

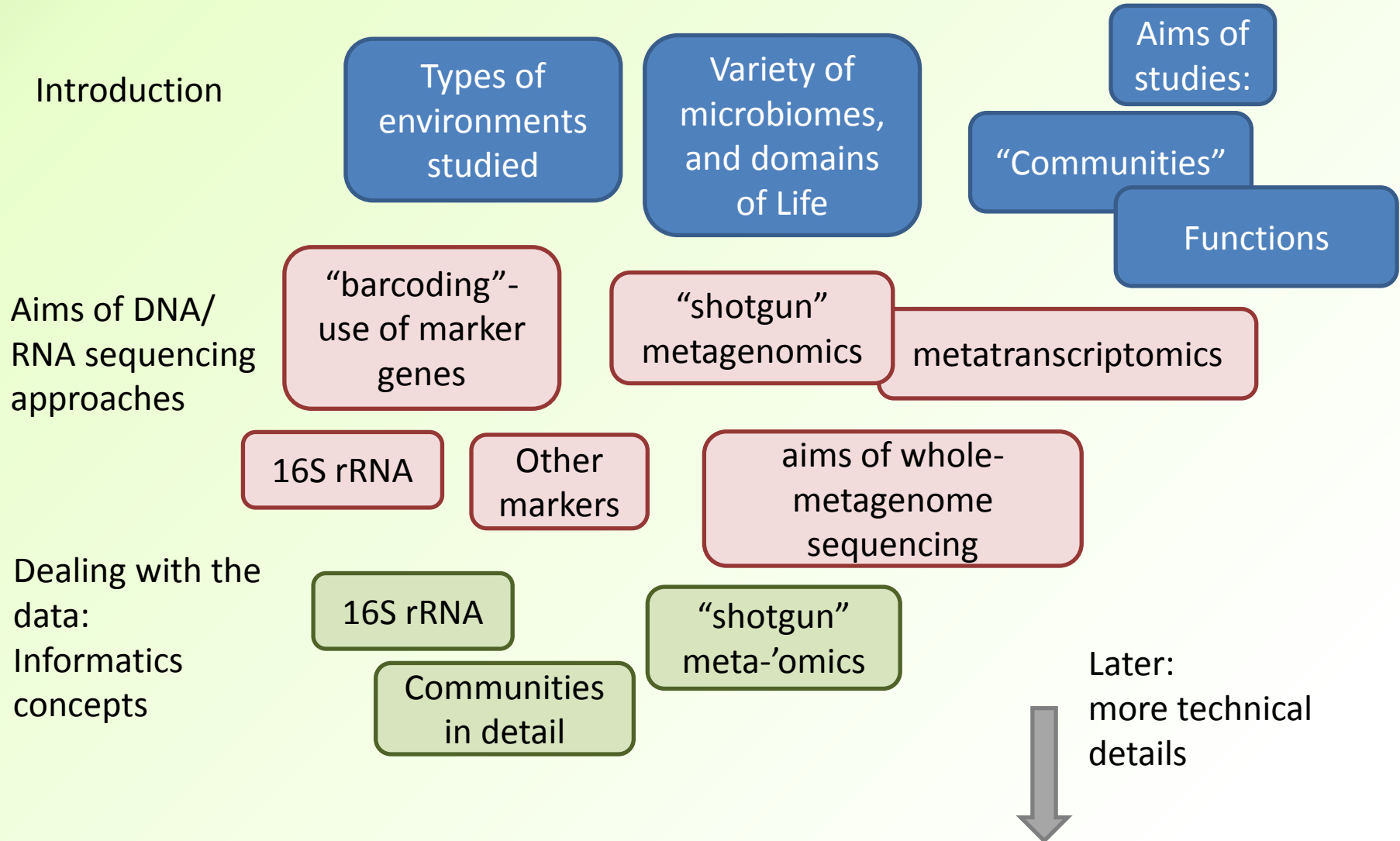
Introducing Microbiome Bioinformatics

Part 4.

Recap: Aims

- **Microbiome analysis**
 - with particular regard to **sequence informatics concepts**
- “Top down” – putting analysis tools and resources in context
- No highly detailed technicalities (yet)
 - No instructions on how to run particular programs
- Why you are using the bioinformatics approaches you use; pros, cons; alternatives

Topics, top-down



Series of talks

- 3 so far
- Open ended... as long there is demand
- Expected to be every 2 weeks, but all dates will be confirmed in advance
 - *Bite-size bioinformatics mailing list*
- The next few will cover: *(not necessarily in this order...)*
 - 16S analysis for community profiling
 - Classification issues (taxonomies etc)
 - Analysing richness and diversity of those communities
 - Dealing with sequencing and other errors
- Informal and flexible
 - Please interrupt and ask questions
 - Suggestions for topics for further focus

Series of talks

- Part 1: 27/1/2017
 - “Biological and Experimental Stuff that a microbiome bioinformatician needs to know”
 - Overview of marker gene sequencing for community analysis
- Part 2: 10/2/2017
 - Overview of whole-metagenome sequencing
- Part 3: 24/2/2017
 - Focus on metatranscriptomics
- Slideshows
 - <http://ghfs1.ifr.ac.uk/ghfs/>

- **Who is in there?**

- In what amounts?

Analysis of **marker genes** ("barcodes")
e.g. for **prokaryotes**: 16S rRNA gene
"16S-barcoding"

Marker-gene barcoding

Who is in there?

**COMMUNITY
ANALYSIS**

What *are*
they doing?

Metatranscriptomics

Metagenomics

What *can*
they do?

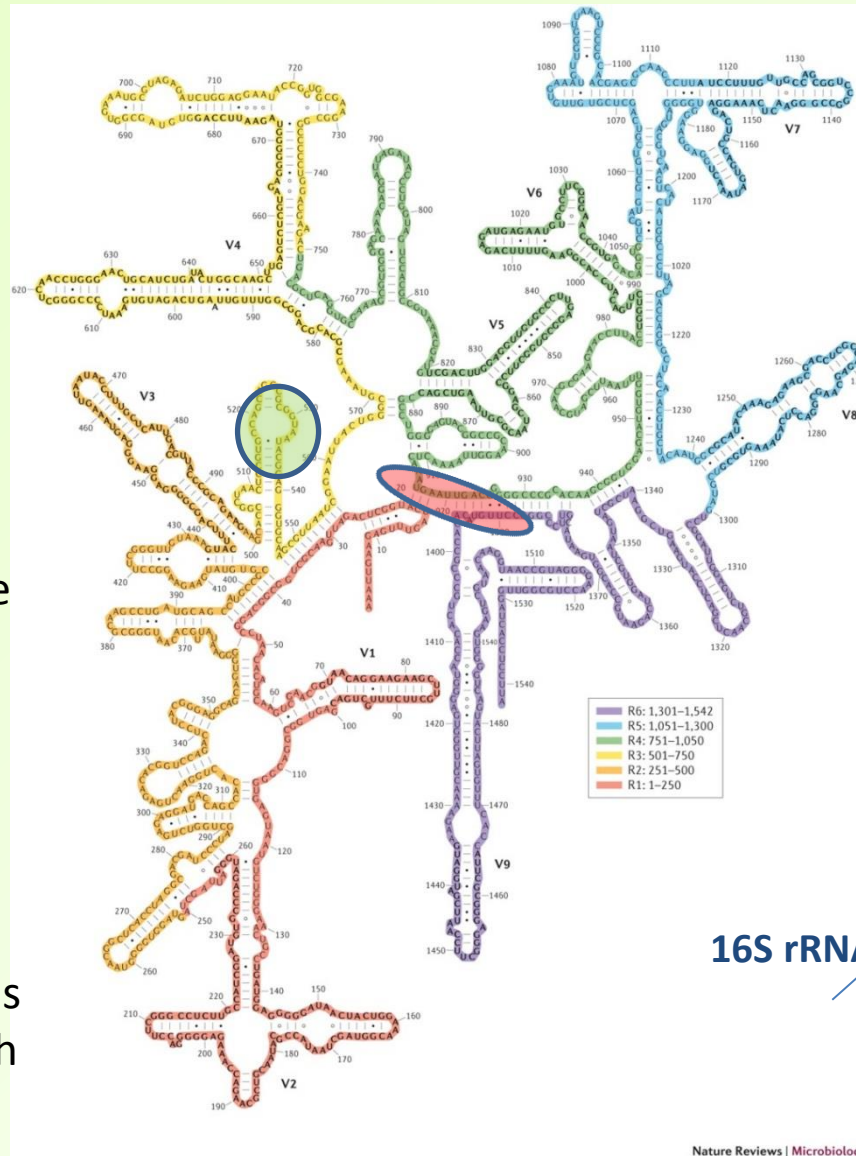




Amplification of a segment of the gene which codes for a **variable** region of the 16S rRNA molecule

→ Primers

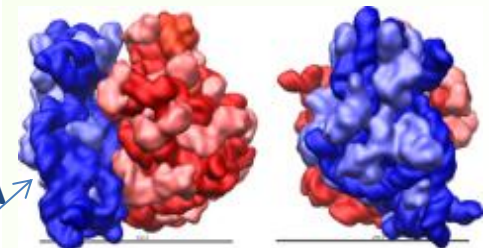
The variable region is chosen to distinguish between taxa



marker gene
("barcode")
for *phylotypes*

gene which codes for...

16S rRNA



Nature Reviews | Microbiology

Community analysis by marker-gene sequencing

Raw, unlabelled reads

Label to indicate bug of origin

In silico labelling

One of a
variety of
methods....

Names could be of an
externally defined
organism, e.g. from a
taxonomy

e.g. "*Lactobacillus reuteri*"
"unclassified Lactobacillales"
etc

Name1

Name2

Name3

Name3

Name1

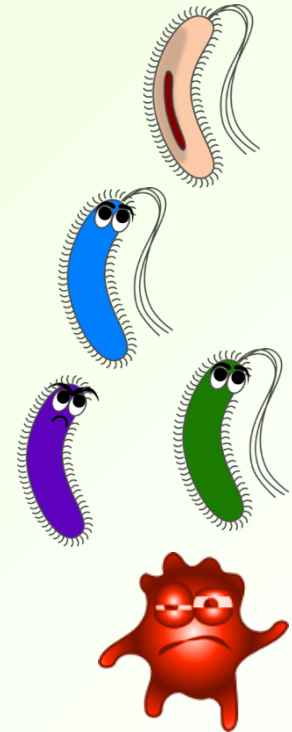
Name2

Name4

...etc..

Or could be **completely
anonymous**, a name
existing only within
your data e.g.
"OTU5432"

- Diversity studies still
possible



First - some considerations

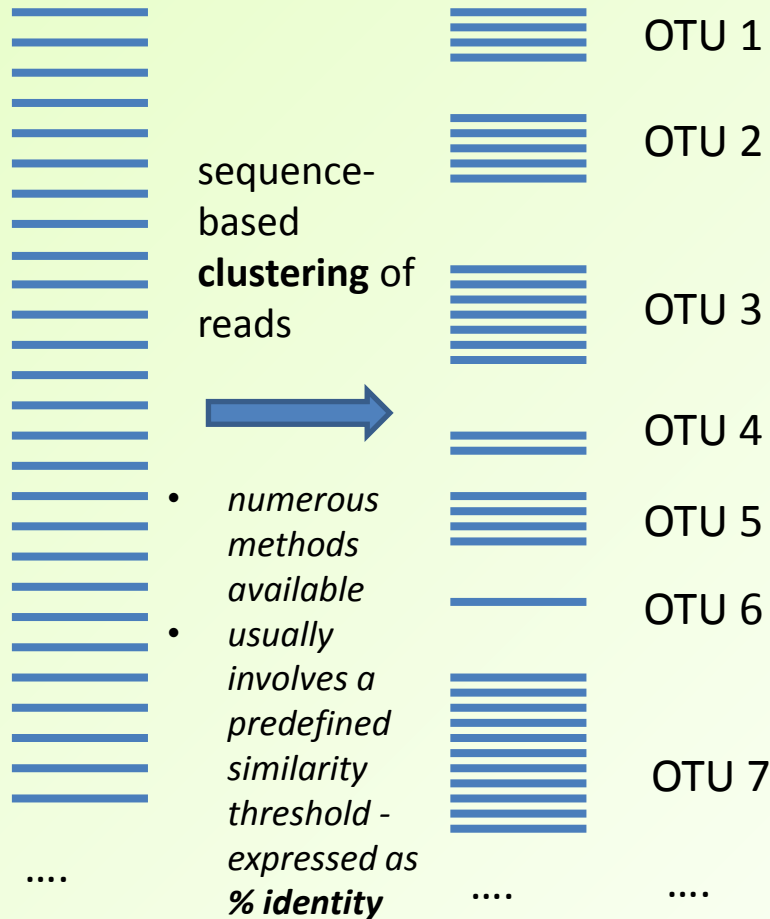
- Using predefined taxonomies
 - Sequences in a **reference database** have taxonomic annotation
- Using Operational Taxonomic Units (OTUs)
 - **What are OTUs?**
 - (for later:
 - **What do they represent?**
 - **Relationship with predefined taxa?**)
- Comparing reads with a database
- Self-comparison (clustering) of a set of reads

So what about all these
methods?

One approach:
Clustering reads into OTUs

collection of reads –
must be
homologous (e.g.,
all are amplicons of
the same 16S
variable region)

Clusters = Operational
Taxonomic Units (OTUs)



Number of
OTUs =
**measure of
richness**

This might be all
you want to do =
end of the analysis.

*Or, if you have
multiple samples,
then you may want
to:*

(2) Assess
differences in
diversity between
sample

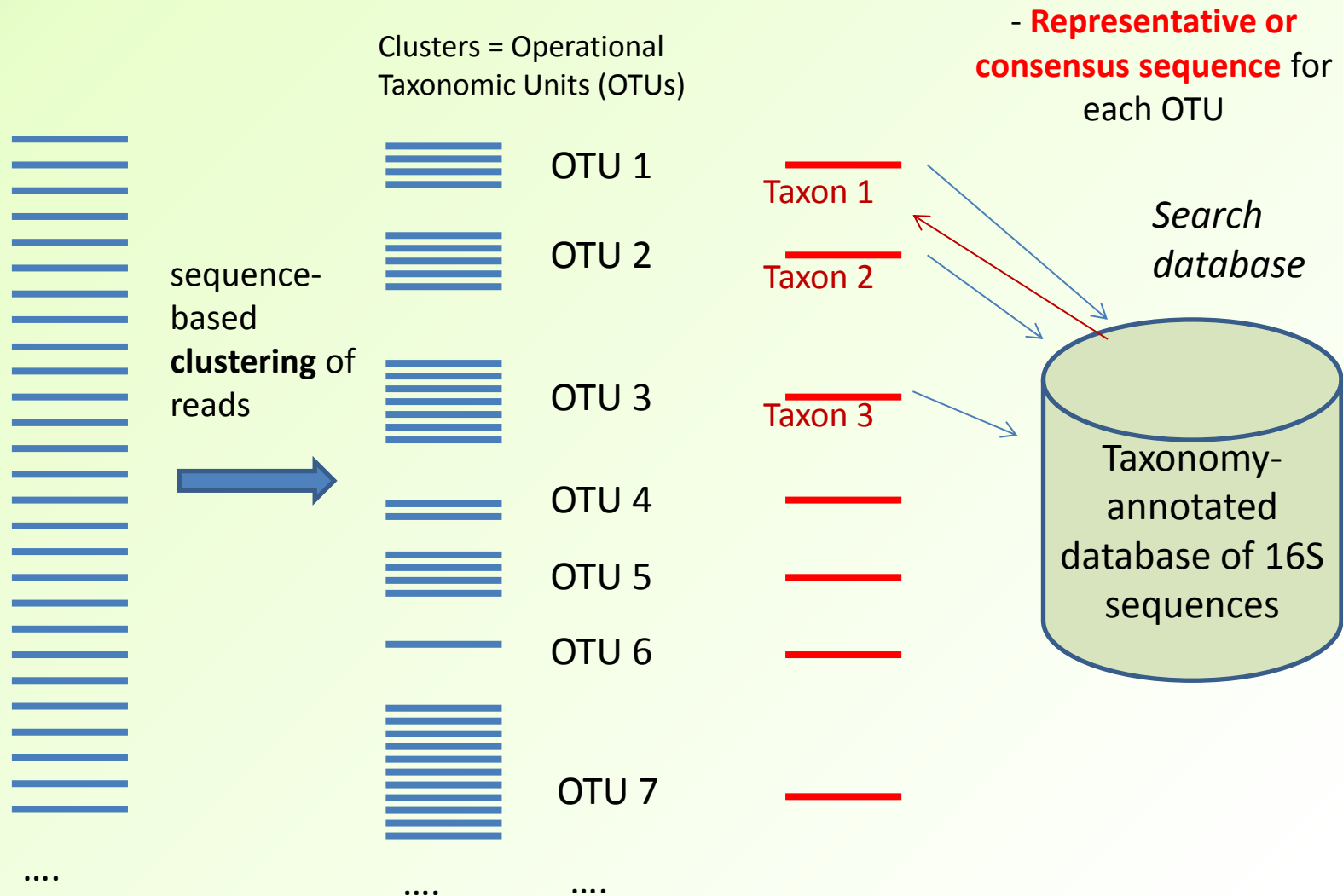
(3) Compare **actual OTUs**
(presence/absence, or
proportions, between samples)

Distribution of
proportions of
OTUs =
**measure of
diversity**

**No
reference
database
needed**

(1) Compare
numbers of OTUs
between samples

Usually (always?) we will also want to **identify** the OTUs
- in terms of a known taxonomic group

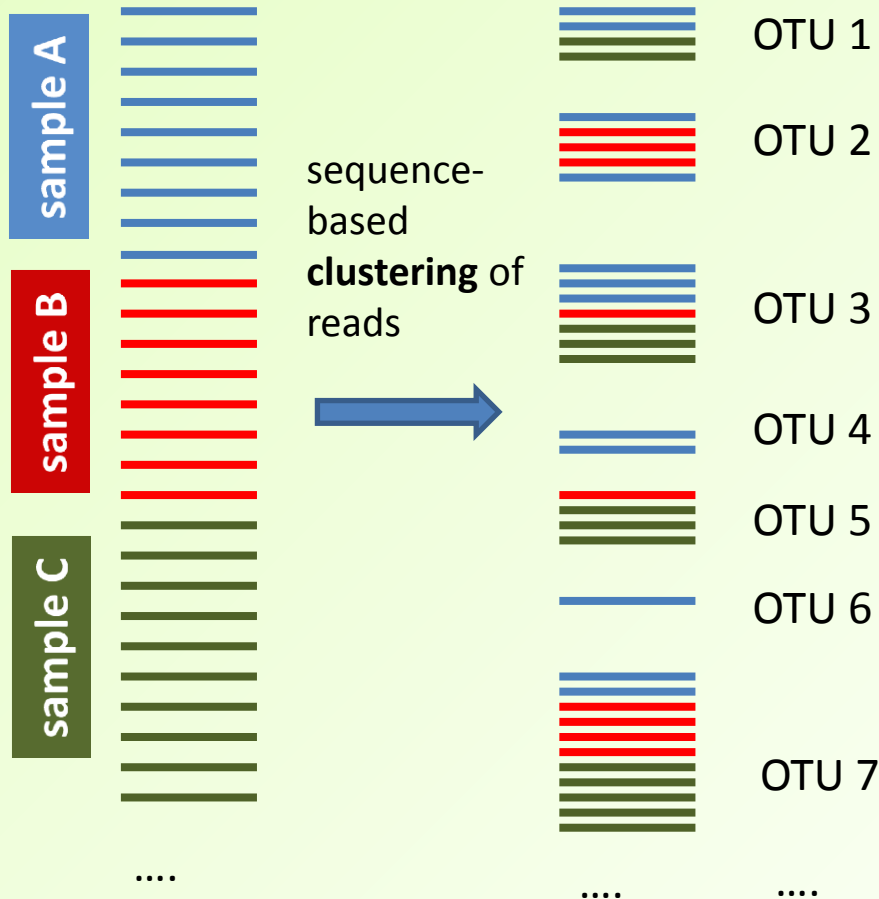


OTU assignment

- “OTU-assignment” is used here to describe the placing of each sequence read into a particular OTU
 - Once that is done, you know **which of your reads are in the same OTUs** as each other
 - **How many different OTUs** there are, etc
- In the general case, that process is distinct from the *identifying* of those OTUs
 - “**Identifying**” necessarily means **using some sort of reference**
 - But - assigning reads to OTUs, and identifying the OTUs, **can** be done as part of a single process
- It all depends on which approach is taken

Clusters = Operational
Taxonomic Units (OTUs)

Dealing with multiple samples



- Ideally, as many sequence reads as possible should be clustered in the same operation
- Which OTU a read is assigned to ***can depend on which other reads are present*** in the clustering exercise
 - May depend on the clustering method used
- So, cluster all your samples' reads together

So, what *are* OTUs?

What do they represent?

How do they relate to taxa?

...and what's so special about the
number 97?

For now....

- A 97% sequence-identity threshold is often used when clustering reads into OTUs
 - **How** this threshold is used depends a lot on the clustering method
- “Notionally”, the resulting OTUs are roughly equivalent to species...
 - But actually, it’s far from that simple
- Also... 97% identity... of what?

- Different methods (even using the same 97% threshold) can produce very different numbers of OTUs

[Please refer to original image in this publication:

http://www.nature.com/nmeth/journal/v10/n10/fig_tab/nmeth.2604_F2.html

from Edgar
(2013) *Nature
Methods* **10**
(10) 996-8

- But does this actually matter?
 - Discuss....next time

Other methods

Not clustering

Taxon-based

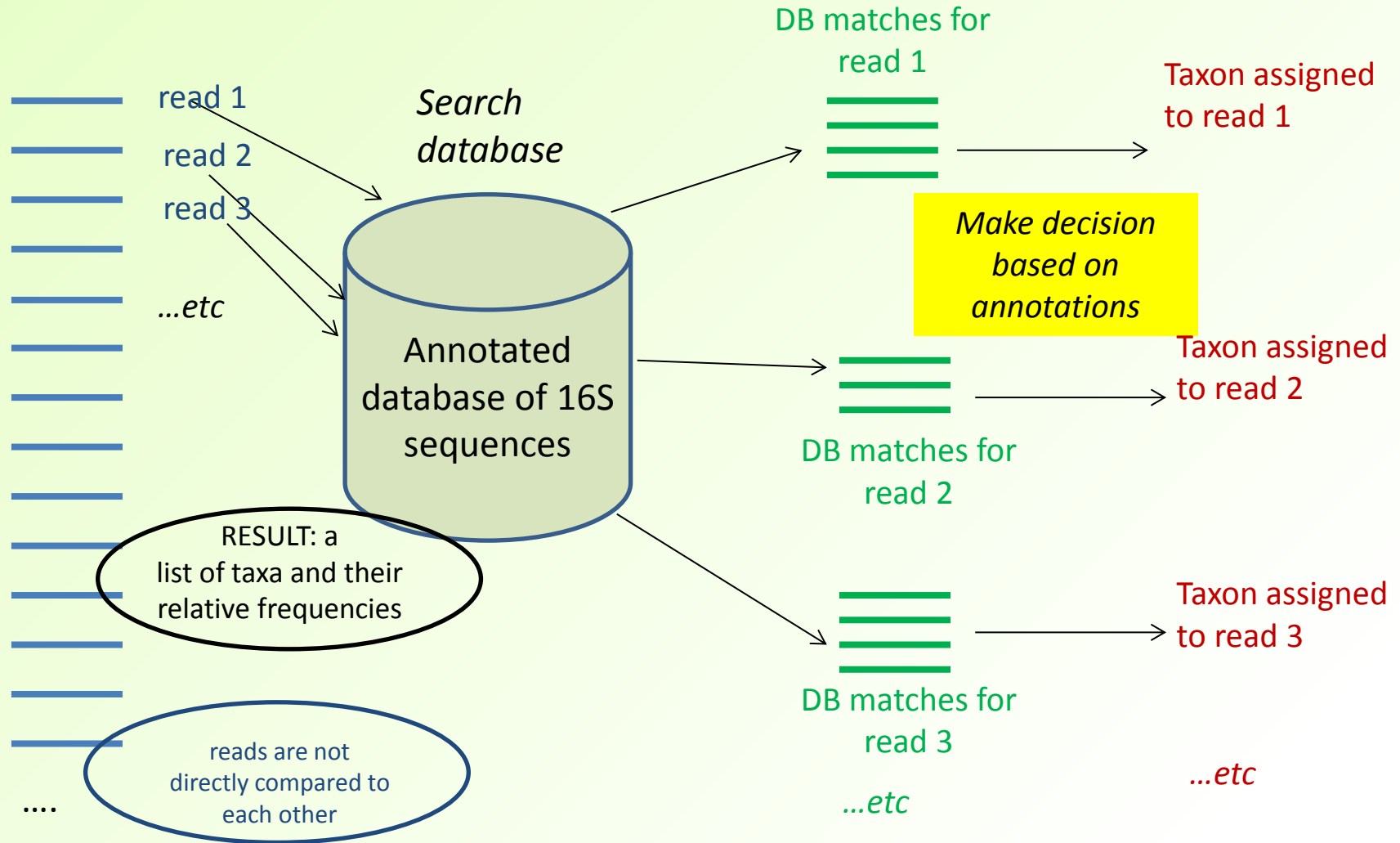
Not OTU-based

Other methods? (not clustering)

- Some methods are a read-by-read approach (“process one read then the next one...”)
 - That is, **each read is processed independently of all the others**
 - This means the process is easy to **parallelise** :
 - many or all reads can be processed at the **same** time
- These methods necessarily involve comparing **each read** with sequences in a reference database
 - There are different approaches
 - In terms of the databases used
 - And **how** the sequence comparison is done

collection of 16S
reads

Assigning a taxonomic classification to each **individual** read : ***one example approach***



Read-by-read methods

- The reads are each compared to a database, and not to each other
- So this is just as applicable to processing reads from shotgun metagenomics/ metatranscriptomics
 - (but is not the only way of doing this)
- One approach is simply a traditional sequence similarity search (e.g. **BLAST**)
 - With huge numbers of query sequences
 - And using a different database (i.e. not just 16S sequences!)
 - But **making sense of the list of hits can be far from straightforward**
- Can be computationally expensive –
 - if your read set is very large
 - and your reference database is very large too (choose wisely)
 - may actually take up more “wall time” than OTU-clustering, if a fast heuristic clustering method is used
 - unless you have a very large number of processors available
- More about this in future sessions

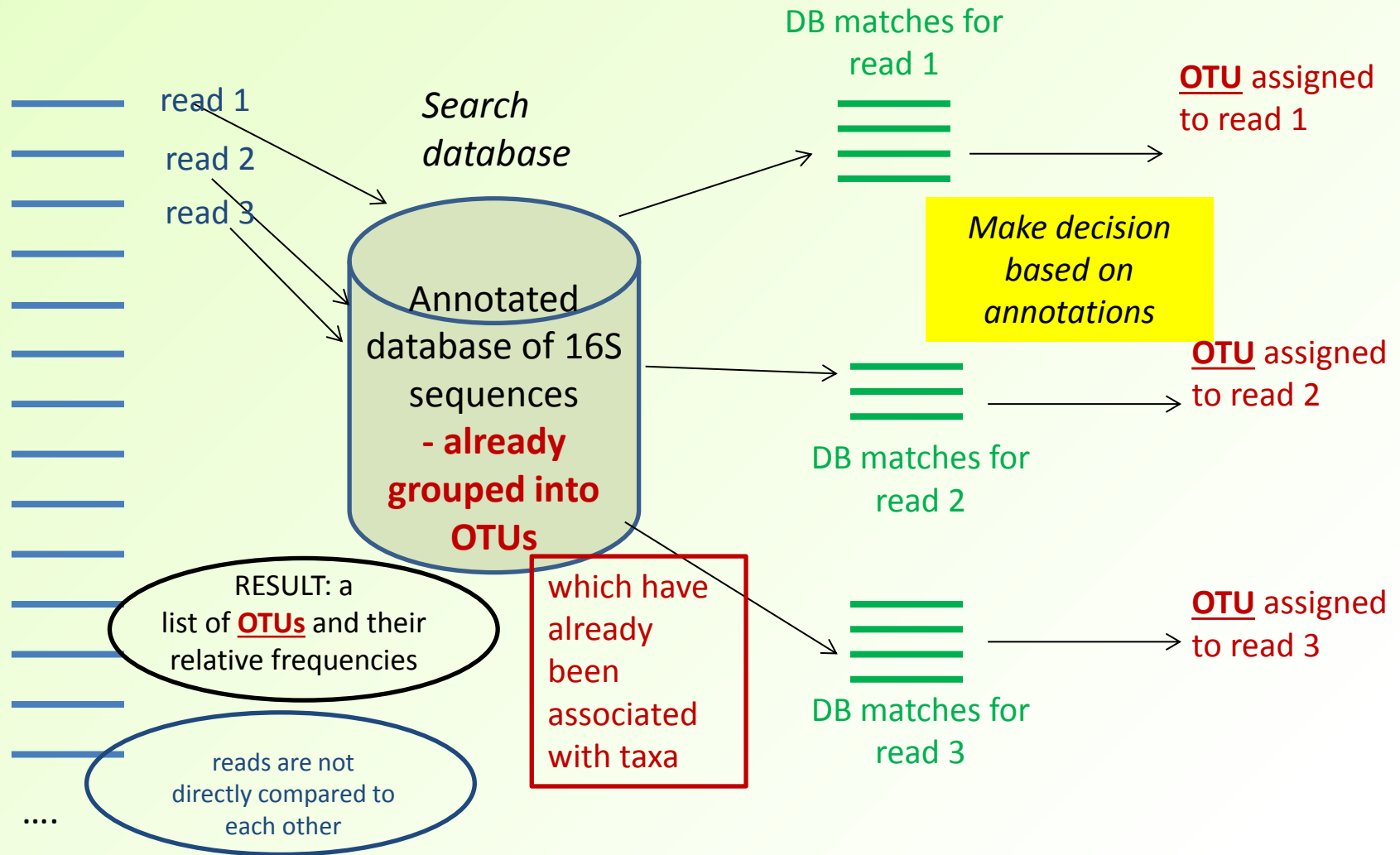
Back to OTU-assignment....

Other approaches to OTU-assignment

- “**OTU-assignment**” used here to refer to all methods of assigning your reads to OTUs
- Traditionally:
 - OTU-assignment == **OTU-clustering**
 - by whatever clustering method
- Comparison of read sets versus themselves:
 - problematic/impossible for extremely large numbers of reads
 - clustering: difficult to **parallelise** computationally
 - The nature of all-versus-all comparison
 - Some widely-used heuristic methods have been developed
- Has motivated alternative methods of OTU-assignment
 - Using reference databases – of known 16S sequences **already clustered into OTUs** by database curators
 - and the OTUs in the database also **annotated with taxa**

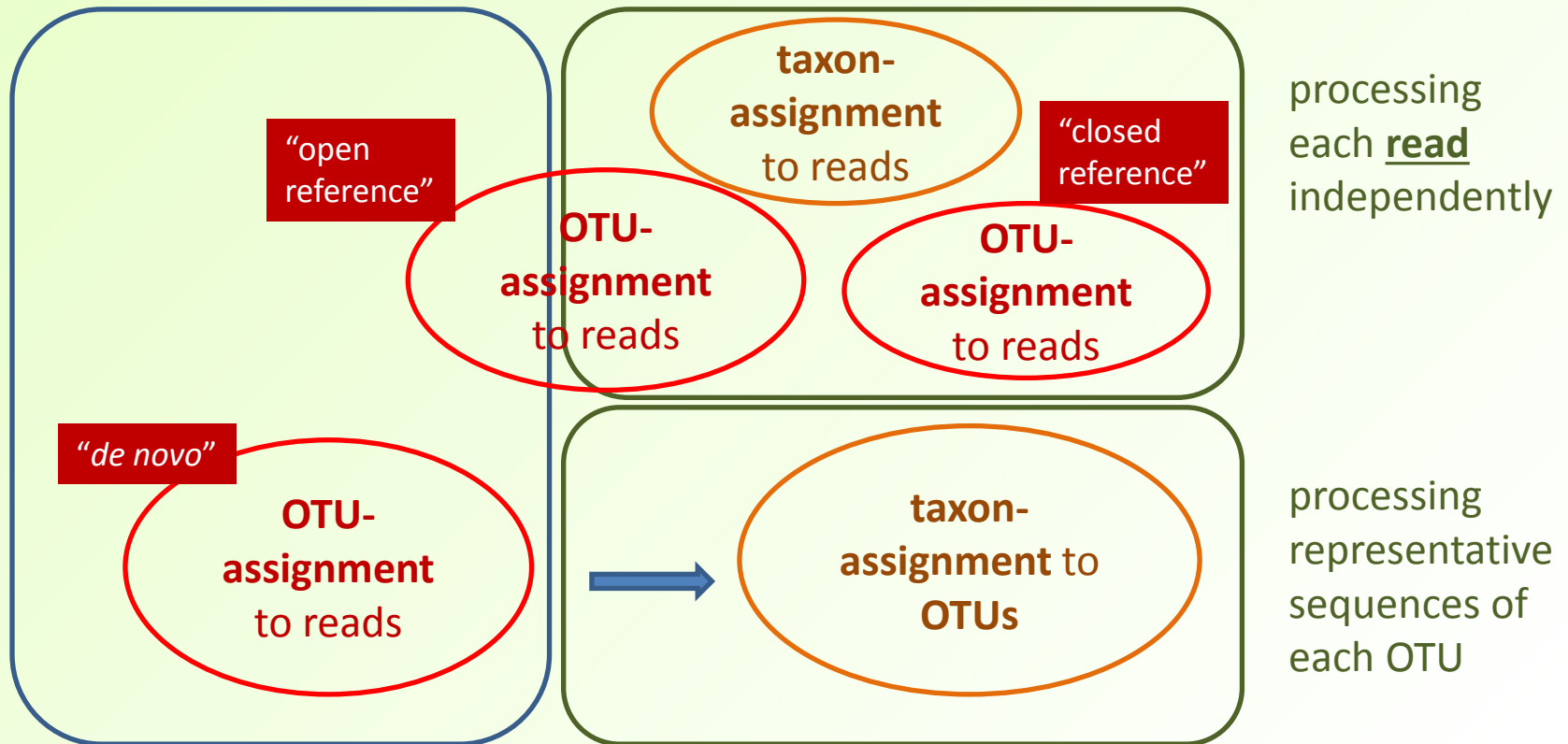
collection of 16S
reads

Assigning each **individual** read to an OTU: *one example approach*



Clustering :
comparing reads
with each other
(“self-referential”)

Using a reference
database



More terminology

- In some contexts, “**OTU-picking**” is used for OTU-assignment
- Potentially confusing: might imply that you are “**picking**” (**selecting**) particular OTUs for your data, from some larger list of OTUs
 - In fact, this is exactly what you *are* doing with some methods
 - such as **open- or closed-reference OTU-picking**
- in a different approach, ***de novo* OTU-picking**, there *is no list of OTUs* to select from:
 - The list of OTUs is self-generating, from within your data

de novo OTU-assignment

- Clustering reads; no reference database
- Advantages
 - Very systematic; transparent; database-independent
 - You won't get different results if you repeat later, when databases will have changed
 - ***Metrics of richness and diversity will remain unchanged***
 - (but you may well get changes in terms of the later ***identification*** step)
- Disadvantages
 - Speed: very difficult to parallelise
 - Collection of reads must be self-compared
 - Not a problem unless your datasets are huge
 - Does not take advantage of external expert “OTU-curation”

Closed-reference OTU-picking

- Compare each read to reference DB sequences
 - Discard reads which don't match
- Advantages
 - Potentially, speed: can be parallelised
 - Assignment/identification of each read is independent of others
 - Makes use of external curation: use of far larger database collections may enable better OTU-definitions
- Disadvantages
 - Completely dependent on a database
 - ***Even basic metrics of richness and diversity are subject to change, if the reference database changes***
 - If your reads cannot be closely matched to a database sequence, they are ignored
 - If you have novel organisms in your sample, this could be disastrous

Open-reference OTU-picking

- Initially takes the same approach as closed-reference OTU-picking
- But uses a *de-novo* approach for the unidentified reads
- Information on novel organisms will therefore be retained
- Still has the problems of complete database-dependence
- Still has the advantages of access to externally defined OTUs

Whether OTU-clustering or
independent read-by-read
approaches....

*That sounds like a lot of sequence
comparison*

How are the read sequences compared.... With DB sequences?

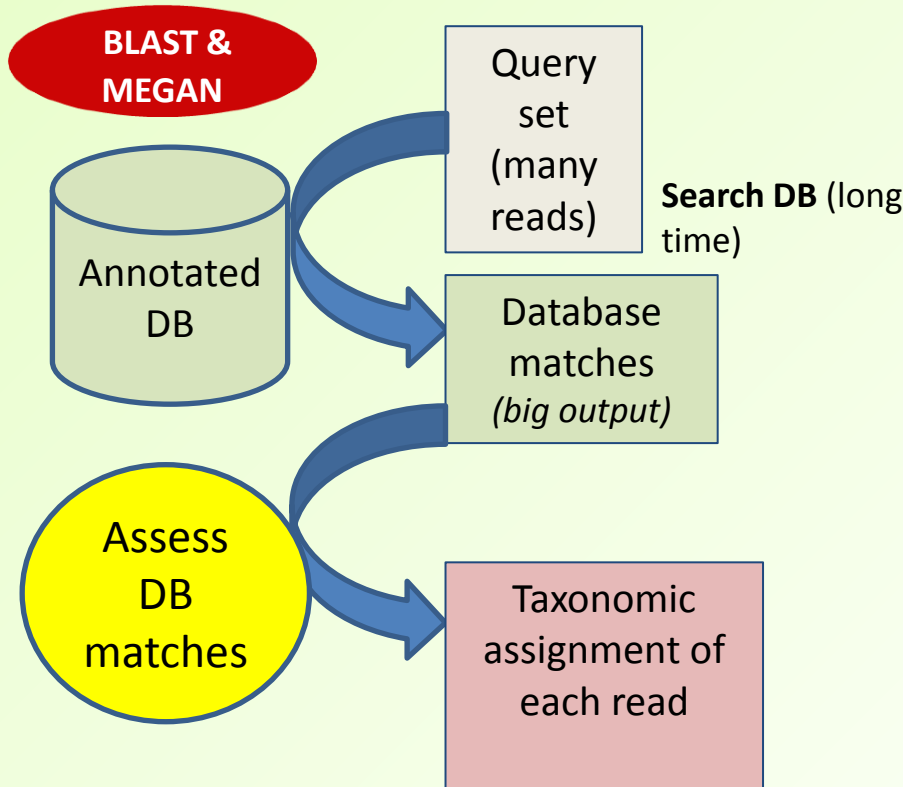
- Some methods use each sequence explicitly – they are **alignment-based methods**
- But:
- Do we need to compare every read sequence with the reference database?
 - This can be time-consuming
 - and produce very large data files
- Are there short-cuts?

Sequence features

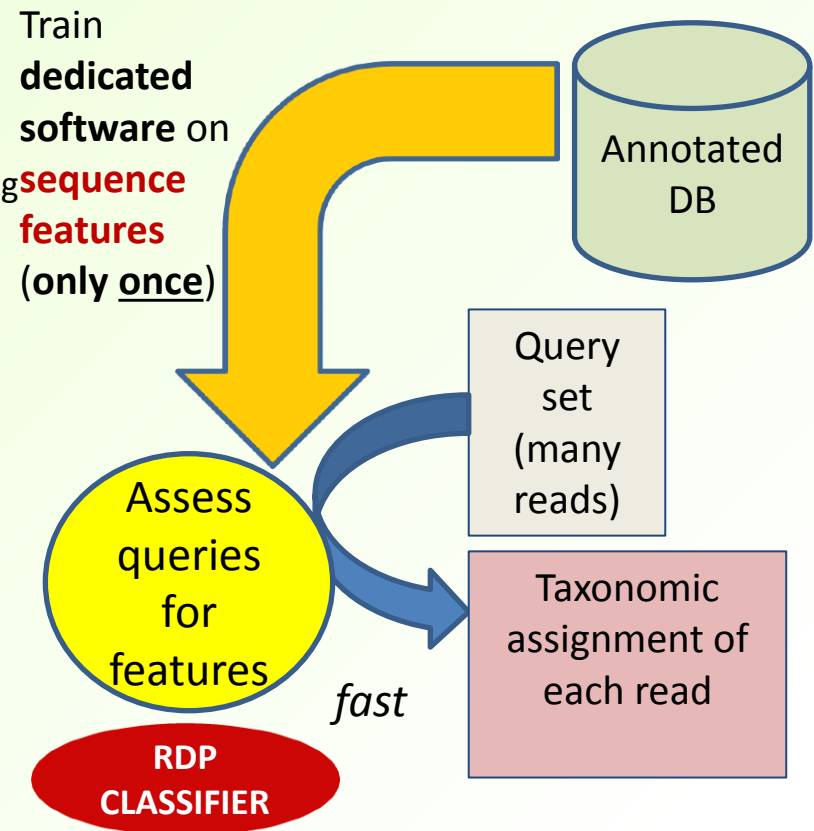
- An alternative to using the explicit DNA sequence itself
- For example, **composition** of the sequence
 - E.g. frequencies of particular k -mers
- Comparison between sequences' features is much faster than doing a sequence alignment
- May be pre-calculated for sequences in a large reference database
- Calculation for your query set of reads is relatively fast

Direct and indirect comparison with reference database

Direct: two user steps



Indirect: one user step



How are the read sequences compared.... With each other?

- As in, clustering
- Again, shortcuts are possible
- E.g. pre-screening a large set of reads –
 - using sequence features to filter out pairs of reads of no interest
 - i.e. which cannot possibly have sufficient identity (e.g. 97%)
- Then perform full comparison between pairs of reads which remain