DADA2

ANSC516

Pop Quiz

Find the following file in

/depot/microbiome/data/ANSC516/class_materials

`taxonomy.qza`

DADA2: High-resolution sample inference from Illumina amplicon data

Benjamin J Callahan[1], Paul J McMurdie[2],
Michael J Rosen[3], Andrew W Han[2], Amy Jo A Johnson[2] &
Susan P Holmes[1]

We present the open-source software package DADA2 for
modeling and correcting Illumina-sequenced amplicon errors
(https://github.com/benjjneb/dada2). DADA2 infers sample
sequences exactly and resolves differences of as little as
1 nucleotide. In several mock communities, DADA2 identified
more real variants and output fewer spurious sequences than
other methods. We applied DADA2 to vaginal samples from a
cohort of pregnant women, revealing a diversity of previously
undetected *Lactobacillus crispatus* variants.

The importance of microbial communities to human and envi-

We previously introduced the Divisive Amplicon Denoising
Algorithm (DADA), a model-based approach for correcting
amplicon errors without constructing OTUs[5]. DADA identified
fine-scale variation in 454-sequenced amplicon data while out-
putting few false positives[2,5].

Here we present DADA2, an open-source R package (https://
github.com/benjjneb/dada2, **Supplementary Software**) that extends
and improves the DADA algorithm. DADA2 implements a new qual-
ity-aware model of Illumina amplicon errors. Sample composition is
inferred by dividing amplicon reads into partitions consistent with
the error model (Online Methods). DADA2 is reference free and
applicable to any genetic locus. The DADA2 R package implements
the full amplicon workflow: filtering, dereplication, sample infer-
ence, chimera identification, and merging of paired-end reads.

We compared DADA2 to four algorithms (Online Methods):
UPARSE, an OTU-construction algorithm with the best published
false-positive results[9]; MED, an algorithm with the best published
fine-scale resolution in Illumina amplicon data[11]; and the popular
mothur (average linkage) and QIIME (uclust) OTU methods[7,8].

We benchmarked these algorithms on three mock commu-
nity data sets: Balanced, HMP, and Extreme (Online Methods
and **Supplementary Table 1**), each sequenced at a depth of over

Correcting sequencing error

- How do we eliminate sequencing error?
- Each sequencing platform has different pattern of error-making.
- Roche 454 (2006-~2015)
  - long reads
  - High error rate
- Illumina (~2012-present)
  - Shorter reads
  - Lower error rate
- PacBio (2013-present) or Nanopore (2015-present)
  - Ultralong reads
  - Very high error rate

Past error correcting methods

- Amplicon length (too long or short → remove)
- Alignment to reference (if there is no match → remove)
- Homopolymers (e.g. catgcta**AAAAAAAAA**tatgatcta)
- Chimera removal
- OTUs based on 97% similarity
  - Good at removing sequencing errors
  - But eliminates potentially interesting fine-scale biological variation.
  - Reduces ability for finer-scale classification to distinguish
    - Pathogens and commensals
    - Ecological niches
    - Temporal dynamics

What does DADA2 do?

- Implements quality-aware model of sequencing errors
- Merging of paired end reads
- Chimera filtering
- Divides reads into partitions consistent with the error model
- Reference free (as opposed to closed reference clustering)
  - Could be used with other gene amplicons

https://benjjneb.github.io/dada2/tutorial.html

.fasta file

>Sequence1ID
Sequence (ACTGCATGACTGATGC)
>Sequence2ID
Sequence (GCTAGCTGATGCA)

.fastq file

@Sequence1ID
Sequence (ACTGCATGACTGATGC)
+
Quality (IIHEFG789IIHBAFF)

```
[john2185@bell-fe00:/scratch/bell/john2185/qiime/qiime2-moving-pictures-tut]
orial/emp-single-end-sequences $ head sequences.fastq
@HWI-EAS440_0386:1:23:17547:1423#0/1
TACGNAGGATCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTAGATGGATGTTTAAGTCAGTTGTGA
AAGTTTGCGGCTCAACCGTAAAATTGCAGTTGATACTGGATATCTTGAGTGCAGTTGAGGCAGGGGGGGGATTGG
TGTG
+
IIIE)EEEEEEEGFIIGIIIHIHHGIIIGIIHHHGIIHGHEGDGIFIGEHGIHHGHHGHHGGHEEGHEGGEHE
BBHBBEEDCEDDD>B?BE@@B>@@@@@CB@ABA@@?@@=>?08;3=;==8:5;@6?################
####
@HWI-EAS440_0386:1:23:14818:1533#0/1
CCCCNCAGCGGCAAAAATTAAAATTTTTACCGCTTCGGCGTTATAGCCTCACACTCAATCTTTTATCACGAAGT
CATGATTGAATCGCGAGTGGTCGGCAGATTGCGATAAACGGGCACATTAAATTTAAACTGATGATTCCACTGCA
ACAA
+
64<2$24;1)/:*B<?BBDDBBD<>BDD#################################################
#############################################################################
####
@HWI-EAS440_0386:1:23:14401:1629#0/1
TACGNAGGATCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTAGGCGGACGCTTAAGTCAGTTGTGA
AAGTTTGCGGCTCAACCGTAAAATTGCAGTTGATACTGGGTGTCTTGAGTACAGTAGAGGCAGGGGGGGGGTTG
GGGG
```

Symbol	Q-Score	Symbol	Q-Score
!	0	continued	
"	1	6	21
#	2	7	22
$	3	8	23
%	4	9	24
&	5	:	25
'	6	;	26
(	7	<	27
)	8	=	28
*	9	>	29
+	10	?	30
,	11	@	31
-	12	A	32
.	13	B	33
/	14	C	34
0	15	D	35
1	16	E	36
2	17	F	37
3	18	G	38
4	19	H	39
5	20	I	40

Quality Score	Probability of Incorrect Base Call	Inferred Base Call Accuracy
10 (Q10)	1 in 10	90%
20 (Q20)	1 in 100	99%
30 (Q30)	1 in 1000	99.9%

Online Methods

- Where the real details are

DADA2 pipeline

Filtering

Trims sequences to a specified length, removes sequences shorter than that length, and filters based on the number of ambiguous bases, a minimum quality score, and the expected errors in a read
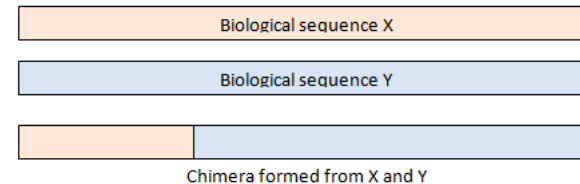
Dereplication

Dereplicated list of unique sequences and their abundances. Also outputs consensus positional quality scores

Denoising

implements the core denoising algorithm to correct errors and form ASVs

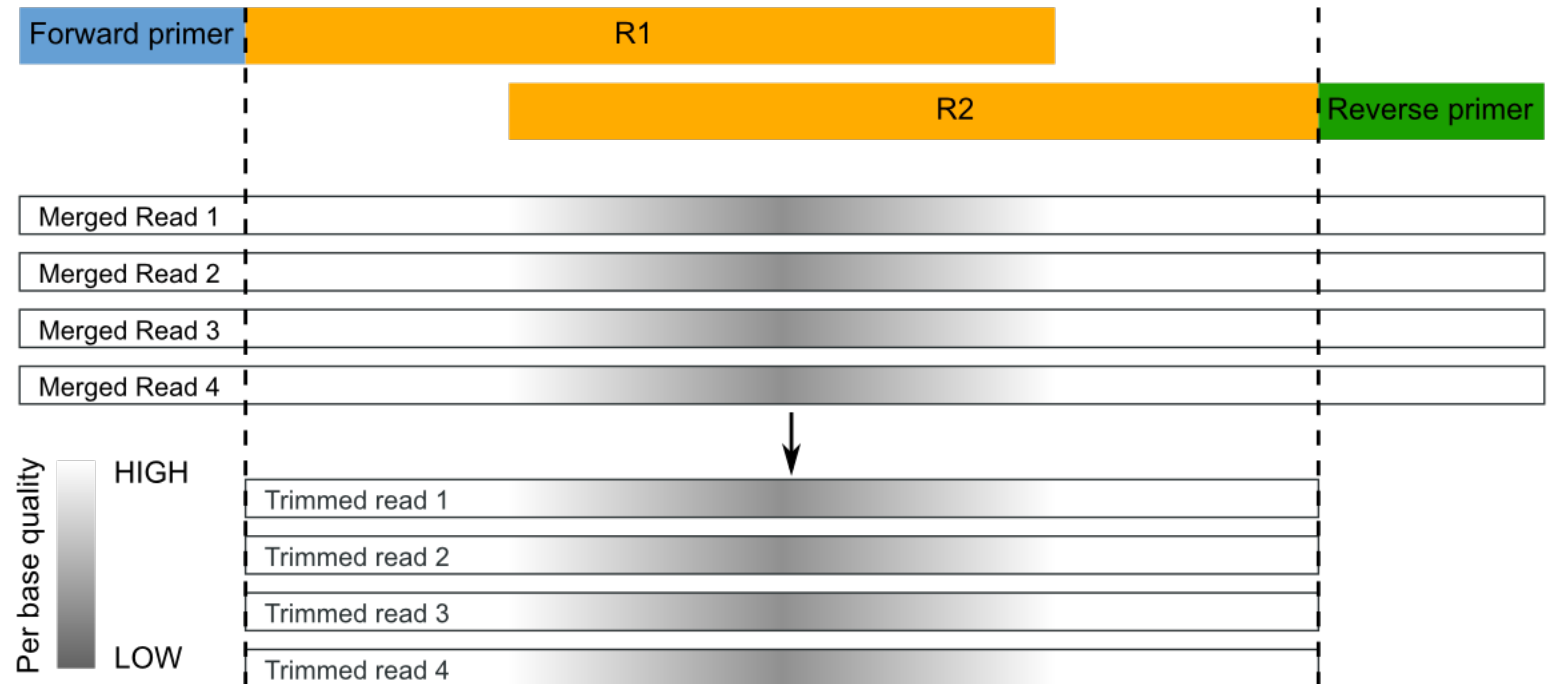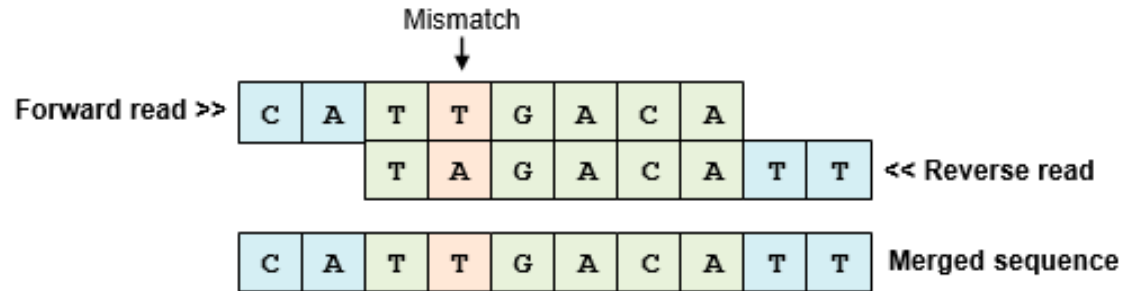identifies sequences that are exact bimeras (two-parent chimeras)

Chimeras

performs a global ends-free alignment between paired forward and reverse reads and merges them together if they exactly overlap

Merging

Biological sequence X

Biological sequence Y

Chimera formed from X and Y

Sequence merging

Core denoising algorithm
Step 1: Sequence comparison

- Pairwise sequence alignment

- Memory intensive

- Uses two heuristics to save memory.

- I'm not too worried about the details, but they warn that different parameters may be needed for other loci (genes) like fungal ITS

Step 2: Error Model

- Quality is accessed for each and every sequence

- Determine the error rate for each substitution (eg. A->T)
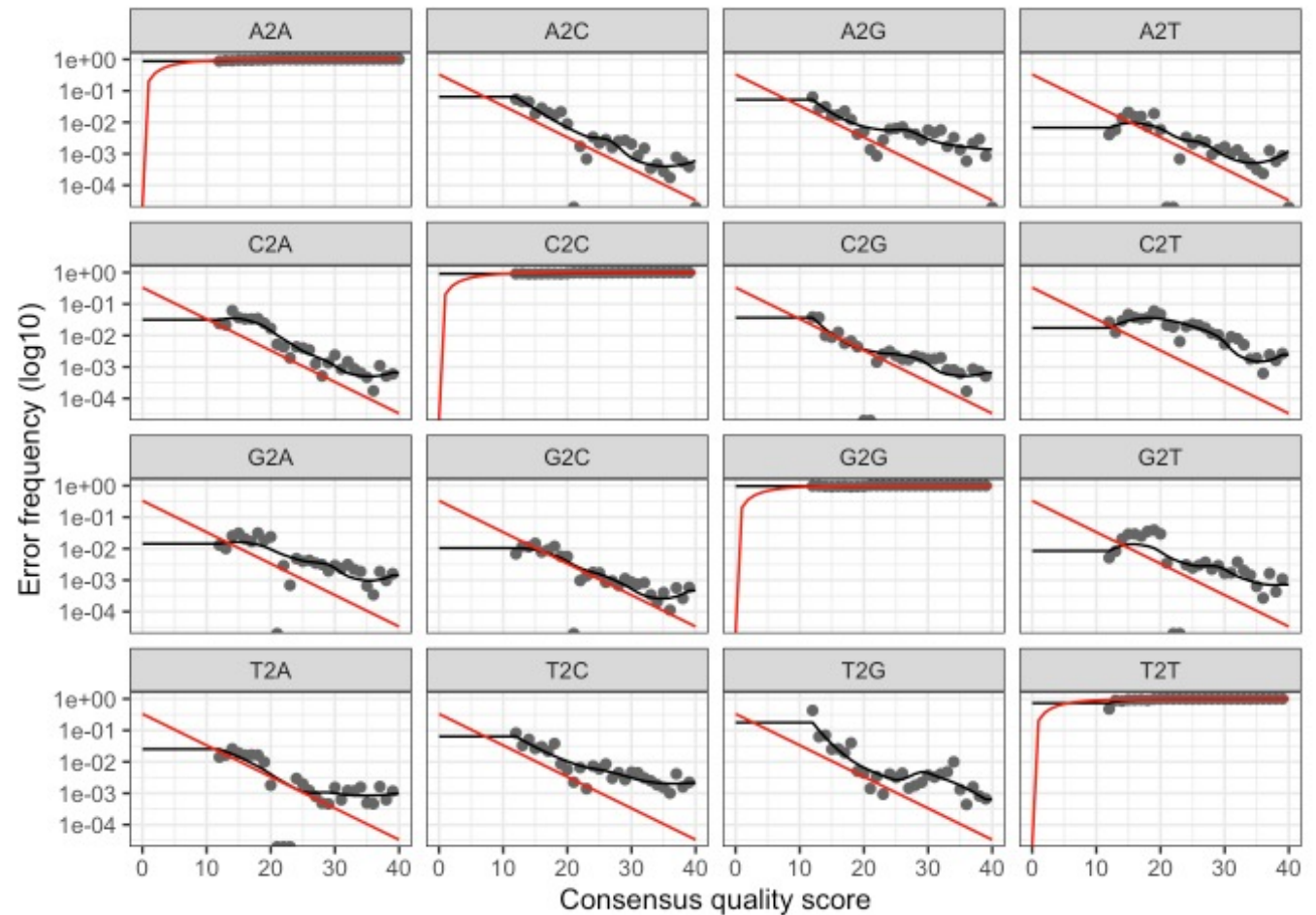
Step 3: abundance p-value

- The abundance *p*-value quantifies the notion that sequence *i* is too abundant to be explained by errors in amplicon sequencing.

- Singletons cannot form their own partitions, and DADA2 will not infer singleton sequences. The effect of this is similar in practice to the UPARSE developer's recommendation to remove singleton sequences before picking OTUs, and in both cases is driven by the difficulty in robustly differentiating singleton errors from real singleton sequences.

Step 4: Divisive partitioning algorithm

- Iterative process of assigning sequences to partitions.
- Aware of the error models created earlier

Step 5: Error model parameterization

- records the mismatches between every sequence and the center of its partition and counts each type of mismatch

- The resulting table of observed mismatches represents the errors inferred by DADA2 and can be used to estimate the parameters of the error model

DADA2 pipeline

Filtering

Trims sequences to a specified length, removes sequences shorter than that length, and filters based on the number of ambiguous bases, a minimum quality score, and the expected errors in a read
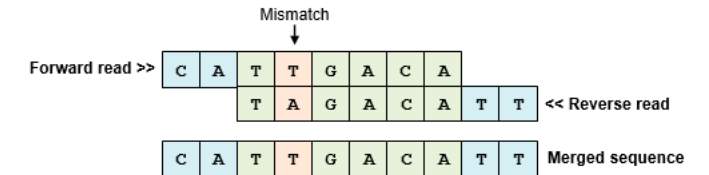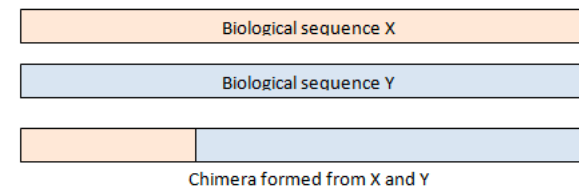
Dereplication

Dereplicated list of unique sequences and their abundances. Also outputs consensus positional quality scores

Denoising

implements the core denoising algorithm to correct errors and form ASVs

identifies sequences that are exact bimeras (two-parent chimeras)

Chimeras

performs a global ends-free alignment between paired forward and reverse reads and merges them together if they exactly overlap

Merging

Mismatch

Forward read >> C A T T G A C A

T A G A C A T T << Reverse read

C A T T G A C A T T Merged sequence

Biological sequence X

Biological sequence Y

Chimera formed from X and Y

End result

```
##          input filtered denoisedF denoisedR merged nonchim
## F3D0     7793     7113      6996      6978     6551    6539
## F3D1     5869     5299      5227      5239     5025    5014
## F3D141   5958     5463      5339      5351     4973    4850
## F3D142   3183     2914      2799      2833     2595    2521
## F3D143   3178     2941      2822      2868     2553    2519
## F3D144   4827     4312      4146      4224     3622    3483
```

Implemented in Qiime2

```
qiime dada2 denoise-single \
  --i-demultiplexed-seqs demux.qza \
  --p-trim-left 0 \
  --p-trunc-len 120 \
  --o-representative-sequences rep-seqs-dada2.qza \
  --o-table table-dada2.qza \
  --o-denoising-stats stats-dada2.qza
```

I recommend trimming the first 13 bases.
p-trunc-len depends on the quality. The 25 percentile of the Q value should stay above 20
What other parameters can you change?