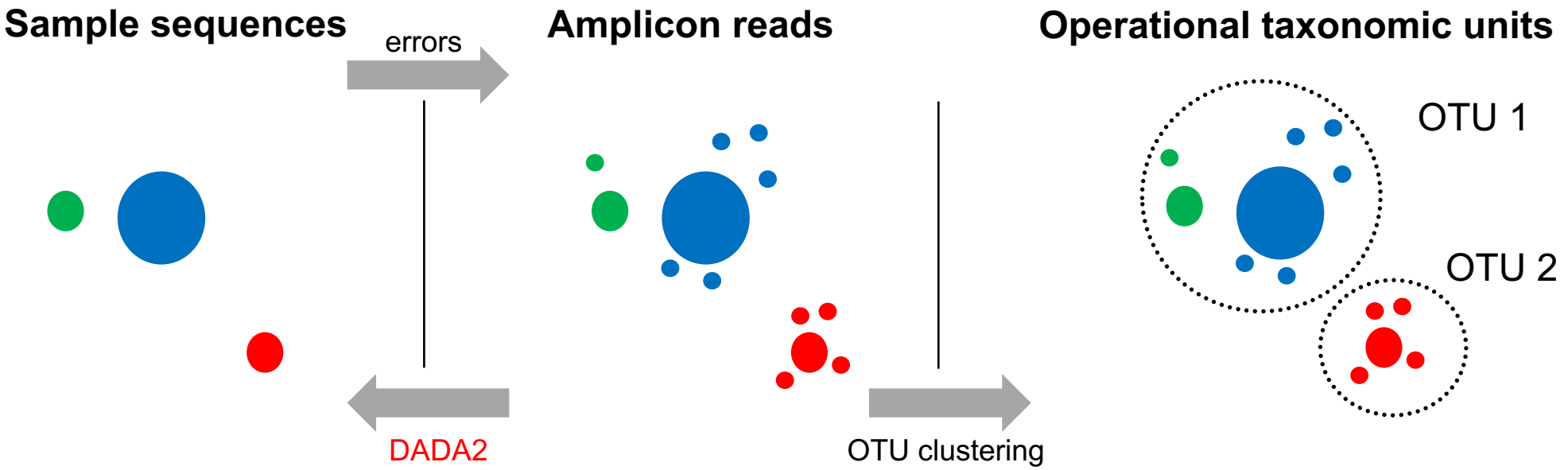
  - small EE = high quality; large EE = low quality

- By setting a maximum expected errors (maxEE) threshold, we can discard reads with EE > maxEE



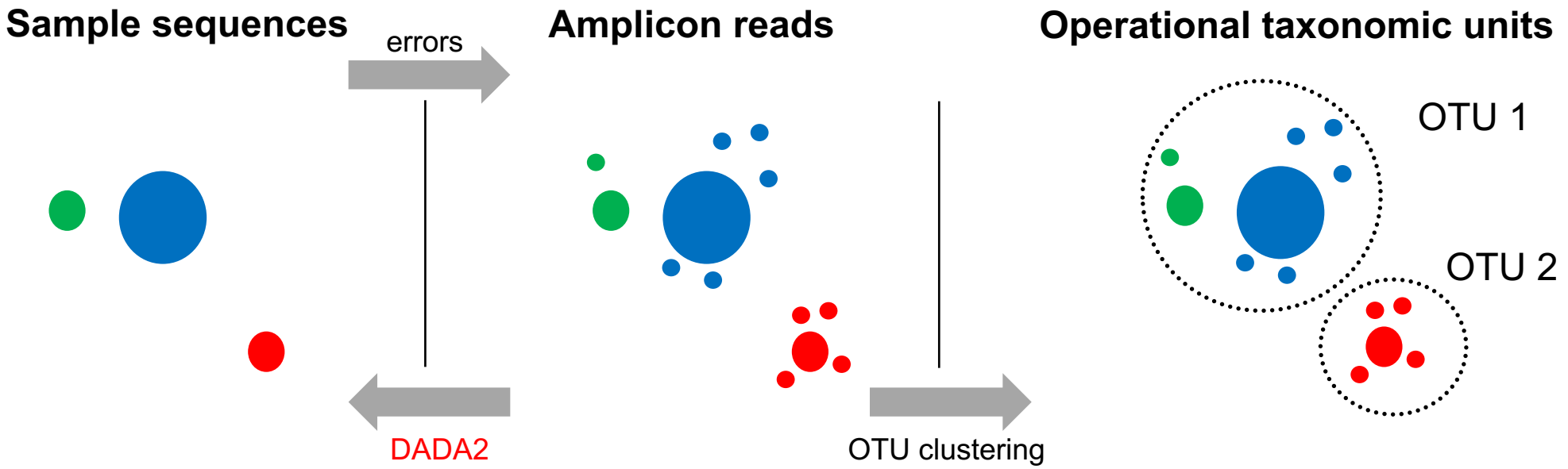
The Divisive Amplicon Denoising Algorithm (DADA)

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



The Divisive Amplicon Denoising Algorithm (DADA)

- *DADA2 tries to decipher whether each amplicon is a true biological sequence or is the result of an error because of the sequencing process*



DADA2: the error model

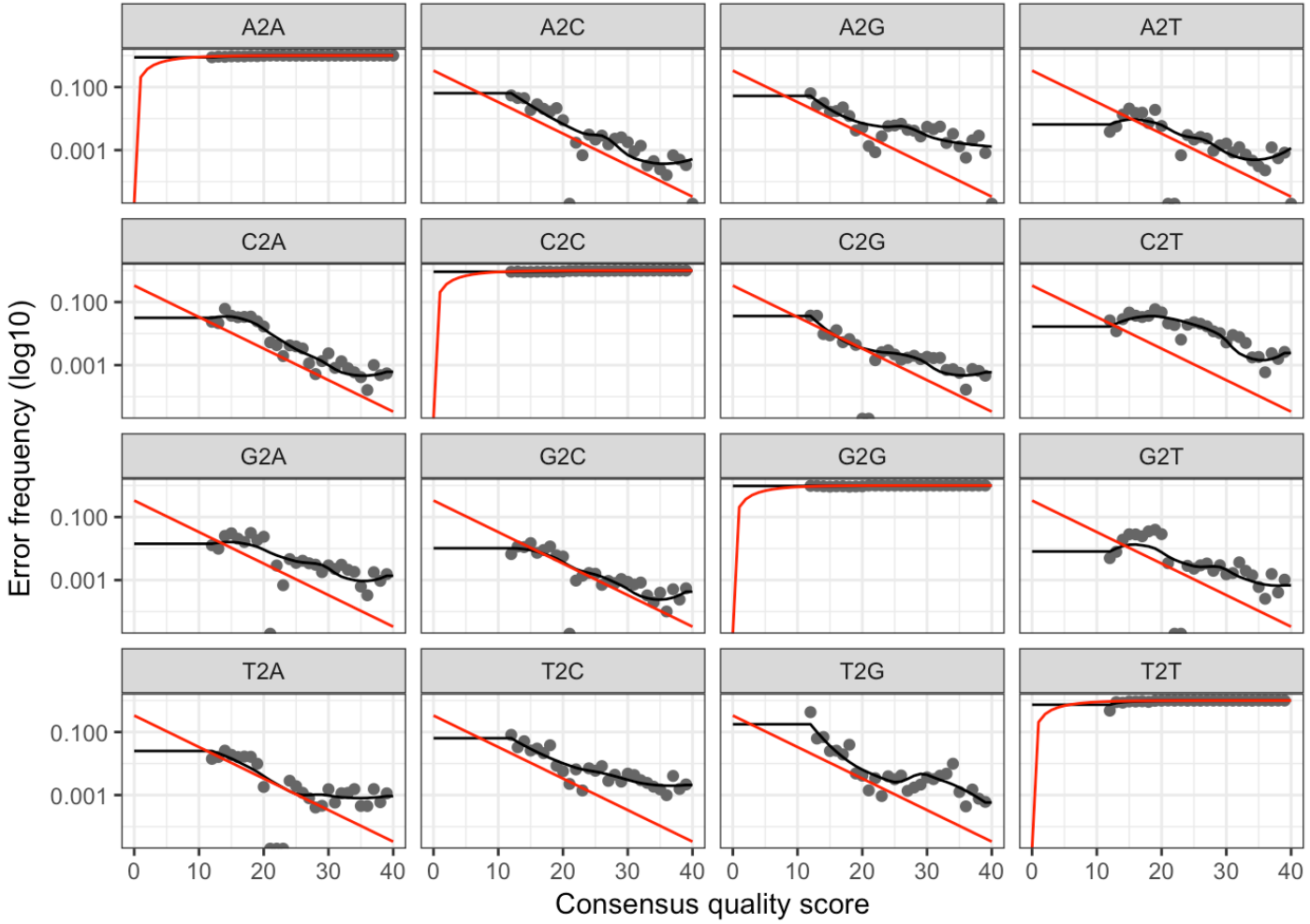
- The rate at which an amplicon read with sequence *i* is produced from sample sequence *j* is reduced to the product over the transition probabilities between the *L* aligned nucleotides.

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

- The transition probability between aligned nucleotides is allowed to depend on the original nucleotide, substituting nucleotide, and associated quality score, for example, $p(A \rightarrow T, 9)$

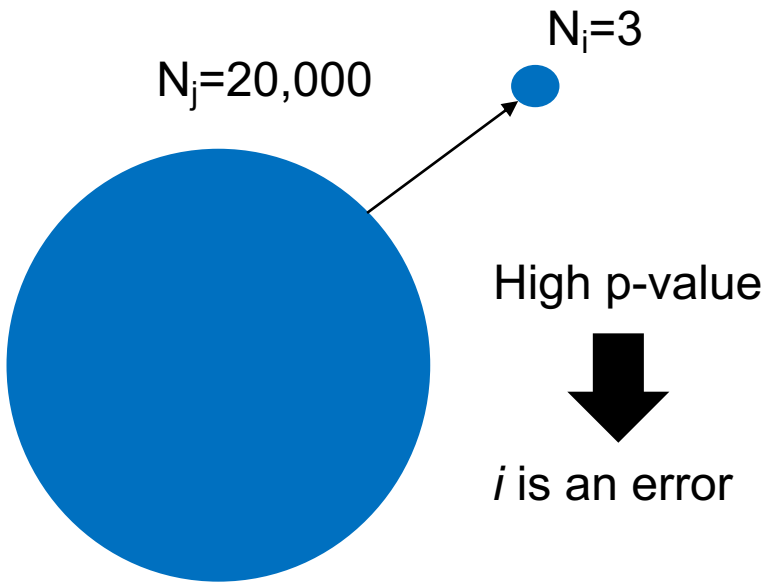
	A	C	G	C	T	C	T	G	A	G	C	G	G	T	A	A	G	C	A	C	T	A	A	G	T	C	A	C	A	C	T	G	
	40	40	40	37	32	31	35	22	30	35	37	25	34	27	25	23	25	27	9	25	25	27	24	23	25	26	27	25	26	27	24	27	
	12	14	15	27	25	27	12	28	27	26	23	28	26	25	27	28	27	26	9	25	27	33	34	35	32	35	36	34	36	37	34	36	
	A	C	G	C	T	C	A	G	A	G	C	G	G	T	A	A	G	C	T	C	T	A	A	G	T	C	A	C	A	C	T	G	

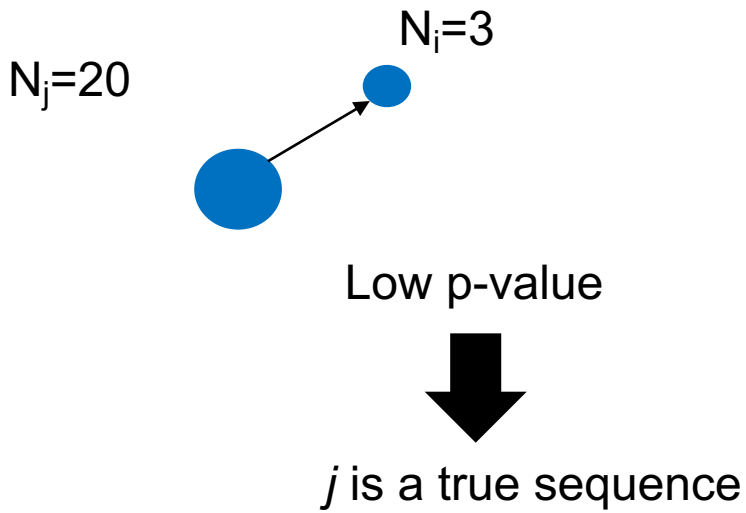
DADA2: the error model



DADA2: the abundance p-value

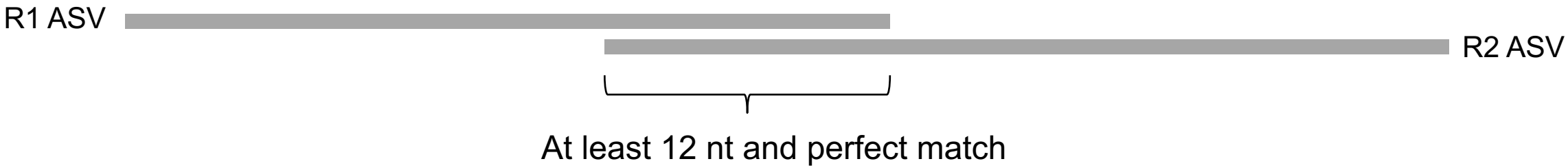
- The abundance *p*-value quantifies the notion that sequence *i* is too abundant to be explained by errors in amplicon sequencing.
- It measures the probability of a given amplicon abundance given the model. That is, the likelihood that an amplicon is produced n times because errors.





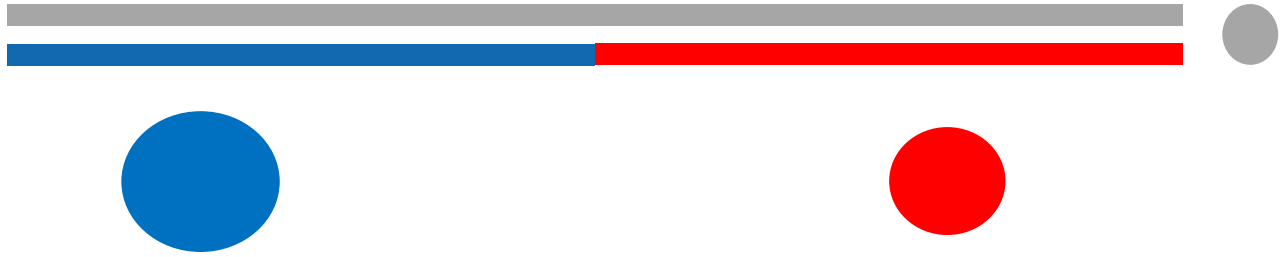
Merging and chimera removal

- Merging:

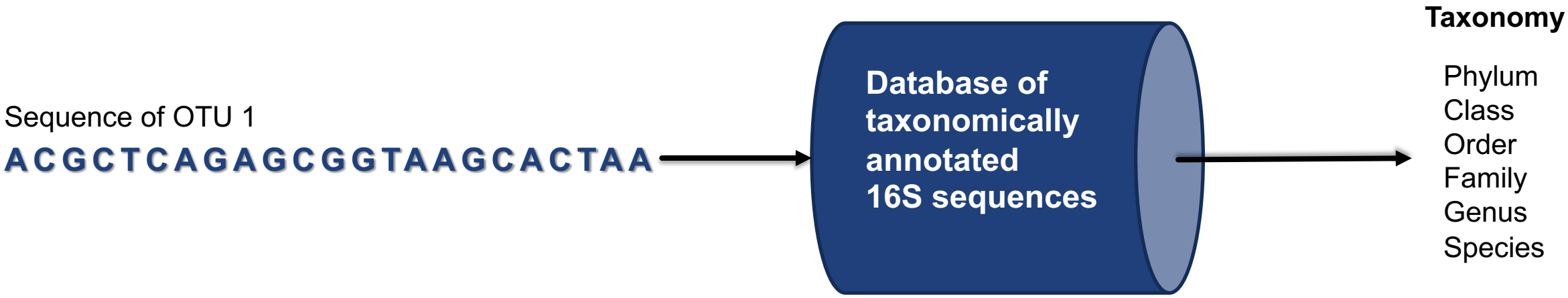


- Chimera detection and removal:

Chimeric sequences are identified if they can be exactly reconstructed by combining a left-segment and a right-segment from two more abundant "parent" sequences



Taxonomic annotation of ASVs



- Prediction of ASV taxonomy
- Each is compared to a database of annotated 16S rRNA gene sequences
- Sequences are classified to a phylum, class, family etc.