

# VSEARCH and Swarm: Tools for analysis of microbiome sequencing data

BI09905MERG1 Course, UiO, 9 April 2025

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

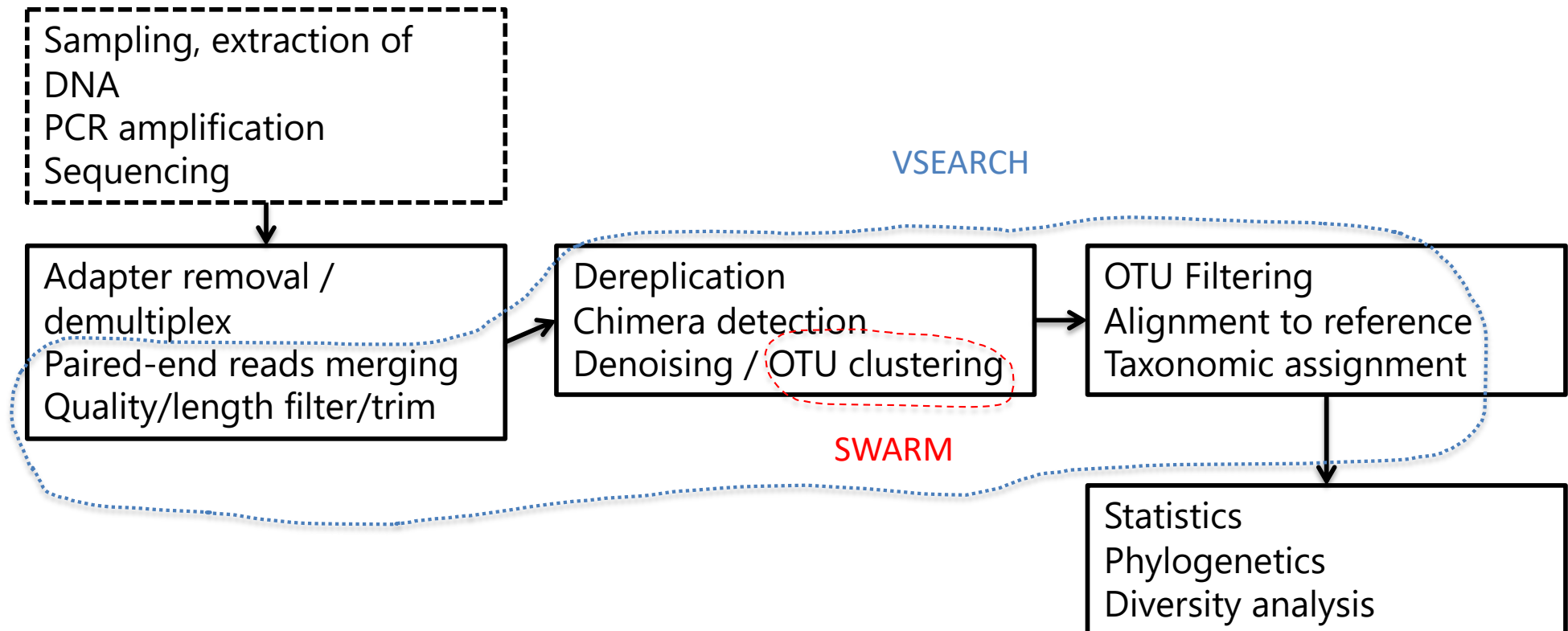


UiO : **University of Oslo**



**Oslo  
University Hospital**

# Amplicon sequence analysis workflow



# VSEARCH: a versatile analysis tool

- Versatile command-line tool to analyse DNA sequence data in microbiome projects
- Well suited for amplicon data
- Free of charge
- Open-source software
- Robust, well-tested
- Available on many platforms and architectures
- 64 bit, able to handle very large databases (>4GB)
- A drop-in replacement for USEARCH in many cases

# Features of VSEARCH v2.30.0

VSEARCH includes 48 commands with 197 options:

- Chimera detection (reference or *de novo* (uchime, uchime2 or uchime3))
- Clustering (optionally after sorting by abundance or length, or unoise-like or using little memory)
- Detection and decompression of compressed input files (.gz, .bz2)
- Dereplication of sequences (full length, prefix & id) and rereplication
- Extraction of sequences and sub-sequences from large FASTA files
- FASTQ encoding detection and conversion, SFF to FASTQ conversion, FASTQ to/from FASTA
- Masking of low-complexity regions
- Orienting of sequences in same direction as database sequences
- Paired-end reads merging and joining
- Pairwise global sequence alignment (all vs all)
- Restriction site cutting
- Reverse complementation of sequences
- Searching (global alignment and exact matches)
- Sequence quality statistics and filtering
- Shuffling and sorting (abundance and length)
- Subsampling of sequences
- Taxonomic classification (SINTAX, LCA)
- UDB files: Building, extraction and statistics of fast-loading pre-indexed database files



# Getting help with VSEARCH

Commands:

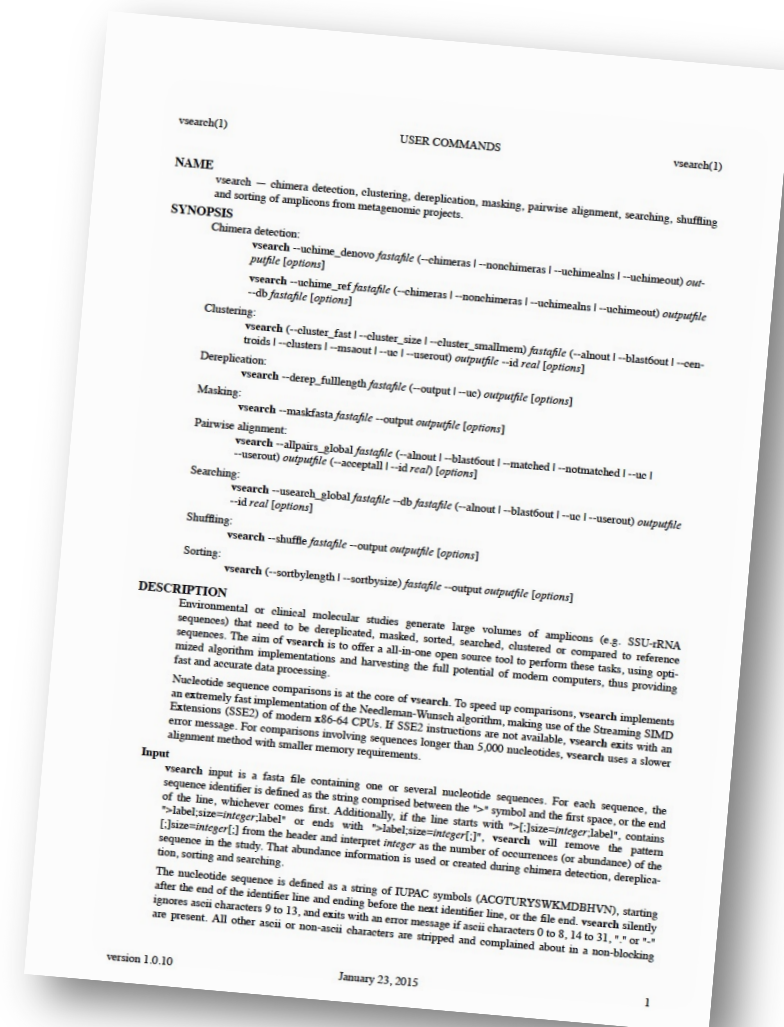
`vsearch`

`vsearch --help | less`

`man vsearch`

Manual also available as PDF:

`vsearch_manual.pdf`

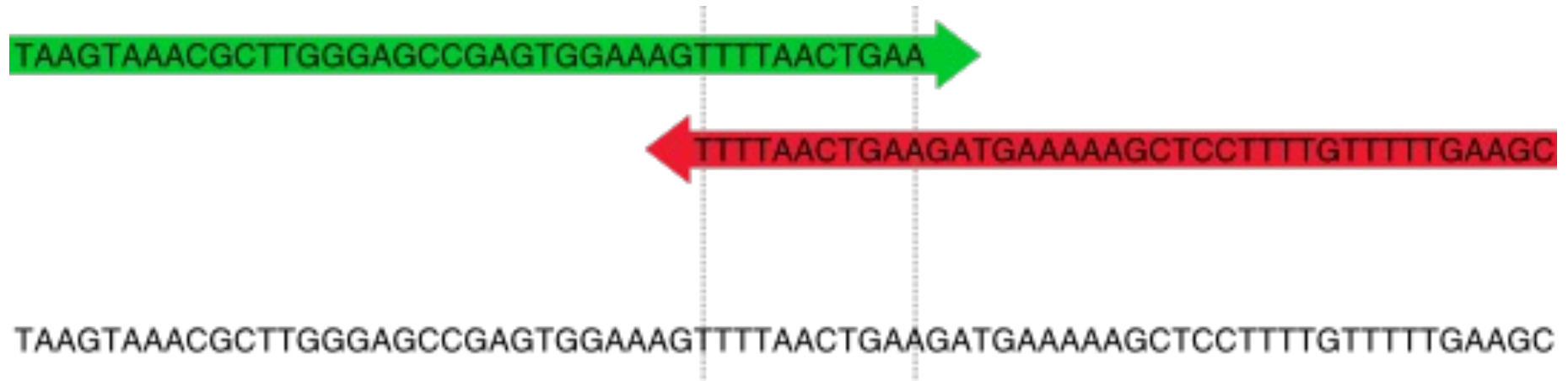


# Merging paired-end reads

```
vsearch --fastq_mergepairs forward.fastq      \  
  --reverse reverse.fastq                    \  
  --fastqout merged.fastq                    \  
  --fastq_allowmergestagger                  \  
  --fastqout_notmerged_fwd notmerged.fwd.fastq \  
  --fastqout_notmerged_rev notmerged.rev.fastq
```

Merges overlapping forward and reverse reads into one sequence, if possible. Unmerged reads are written to separate files. Allows for staggered overlapping reads (in case of very short fragments).

# 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

# Filtering reads

```
vsearch --fastq_filter input.fastq      \  
  --fastq_maxee 1.0                      \  
  --fastq_maxns 0                        \  
  --fastq_minlen 100                     \  
  --fastqout filtered.fastq              \  
  --fastaout filtered.fasta              \  
  --relabel abc                          \  

```

Removes or truncates reads that does not satisfy given requirements.  
Could be number of N's, length, quality/error, etc.



# Dereplication

```
vsearch --fastx_uniques input.fasta \
      --fastaout derep.fasta \
      --minuniquesize 2 \
      --sizein --sizeout
```

Groups strictly identical sequences into one entry in the FASTA file. Adds a "size" attribute to the FASTA header with the abundance (number of identical copies) of the sequence. In the example, only those with abundance at least 2 are included in the output.

Alternatives: derep\_fulllength, derep\_prefix, derep\_smallmem

```
>abc;size=123
acgtagtcagactgtcagactgtg
```

# Sequence similarity clustering

```
vsearch --cluster_size input.fasta \
  --id 0.97 \
  --centroids centroids.fasta \
  --sizein \
  --sizeout
```

Groups similar sequences into clusters (OTUs) using a fast, heuristic and greedy centroid-based algorithm. May specify identity threshold (e.g. 97%).

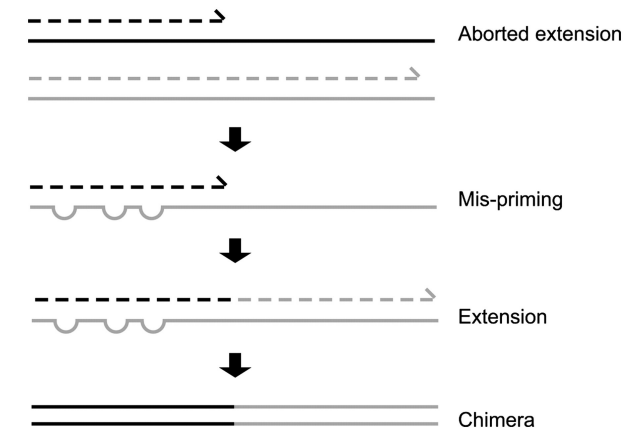
# Chimera detection

```
vsearch --uchime_denovo input.fasta \
      --nonchimeras nonchimeras.fasta \
      --chimeras chimeras.fasta \
      --sizein --sizeout
```

Detects potential chimeric sequences in the input file and writes the classified sequences into separate files.

# Chimeras

- Chimeric DNA sequences are unwanted artifacts that may form during PCR amplification
- Chimeras contain segments from 2 or more parental sequences
- Problem in amplicon-based microbiome analysis: Inflates the apparent diversity of organisms in the sample unless removed.
- Chimeras are often numerous, but with low abundance.
- Use either the dataset itself (uchime\_denovo) or a reference database (uchime\_ref) during analysis
- VSEARCH implements the UCHIME algorithm (Edgar *et al.*, 2011)
- A new algorithm for long high-quality sequences with potentially more than two parents is in development in collaboration with Kauserud, Krabberød, Stokke, Skrede & Thoen at IBV, UiO



# Chimera detection

Query ( 250 nt) ch31\_m2\_90\_95/sp8:0-149/sp62:149-249/N=2/top=sp8:92.4%  
 ParentA ( 250 nt) sp8/name=Clostridiummethylopentosum\_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-CaCCTCTCAGgtCGGCTAcgcatcgt- 78
Diffs      A      NNNNNN?      A      A      AA      AAAAAA
Votes      +      000000      +      +      ++      +++++
Model      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
  
```

```

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      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxxxxxxxxxxxxxBBBB
  
```

```

A      157 AAAcCATGCGATTTCgTTATgTTATGCGGTATTAgcgTTCgTTTCc--AAacGtTATtCCCctcTgtAAGGCAGGTTgCt 234
Q      158 AAATCATGCGATTCTCTTATATTATGCGGTATTAAATCTTCCTTTTCG--AA--GCTATCCCCACTTTGAAGGCAGGTTACC 233
B      156 AAATCATGCGATTCTCTTATATTATGCGGTATTAAATCTTCCTTTTCGgaAg--GCTATCCCCcacTTtgAGGCAGGTTACC 233
Diffs      B      BB      B      BBB      B      B      A      B      B      N?N BNa      B B
Votes      +      ++      +      +++      +      +      +      +      +      000 +0!      + +
Model      BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
  
```

```

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

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

# Sequence similarity search

```
vsearch --usearch_global query.fasta      \  
  --db database.fasta                    \  
  --id 0.99                             \  
  --maxaccepts 3                         \  
  --maxrejects 1000                     \  
  --alnout alignments.txt                \  
  --otutabout otutable.txt
```

Generic heuristic sequence similarity search using global alignment with many adjustable parameters.

# Taxonomic classification with SINTAX

```
vsearch --sintax clusters.fasta          \  
      --db silva.fasta                  \  
      --sintax_cutoff 0.8                \  
      --tabbedout classified.tsv
```

Rapid classification of sequences using a taxonomy given in a specially formatted database. Uses the SINTAX algorithm. Results may vary slightly due to randomness in the algorithm.

```
>AY232296.1.1383;tax=k:Archaea,p:Euryarchaeota,c:Halobacteria,o:Halobacteriales,f:Halobacteriaceae,g:Natrinema,s:Natrinema_sp._HM06;
```

# Taxonomic classification with LCA

```
vsearch --usearch_global clusters.fasta      \  
  --db silva.fasta                          \  
  --id 0.99                                \  
  --maxaccepts 20                           \  
  --maxrejects 1000                         \  
  --lca_cutoff 0.95                         \  
  --lcaout classified.tsv
```

Simple and rapid classification of sequences using a taxonomy given in a specially formatted database. Retrieves top 20 sequences that are at least 99% identical and finds their lowest common ancestor (LCA), allowing for 5% (i.e. 1 of 20) divergent sequences.



# General heuristic search algorithm

Approach used during search, chimera detection, and clustering

For each query sequence:

- Sort target (database) sequences by decreasing number of 8-mers shared with the query sequence
- Consider target sequences in order and align the query to each candidate target sequence
- Stop when  $A$  targets (*maxaccepts*, default 1) have been accepted, i.e. they satisfy the accept criteria (e.g. id > 97%)
- Stop when  $R$  targets (*maxrejects*, default 32) have been rejected, i.e. they do not satisfy the accept criteria.

# Converting old sequence files

Conversion from SFF to FASTQ:

```
vsearch --sff_convert input.sff \
      --fastqout output.fastq \
      --sff_clip \
      --fastq_asciiout 33
```

Conversion from old phred 64 FASTQ files to Phred 33 FASTQ:

```
vsearch --fastq_convert old.fastq \
      --fastqout new.fastq \
      --fastq_ascii 64 \
      --fastq_asciiout 33
```

Phred 64 base quality encoding was used in Solexa and early Illumina (before version 1.8) files. Today, phred 33 is almost always used.

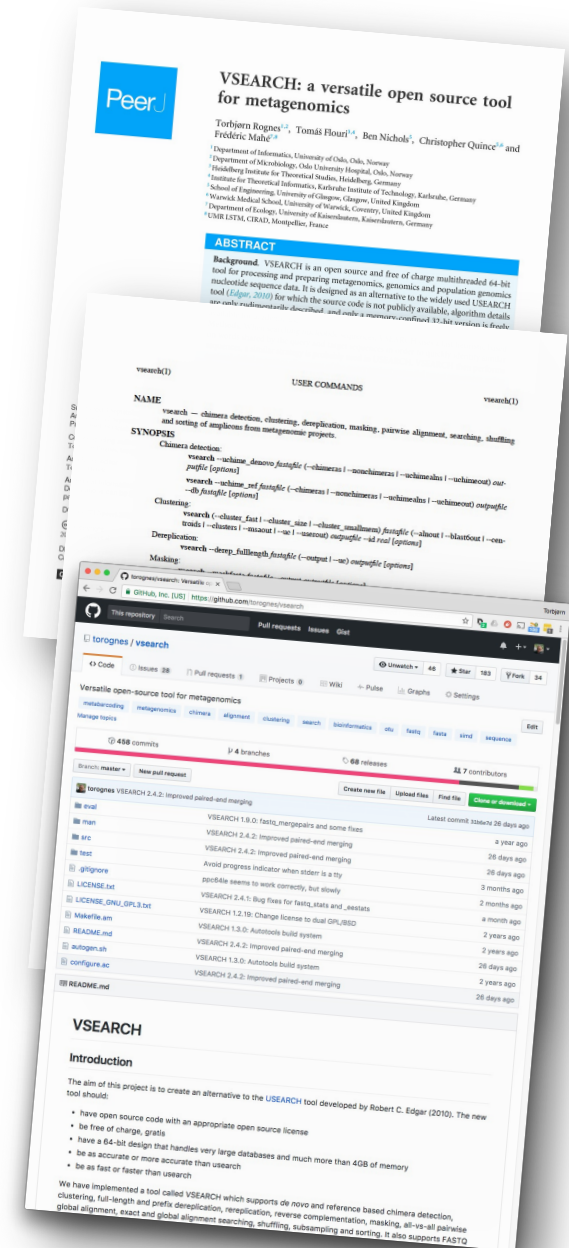
# Using pipes and compressed files

```
cat input.fastq.gz | \
vsearch --fastx_uniques - \
  --gzip_decompress \
  --sizeout --quiet --fastaout - | \
vsearch --cluster_size - \
  --id 0.97 --sizein -sizeout --centroids - \
> otus.fasta 2> errors.txt
```

- Use shell operators to set up pipes (|), redirect input (<), redirect output (>) or errors (2>).
- Replace file names with “-” to read or write from standard input (stdin) or output (stdout).
- Use “--quiet” to silence messages usually written to stdout.
- Warnings and errors are written to standard error (stderr).
- Use the option “--gzip\_decompress” or “--bzip2\_decompress” to decompress input from stdin, otherwise decompression is automatic.

# VSEARCH availability & documentation

- VSEARCH source code: <https://github.com/torognes/vsearch>
- 64-bit binary binaries for Linux (Intel x86, ARM64, PPC, ...), macOS (Intel x86, Apple Silicon) and Windows available
- Dual open-source license: GNU AGPL v3 or BSD 2-clause
- May be used directly for clustering and chimera detection in mothur
- QIIME 2 plugin; Conda, Homebrew, Debian packages; Galaxy wrapper, ...
- 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](https://doi.org/10.7717/peerj.2584) (>9000 citations)
- User manual with command and option details (over 50 pages)
- Extensively tested with >2000 unit tests
- Wiki, issue tracker, online support forum etc
- Suggest new features! Report bugs! Ask for help!



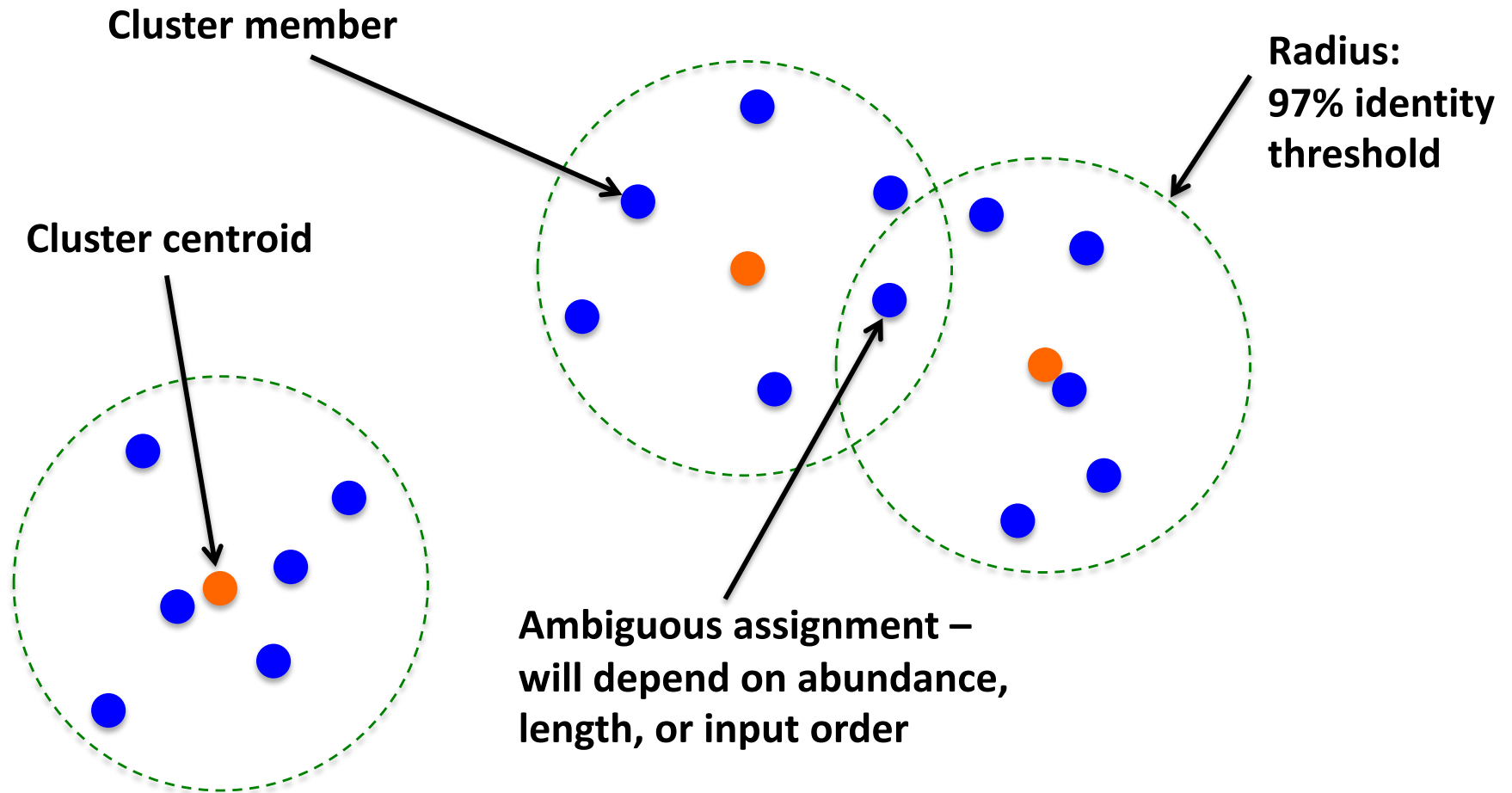
# Rsearch

- Rsearch is an R package designed for handling and analysing targeted sequencing data
- R-interface to VSEARCH
- Includes additional functions for plotting, optimization of sequence trimming and filtering thresholds, long read analysis, etc
- Developed by Cassandra Hjortdahl *et al.* at NMBU
- To be released soon.

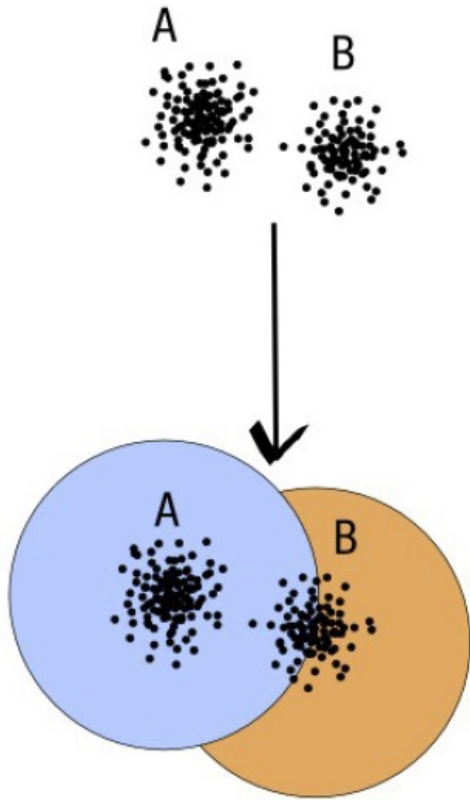
# Swarm: an alternative clustering method

- Alternative method for clustering amplicon sequences
- Clusters 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:
  - global clustering threshold
  - input order dependence
- High resolution
- Often generates many more clusters than other methods, may need post-processing, e.g. LULU/MUMU
- Very fast: Linear time and space complexity: fast and limited memory demands. Allows huge datasets to be clustered in reasonable time

# Heuristic centroid-based clustering



# Clustering thresholds



compromise threshold  
un-adapted threshold

defined by maximum  
distance from centroid

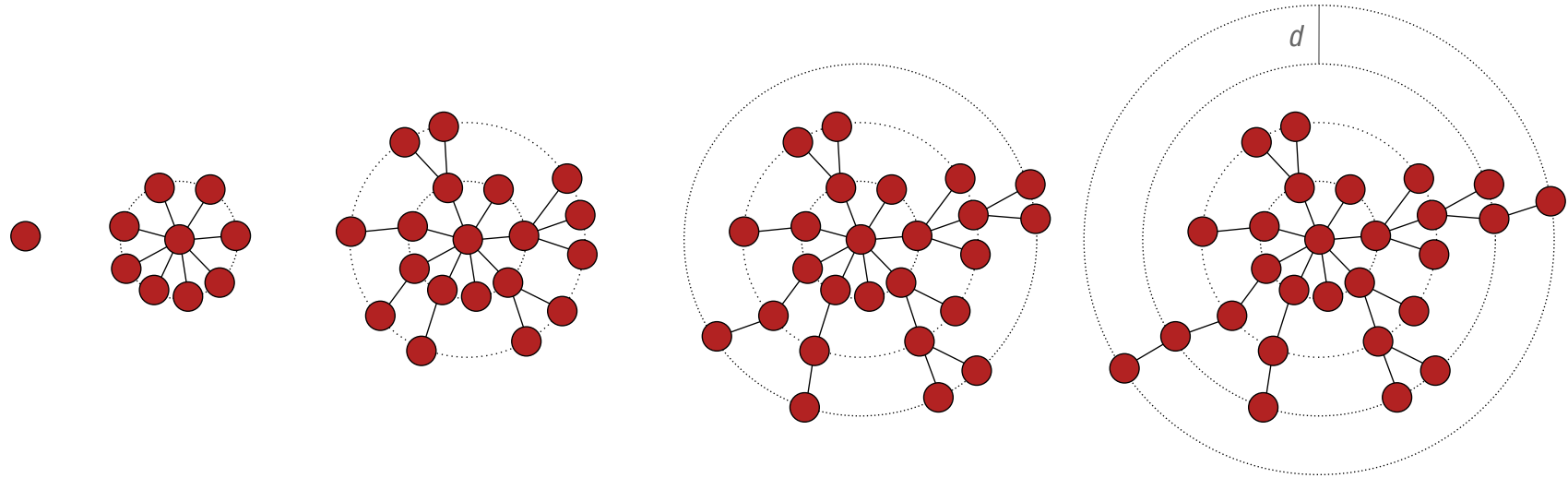


natural limits of OTUs

defined by minimum  
distance separating clusters



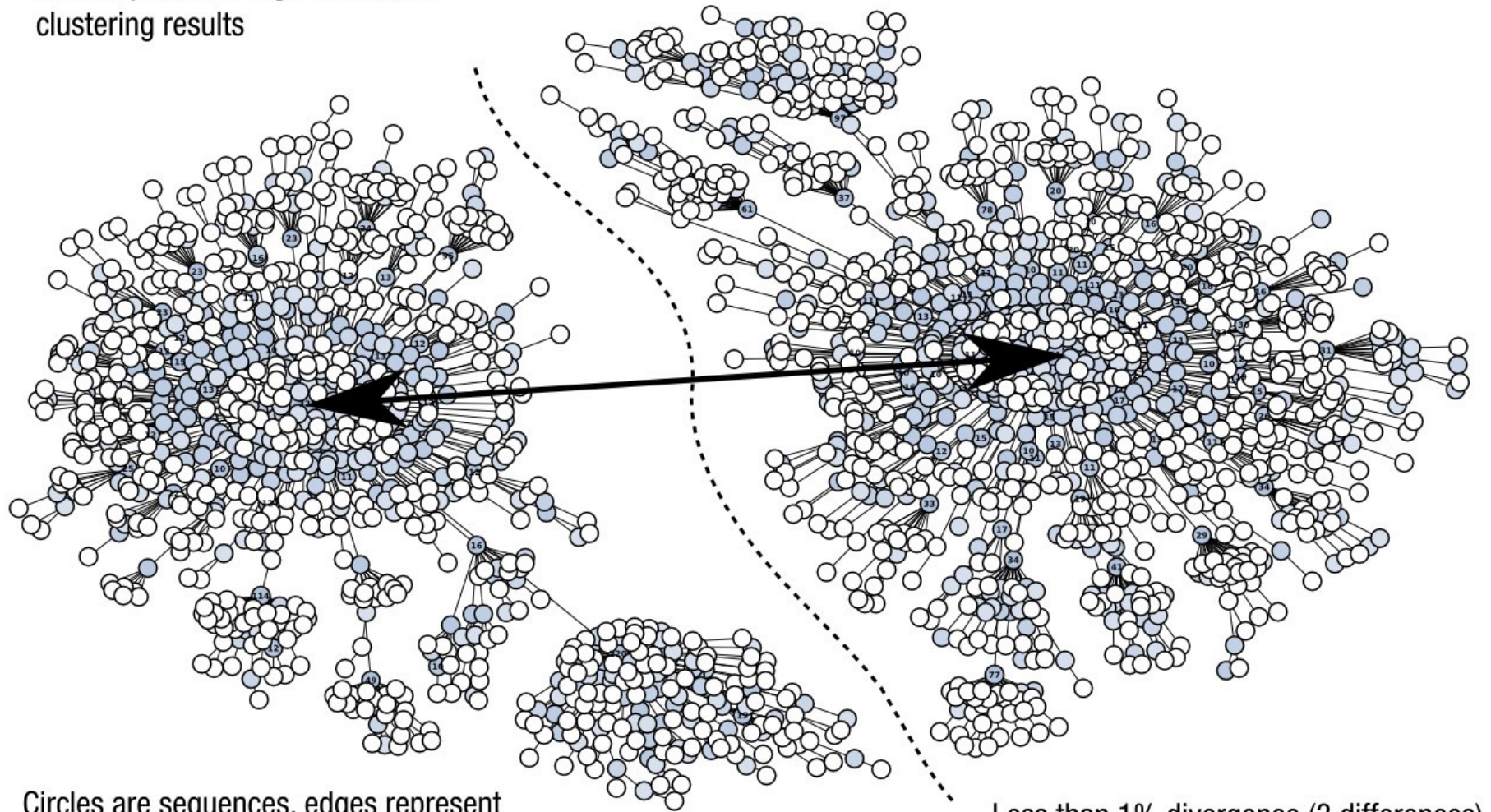
# Linking amplicons



- 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.

# Visualising the clustering results

Swarm produces high-resolution clustering results



Circles are sequences, edges represent one difference (substitution or indel)

Less than 1% divergence (3 differences) between the two peaks of abundance

# SWARM availability & documentation

- Available on GitHub: <https://github.com/torognes/swarm>
- Free of charge
- GNU AGPL v3 open-source license
- Extensive documentation
- Modern C++ code
- Extensively tested with >800 unit tests
- Precompiled 64-bit binaries for Linux (Intel x86, PPC, ARM), macOS (Intel x86, Apple Silicon) and Windows
- Publications (>1600 citations in total)
  - 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.
  - Mahé F, Rognes T, Quince C, de Vargas C, Dunthorn M. (2015)  
**Swarm v2: highly-scalable and high-resolution amplicon clustering.** PeerJ, 3, e1420.
  - Mahé F, Czech L, Stamatakis A, Quince C, de Vargas C, Dunthorn M, Rognes T (2022)  
**Swarm v3: towards tera-scale amplicon clustering.** Bioinformatics, 38, 267-269.

# Main collaborators

**Frédéric Mahé**

CIRAD, Montpellier, France

**Lucas Czech**

Heidelberg Inst. for Theoretical  
Studies, Germany

**Tomáš Flouri**

Heidelberg Inst. for Theoretical  
Studies, Germany

**Christopher Quince**

Warwick & Glasgow Univ., UK

**Håvard Kauserud, Anders K.**

**Krabberød, Embla E. Stokke, Inger**

**Skrede & Ella Thoen**

IBV, UiO

**Lars Snipen, Hilde Vinje,**

**Cassandra Hjortdahl, Knut Rudi**

NMBU



An underwater photograph showing sunlight rays filtering through the water, creating a serene and deep blue environment. The rays are visible as bright, diagonal beams of light against the darker blue background of the water. The water surface is visible at the top, with ripples and reflections of light.

**Thank you!**