VSEARCH and Swarm: Tools for clustering and other analyses of microbiome sequencing data

BIO9905MERG1 Course, UiO, 26 March 2019

Torbjørn Rognes
Dept. of Informatics, UiO & Dept. of Microbiology, OUS
torognes@ifi.uio.no

UiO : University of Oslo

Oslo University Hospital

Motivation: Problems with USEARCH

Frustration in the metagenomics community with the **USEARCH** tool (Edgar, 2010) often used for many purposes in metagenomics projects, including the QIIME pipeline.

Main problems:

- Inner workings only rudimentary described: we don't really know how it works
- Source code is not publicly available: we cannot improve on it or adapt it
- Only a 32-bit version freely available for academic use: it cannot be used with large datasets (64-bit version requires USD 1485 payment)

2

Aims

Our aim was to develop a new tool that is:

- Open source code software
- Able to handle very large databases (>4GB)
- Free of charge
- A drop-in replacement for USEARCH
- At least as accurate as USEARCH
- At least as fast as USEARCH

3

Features of VSEARCH v2.12.0

- Chimera detection (*de novo* and reference)
- Clustering (after sorting by abundance or length, or using initial order)
- Dereplication of sequences (full length and prefix) and rereplication
- FASTQ encoding detection and conversion, SFF to FASTQ conversion
- Masking of low-complexity regions with Dust
- Paired-end reads merging and joining
- Pairwise global sequence alignment (all vs all)
- Reverse complement sequences
- Searching (global alignment and exact matches)
- Sequence quality statistics and filtering
- Shuffling and sorting (abundance and length)
- Subsampling of sequences
- Taxonomic classification (SINTAX)
- >200 command-line options
- Automatic detection and reading of compressed files



4

Selected command line examples

```
vsearch --help
vsearch --allpairs_global FILENAME --id 0.5 --alnout FILENAME
vsearch --cluster_size FILENAME --id 0.97 --centroids FILENAME
vsearch --derep_fulllength FILENAME --output FILENAME
vsearch --fastq_chars FILENAME
vsearch --fastq_eestats FILENAME --output FILENAME
vsearch --fastq_mergepairs FILENAME --reverse FILENAME --fastqout FILENAME
vsearch --fastx_filter FILENAME --fastaout FILENAME --fastq_maxee 0.5
vsearch --fastx_mask FILENAME --fastaout FILENAME
vsearch --fastx_subsample FILENAME --fastaout FILENAME --sample_pct 1.0
vsearch --search_exact FILENAME --db FILENAME --alnout FILENAME
vsearch --shuffle FILENAME --output FILENAME
vsearch --sortbysize FILENAME --output FILENAME
vsearch --uchime_denovo FILENAME --nonchimeras FILENAME
vsearch --uchime_ref FILENAME --db FILENAME --nonchimeras FILENAME
vsearch --usearch_global FILENAME --db FILENAME --id 0.97 --alnout FILENAME
```

More features and options

- Almost drop-in replacement for USEARCH v 7
- Over 200 command-line options
- A wide range of different output file formats (fasta, fastq, blast-like tab-separated, user-defined tab-separated, human readable alignments, otu tables in 3 formats, uclust-like, fastapairs, chimera alignment, ...)
- Can automatically read gzip or bzip2 compressed input files

General search algorithm

For each query sequence:

- Sort target (database) sequences by decreasing number of *k*-mers shared with the query sequence
- Consider target sequences in order and align the query to each candidate target sequence
- Stop when *A* targets (default 1) have been accepted, i.e. they satisfy the accept criteria (e.g. id > 97%)
- Stop when *R* targets (default 32) have been rejected, i.e. they do not satisfy the accept criteria.

7

Heuristic search algorithm

- The sequence database is indexed
- Which database sequences contain at least one occurrence of a given k-mer?
- 4^k possible k-mers
- Quickly identify which database sequences share the most k-mers with the query
- By default k=8 (*wordlength*)
- By default at least 12 shared k-mers are required (*minwordmatches*), but never more than exist in the query
- Repeated k-mers counts only once

8

Example with *k*=2

Query: AACGTCCTGCATCGATC

```
k-mers:            AA AC CG GT TC CC CT TG GC CA AT TC CG GA AT TC
Sorted:            AA AC AT AT CA CC CG CG CT GA GC GT TC TC TC TG
Remove duplicates: AA AC AT -- CA CC CG -- CT GA GC GT TC -- -- TG
```

Target: AACGGCCTGCATAGATC

```
k-mers:            AA AC CG GG GC CC CT TG GC CA AT TA AG GA AT TC
Sorted:            AA AC AG AT AT CA CC CG CT GA GC GC GG TA TC TG
Remove duplicates: AA AC AG AT -- CA CC CG CT GA GC -- GG TA TC TG

Shared k-mers:     AA AC    AT    CA CC CG CT GA GC         TC TG
```

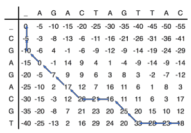
11 shared *k*-mers

9

Alignment differences

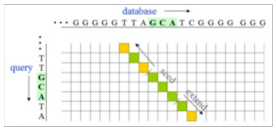
VSEARCH:

- optimal global alignment algorithm with backtracking (Needleman-Wunsch + Gotoh), 8-way SIMD parallelised



USEARCH:

- Seed-and-extend heuristic gapped alignment, as in BLAST

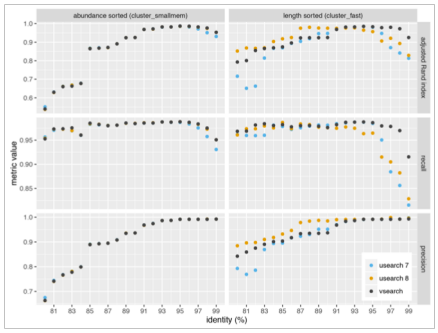


10

Clustering accuracy

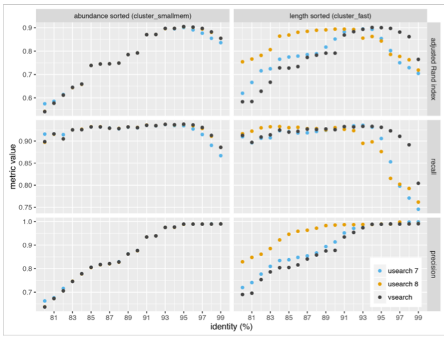
- Tested the ability of methods to cluster together organisms that are already classified as belonging to the same clade in the tree of life
- 2 datasets: even and uneven mock communities with 57 different species, 16S rRNA V4, MiSeq 250bp PE
- 100 different sequence orders (by shuffling)
- 3 methods: USEARCH v 7, USEARCH v 8, VSEARCH
- 2 sort options: length or abundance
- 20 clustering thresholds (divergence of 1-20%)
- 3 metrics: adjusted rand index, recall and precision
- Median values plotted

Clustering accuracy - uneven dataset



12

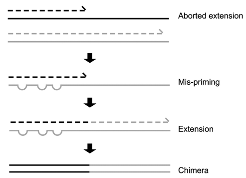
Clustering accuracy - even dataset



13

Chimeras

- Chimeric DNA sequences may form during PCR amplification
- Chimeras contain segments from 2 or more parent sequences
- Will inflate the apparent diversity of organisms in the sample unless removed
- VSEARCH implements the UCHIME algorithm (Edgar *et al.*, 2011)



14

Chimera detection example

```
Query   (  250 nt) ch31_m2_90_95/sp8:0-149/sp62:149-249/N=2/top=sp8:92.4%
ParentA (  250 nt) sp8/name=Clostridiummethylpentosum_0_1_2
ParentB (  250 nt) sp62/name=Clostridiumsporogenes_0_3_2

A     1 TGCTGCCTCCCGTAGGAGTCTGGGCCGTGTctcagtcCCAATGTGGCCGTT-CAACCTCTCAGTCCGGCTA-CTGATCGt 78
Q     1 TGCTGCCTCCCGTAGGAGTCTGGGCCGTGTTCAGTCGCCAATGTGGCCGTTCCAACCTCTCAGTCCGGCTAGCTGATCG- 79
B     1 TGCTGCCTCCCGTAGGAGTCTGGaCCGTGTctcagttCCAATGTGGCCGaT-CAcCCTCTCAGgtCGGCTAcgcatcgt- 78
Diffs                          A      NNNNNN?            A    A        AA       AAAAAAA
Votes                          +      0000000            +    +        ++        +++++
Model   AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

A    79 CGACTTGGTGAGCCATTACCTCACCAACTATcTAATCAGA-CGCGAGCCCATCTTaCAGCGATATAATCTTTGAT-AAcA 156
Q    80 CGACTTGGTGAGCCATTACCTCACCAACTAT-TAATCAGACCGCGAGCCCATCTT-CAGCGATATAATCTTTGATAAAAA 157
B    79 tGcCTTGGTaAGCCgTTACCTtACCAACTAg-ctAatgcgCCGCGgGtCCATCTc-aAagcAataAATCTTTGAT-AAAA 155
Diffs   A A      A    A      A        A AA AAAAA     A A      A A AAA AAA             B
Votes     +      +    +      +           + ++++      + +          +++ +++             +
Model   AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxxxxxxxxxxxxxBB

A   157 AAAcCATGCGATTCcgTTATgTTATGCGGTATTAgcgTTCgTTTCc--AAacGtTATtCCCctcTgtAAGGCAGGTTgCt 234
Q   158 AAATCATGCGATTCTCTTATATTATGCGGTATTAATCTTCCTTTCG--AA--GCTATCCCCACTTTGAAGGCAGGTTACC 233
B   156 AAATCATGCGATTCTCTTATATTATGCGGTATTAATCTTCCTTTCGgaAg--GCTATCCCCcacTTtgAGGCAGGTTACC 233
Diffs      B          BB    B             BBB   B    B   A   B   B   N?N BNa         B B
Votes      +          ++    +             +++   +            +   +   000 +0!         + +
Model   BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB

A   235 CACGTGTTACTCACCC- 250
Q   234 CACGTGTTACTCACCCG 250
B   234 CACGTGTTACTCACCCG 250
Diffs
Votes
Model   BBBBBBBBBBBBBBBBB

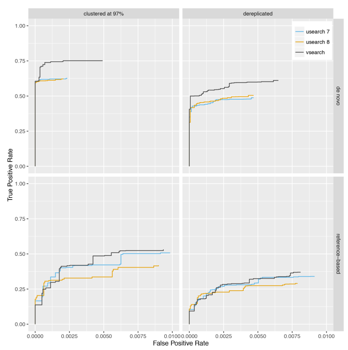
Ids.  QA 88.9%, QB 82.7%, AB 80.5%, QModel 94.7%, Div. +6.5%
Diffs Left 34: N 0, A 7, Y 27 (79.4%); Right 19: N 1, A 4, Y 14 (73.7%), Score 0.8952
```

15

Chimera detection accuracy

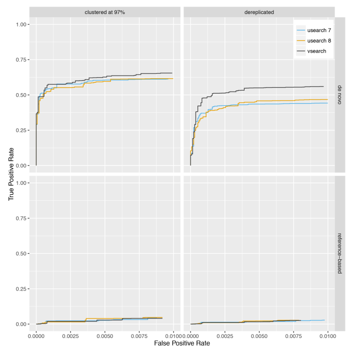
- Data from Greengenes and SILVA databases (Chimera-checked 16S rRNA db)
- PCR simulation using Simera tool
- Chimeras created by combining two different sequences
- Added Illumina noise
- First performed simple dereplication or clustering at 97%
- Performed chimera detection using USEARCH v7, USEARCH v 8 and VSEARCH
- Using either the *de novo* approach (uchime_denovo) or with a reference database of clean sequences (uchime_ref) (the RDP classifier database)
- True and false positives and negatives identified, sorted by score
- ROC-like graphs plotted

Chimera detection - Greengenes



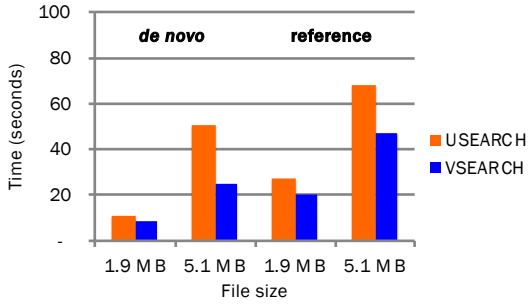
17

Chimera detection - SILVA



18

Chimera detection speed



19

Low-complexity masking with Dust

- Dust is an old algorithm for removing low-complexity regions from DNA sequences
- Tatusov & Lipman, ca 1995 (unpublished)
- Code available
- Used in BLAST and other programs
- Examples: aaaaaaa, acacacacaca, acgacgacgac
- Dust optimized and parallelised in VSEARCH
- Dust is much faster in VSEARCH than in USEARCH
- USEARCH also includes a default alternative masking algorithm called Fastnucleo that is faster but which masks numerous short regions

Examples of low-complexity filtering

Original sequence

AGCTCCAGGAACGTGTGTTAATGTTGTTGCAGTTAAAAAGCTCGTAGTTGGATGTGTGCGACGTTGCACGAATCGGCCCG
GCAGGATGCTGTGTGCTGTTCTCGCGTCGCATTCCGCGCCGCCGCGCCNNNNNNNNNNTTAGCTCGCGTGCGCGTACAC
CAAAGGCGTGCGGCGCCGTTTACTTTGAGAAAATGAGAGTGCTCCAAGCAGCCGAACTCTGTTCGGGCGTTGTATCATGC
AGCATGGGATAATGGAACAGGATCTCGCTCCGCGTTTCGTTGGTTTTTGGTGGCGGGATAATGATTAACAGAAACGGGCG
GNNNTACTGGTATTCACATGTCAGAGGTGAAATTCTTGGATTCTGTGAGGACCAACCTATGCGAAAGCGTCTGCCTAGGA
CGTTTTCA

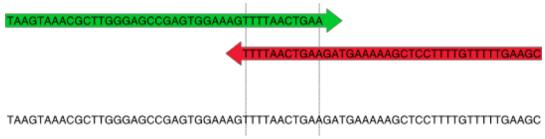
VSEARCH Dust

AGCTCCAGGAACGTGTGTTAATGTTGTTGCAGTTAAAAAGCTCGTAGTTGGATGTGTGCGACGTTGCACGAATCGGCCCG
GCAGGATGCTGTGTGCTGTTNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNTACAC
CAAAGGCGTGCGGCGCCGTTTACTTTGAGAAAATGAGAGTGCTCCAAGCAGCCGAACTCTGTTCGGGCGTTGTATCATGC
AGCATGGGATAATGGAACAGGATCTCGCTCCGCGTTTCGTTGGTTTTTGGTGGCGGGATAATGATTAACAGAAACGGGCG
GGGGTACTGGTATTCACATGTCAGAGGTGAAATTCTTGGATTCTGTGAGGACCAACCTATGCGAAAGCGTCTGCCTAGGA
CGTTTTCA

USEARCH Fastnucleo

AGCTCCAGGAACGNNNNNNTAATGTTGTTGCAGTTAANNNGCTCGTAGTTGGATNNNNNNCGACGTTGCACGAATCGGCCCG
GCAGGATGCTNNNNNCTGTTCTCGCGTCGCATTCCGCGCCGCCGCGCCNNNNNNNNNNTTAGCTCGCGTGCGCGTACAC
CAAAGGCGTGCGGCGCCGTTTACTTTGAGAAAATGAGAGTGCTCCAAGCAGCCGAACTCTGTTCGGGCGTTGTATCATGC
AGCATGGGATAATGGAACAGGATCTCGCTCCGCGTTTCGTTGGTTNNNGGTGCGGGATAATGATTAACAGAAACGGGCG
GNNNTACTGGTATTCACATGTCAGAGGTGAAATTCTTGGATTCTGTGAGGACCAACCTATGCGAAAGCGTCTGCCTAGGA
CGTTTTCA

Merging paired-end reads



Read 1 (green): Forward sequence
Read 2 (red): Reverse complementary sequence

- Find best overlap between the two sequences
- Take the quality score (error probability) of each base into account

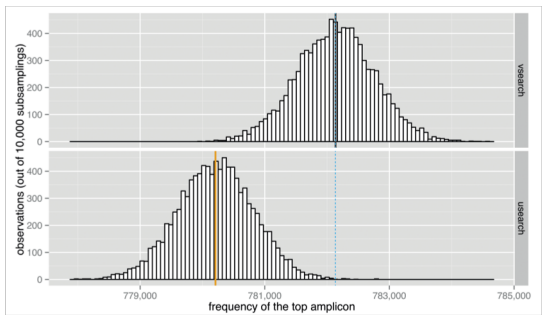
22

Paired-end reads merging

Bacterium	Program	Pairs	Merged	Correct	%Merged	%Cor/Mer	%Cor/All	Time (s)
Staphylococcus aureus	USEARCH 7	647,052	273,438	270,849	42.26	99.05	41.86	11.65
	USEARCH 8	647,052	203,729	202,003	31.49	99.15	31.22	4.69
	VSEARCH	647,052	214,988	213,103	33.23	99.12	32.93	2.15
Methylococcus capsulatus strain Bath	USEARCH 7	673,845	643,903	642,720	95.56	99.82	95.38	14.43
	USEARCH 8	673,845	554,099	553,747	82.23	99.94	82.18	6.27
	VSEARCH	673,845	581,752	581,346	86.33	99.93	86.27	3.61

23

Subsampling bias



24

Some disadvantages of VSEARCH

Compared to USEARCH:

- Slower on long sequences due to full optimal alignment
- no support for amino acid sequences
- no UPARSE pipeline support
- many newer USEARCH commands not supported

Documentation

- Documentation / user manual in man file and PDF (>40-pages)
- Publication:
  Rognes T, Flouri T, Nichols B, Quince C, Mahé F. (2016)
  VSEARCH: a versatile open source tool for metagenomics
  PeerJ 4:e2584 doi: 10.7717/peerj.2584
- Already cited more than 700 times
- Reproducible results. All code and data for generating all results are available: https://github.com/torognes/vsearch-eval
- >1500 unit tests
- Wiki, issue tracker, online support forum etc

- Suggest new features!
- Report bugs!
- Ask for help!



Availability

- VSEARCH source code is available on GitHub: https://github.com/torognes/vsearch
- Dual open-source license:
  - GNU Affero General Public License version 3
  - BSD 2-clause license
- Binary distribution (64 bit) available for
  - Linux x86, ARMv8 and ppc64le
  - macOS 10.9 and later
  - Windows 7 and later
- May be used in mothur for clustering and chimera detection
- QIIME 2 plugin; Conda, Homebrew, Debian packages; Galaxy wrapper



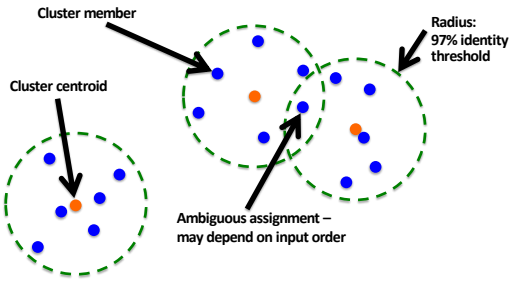
28

Swarm: a new clustering method

- New method for clustering amplicon sequences into Operational Taxonomic Units (OTUs)
- OTUs are groups of sequences that could approximately represent e.g. a species or a genus as we define it
- Single linkage hierarchical clustering approach
- Avoids two problems with traditional methods:
  - input order dependence
  - global clustering threshold
- High resolution
- Linear time and space complexity: fast and limited memory demands. Allows huge datasets to be clustered in reasonable time

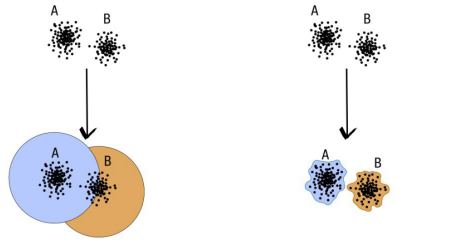
29

Heuristic centroid-based clustering



30

Clustering thresholds



compromise threshold
unadapted threshold

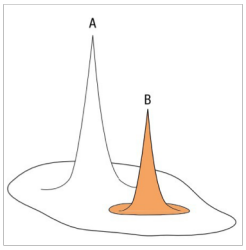
defined by maximum
distance from centroid

natural limits of OTUs

defined by minimum
distance separating clusters

31

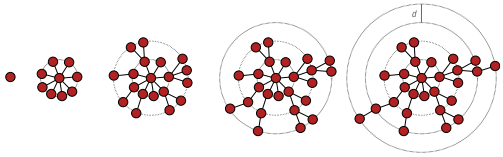
OTUs - mountains in sequence space



The height of the peaks is given by the abundance of each amplicon, while the position is given by the sequence.

32

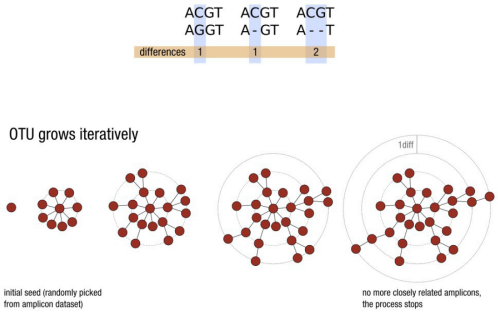
Linking amplicons



- Amplicons (nodes) are linked when distance is less than or equal to *d*.
- Similar to single linkage hierarchical clustering
- Forms a spanning tree
- The adjustable parameter *d* is 1 by default.

33

Linking amplicons



34

Algorithm for linking when d>1

- Find all sequences within distance *d* from each other.
- Need to compute distance between all pairs of sequences using global alignment
- Time complexity: $O(N^2)$, where N is the number of sequences, assuming constant sequence length.

35

Avoiding alignments

We can eliminate most alignments using:

- *k*-mer based filtering of sequences that must be at a distance of more than *d* because they have more than 2*kd* differences in their *k*-mer vectors. Swarm uses k=5.
- triangle inequality to eliminate computation of dist(a,c) if we already know dist(a,b) and dist(b,c).

36

Algorithm for linking when d=1

- Generates all "microvariants" of each sequence
- Microvariants are at distance 1 from the sequence and contain a single mutation, either a substitution, a deletion or an insertion resulting in a different sequence.
- There exists between 6L + 5 and 7L + 4 microvariants of a nucleotide sequence of length L, depending on the contents of the sequence
- Very fast and easy to find sequences that differ by a single base difference

37

Microvariants example

Original sequence (L=2): AT

Microvariants (18 different):

CT, GT, TT	(substitution at position 1)
AA, AC, AG	(substitution at position 2)
AAT, CAT, GAT, TAT	(insertion before position 1)
~~AAT~~, ACT, AGT, ATT	(insertion between 1 and 2)
ATA, ATC, ATG, ~~ATT~~	(insertion after position 2)
T	(deletion at position 1)
A	(deletion at position 2)

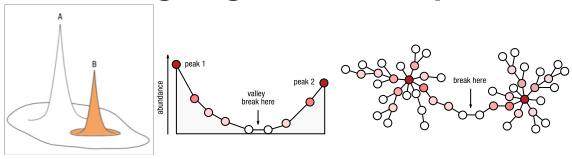
38

Hashing microvariants

- A hash is computed for each sequence in the set and they are all stored in a hash table.
- All microvariants of each sequence is generated, their hash is computed and a lookup in the hash table is performed to see of the microvariant is equal to an existing sequence.
- Time complexity: O(*N*), where *N* is the number of sequences, assuming constant sequence length.

39

Breaking long chains of amplicons

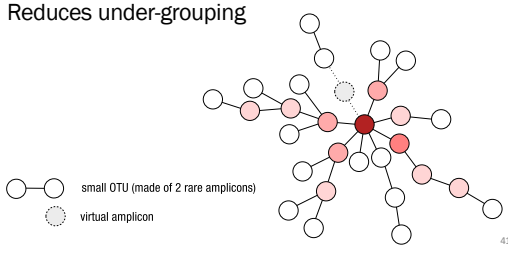


- OTUs can be considered mountains in sequence space with peaks corresponding to the most abundant sequences
- Chains of linked amplicons can be broken to avoid problems with over-grouping
- Breakpoint in valleys between abundance peaks.
- Start at highest abundance amplicons and only link amplicons with lower or equal abundance

40

"Fastidious" mode

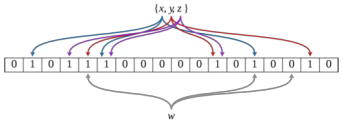
- Heuristic approximation to clustering with d=2
- Still linear complexity (but 3x slower than d=1)
- Connects small OTUs to large OTUs if d=2.
- Reduces under-grouping



41

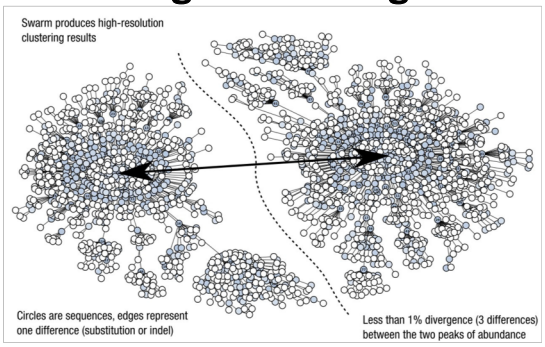
"Fastidious" algorithm

- Stores all non-existing microvariants ("virtual amplicons") outside real amplicons of large OTUs in a Bloom filter, a very space-efficient probablistic data structure.
- Detects "virtual amplicons" by checking the rim of the small OTUs against the Bloom filter.

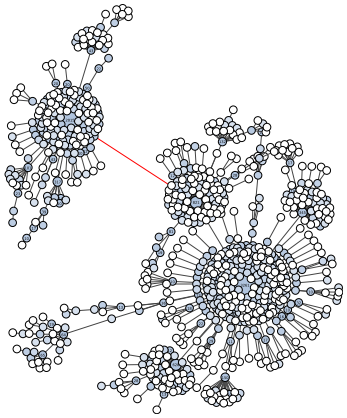


42

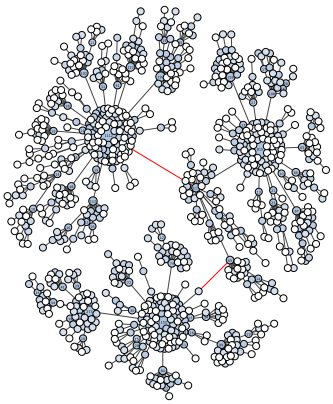
Visualising the clustering results



43



44



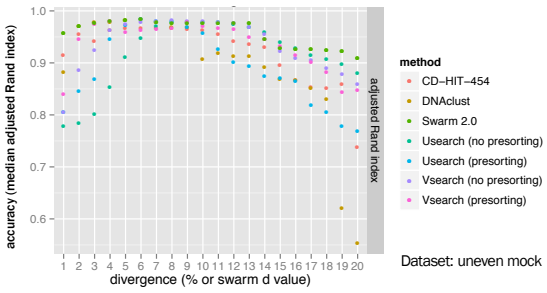
45

Clustering accuracy

- Tested the ability of methods to cluster together organisms that are already classified as belonging to the same clade in the tree of life
- 2 datasets: even and uneven mock communities with 57 different species, 16S rRNA V4, MiSeq 250bp PE
- 100 different sequence orders (by shuffling)
- 5 methods: CD-HIT, DNACLUST, SWARM, USEARCH (both with and without presorting by length)
- 20 clustering thresholds (divergence of 1-20%)
- 3 metrics: adjusted rand index, recall and precision
- Median values plotted

Clustering accuracy (uneven)

Ability of methods to correctly cluster together sequences of the same clade

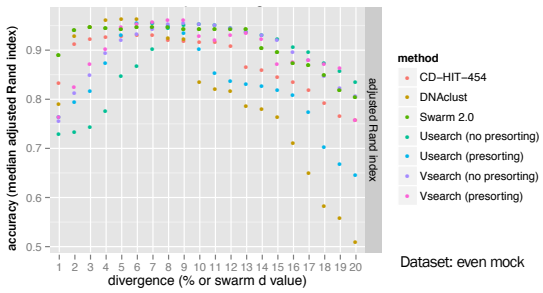


Dataset: uneven mock

47

Clustering accuracy (even)

Ability of methods to correctly cluster together sequences of the same clade



Dataset: even mock

48

SWARM 2 Performance

Dataset	Fastidious	Elapsed time (hh:mm:ss)	Max resident memory (GB)	OTUs	OTU reduction
small	N	01:45:07	41	49,138,023	
small	Y	04:59:05	114	29,485,078	40,0%
large	N	03:41:16	86	107,620,192	
large	Y	11:28:26	239	65,520,237	39,1%

- Fastidious mode seems to increase time and memory by a factor of about 3 while reducing the number of OTUs by about 40%, compared to a regular run with d=1.
- Clustering an even smaller dataset with UCLUST was estimated to take about 5 months (Rideout et al 2014). The same is probably true for swarm with d>1.

Small EMP dataset: 154,896,650 identical amplicons from 1,277,640,415 reads
Large EMP dataset: 314,871,149 identical amplicons from 2,254,207,945 reads
Using 8 cores and threads on an Abel cluster node.

49

Swarm history

- Version 1 (2014):
  - Computes full global alignments between candidate sequences
  - Uses 5-mers and triangle inequality to exclude sequence pairs that certainly cannot be close enough
- Version 2 (2015):
  - Generates all 'microvariants' of each sequence to identify sequences with distance 1 using hashing
  - Microvariants are all single base substitutions, deletions and insertions
  - Linear speed obtained for d=1
- Version 3 (2019):
  - Stores sequences with 2 bits instead of a byte for each base
  - Uses incremental Zobrist hashing instead of CityHash
  - Up to 10x speed improvement compared to version 2 (d=1)
  - Reduced memory usage

50

Swarm version 3

- Currently in development
- Reduced memory footprint due to compression of the DNA sequences by storing 4 nucleotides in each byte instead of just 1
- Greatly increased speed due to faster detection of identical microvariants

51

SWARM v3 Performance for d=1

Dataset	Version	Elapsed time (hh:mm:ss)	Max resident memory (GB)	Speedup
EMP small	Swarm 2	01:45:07	41	
EMP small	Swarm 3	00:16:24	32	6.4
EMP large	Swarm 2	03:41:16	86	
EMP large	Swarm 3	01:07:27	66	3.3

- Clustering an even smaller dataset with UCLUST was estimated to take about 5 months (Rideout et al 2014). The same is probably true for swarm with d>1.

Small EMP dataset: 154,896,650 identical amplicons from 1,277,640,415 reads
Large EMP dataset: 314,871,149 identical amplicons from 2,254,207,945 reads
Using 8 cores and threads on an Abel cluster node.

52

Availability

- Available on GitHub: https://github.com/torognes/swarm
- Free of charge, Open source license (GPL)
- Extensive documentation
- Precompiled 64-bit binaries available for Linux and macOS
- Porting to other CPUs and OSs in development
- Available for *de novo* OTU picking in QIIME 1.9

53

Publications

Swarm 1:

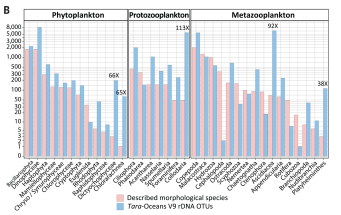
- Mahé F, Rognes T, Quince C, de Vargas C, Dunthorn M. (2014) **Swarm: robust and fast clustering method for amplicon-based studies**. PeerJ 2:e593. doi: 10.7717/peerj.593

Swarm 2:

- Mahé F, Rognes T, Quince C, de Vargas C, Dunthorn M. (2015) **Swarm v2: highly-scalable and high-resolution amplicon clustering**. PeerJ 3:e1420. doi: 10.7717/peerj.1420

54

Swarm used in Tara Oceans study





Unknown and known components of eukaryotic plankton biodiversity

Ribosomal DNA diversity associated with the morphologically known and cataloged part of eukaryotic plankton. The total number of morphologically described species in the literature [red bars, based on (25–27)] and the corresponding total number of Tara Oceans V9 rDNA OTUs (blue bars) are indicated for each of the 35 classical lineages of eukaryotic phyto-, protozoo-, and , etazooplankton.

de Vargas et al. (2015), Science, 348 (6237), 1261605

Main collaborators



Frédéric Mahé
CIRAD, Montpellier, France



Lucas Czech
Heidelberg Inst. for Theoretical Studies, Germany

56

