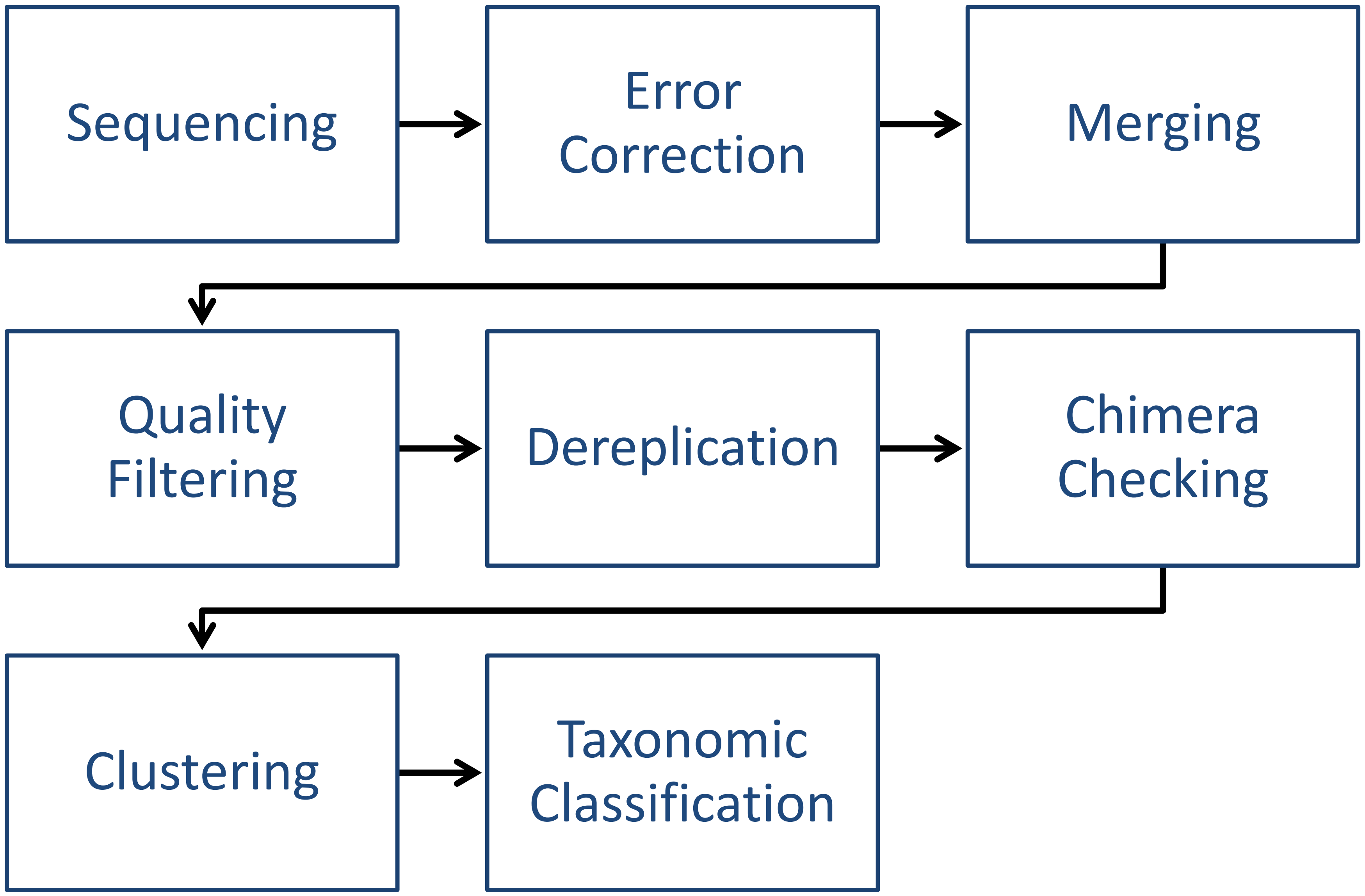
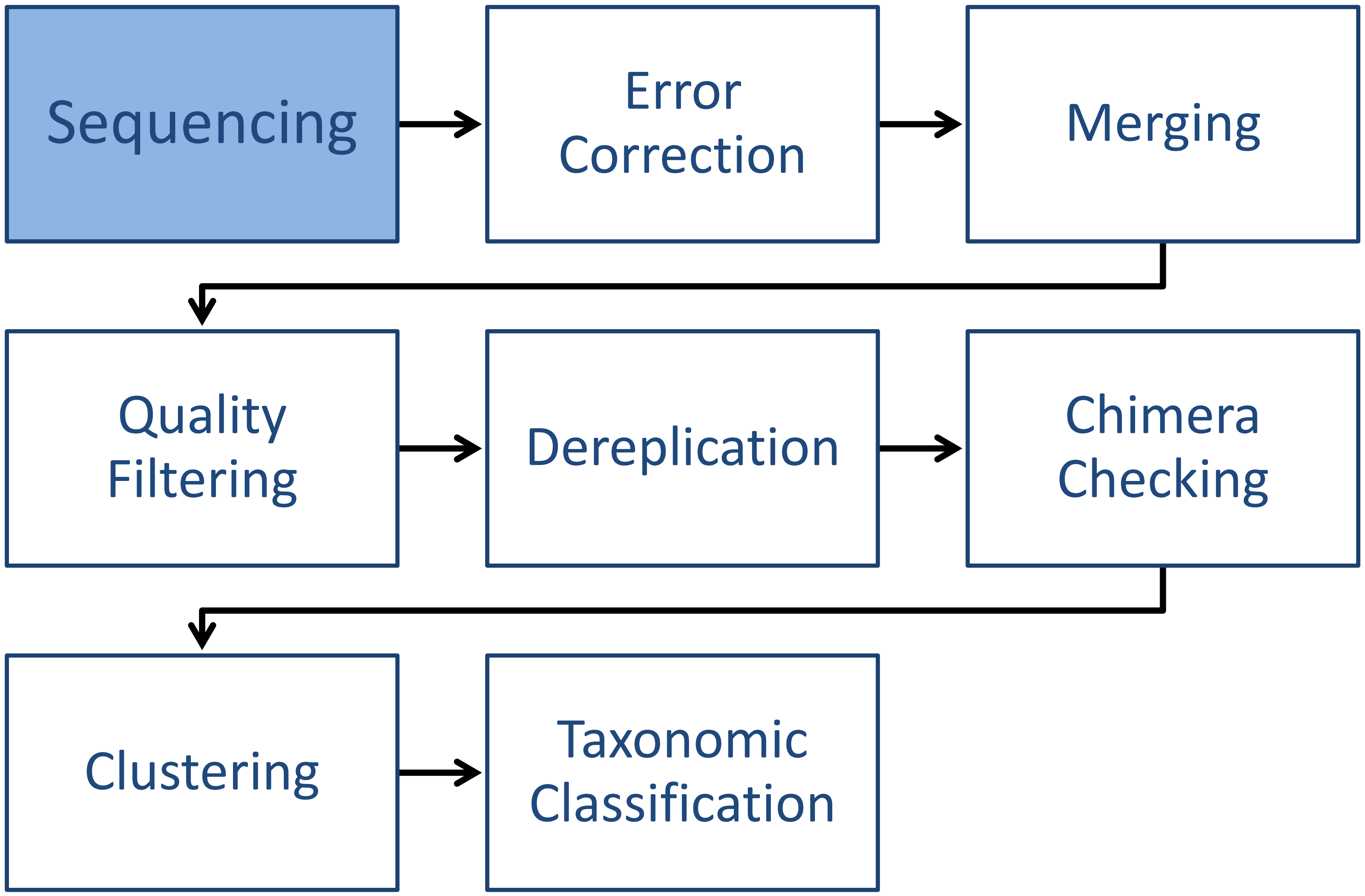
- Course material (lectures, data) will be available at https://github.com/bio9905/course-material


Sequencing
Error Correction
Merging
Quality Filtering
Dereplication
Chimera Checking
Clustering
Taxonomic Classification

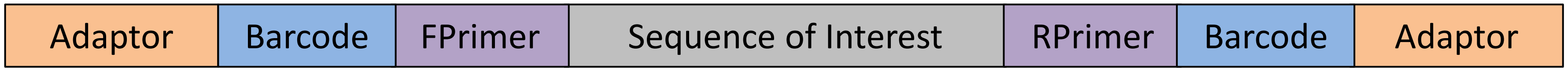

Sequencing
Error Correction
Merging
Quality Filtering
Dereplication
Chimera Checking
Clustering
Taxonomic Classification

NGS Amplicon Sequencing

- NGS platforms include: 454 pyrosequencing, Illumina, SOLiD, PacBio, Ion Torrent, Nanopore
- Most common platform for amplicon sequencing is paired-end Illumina

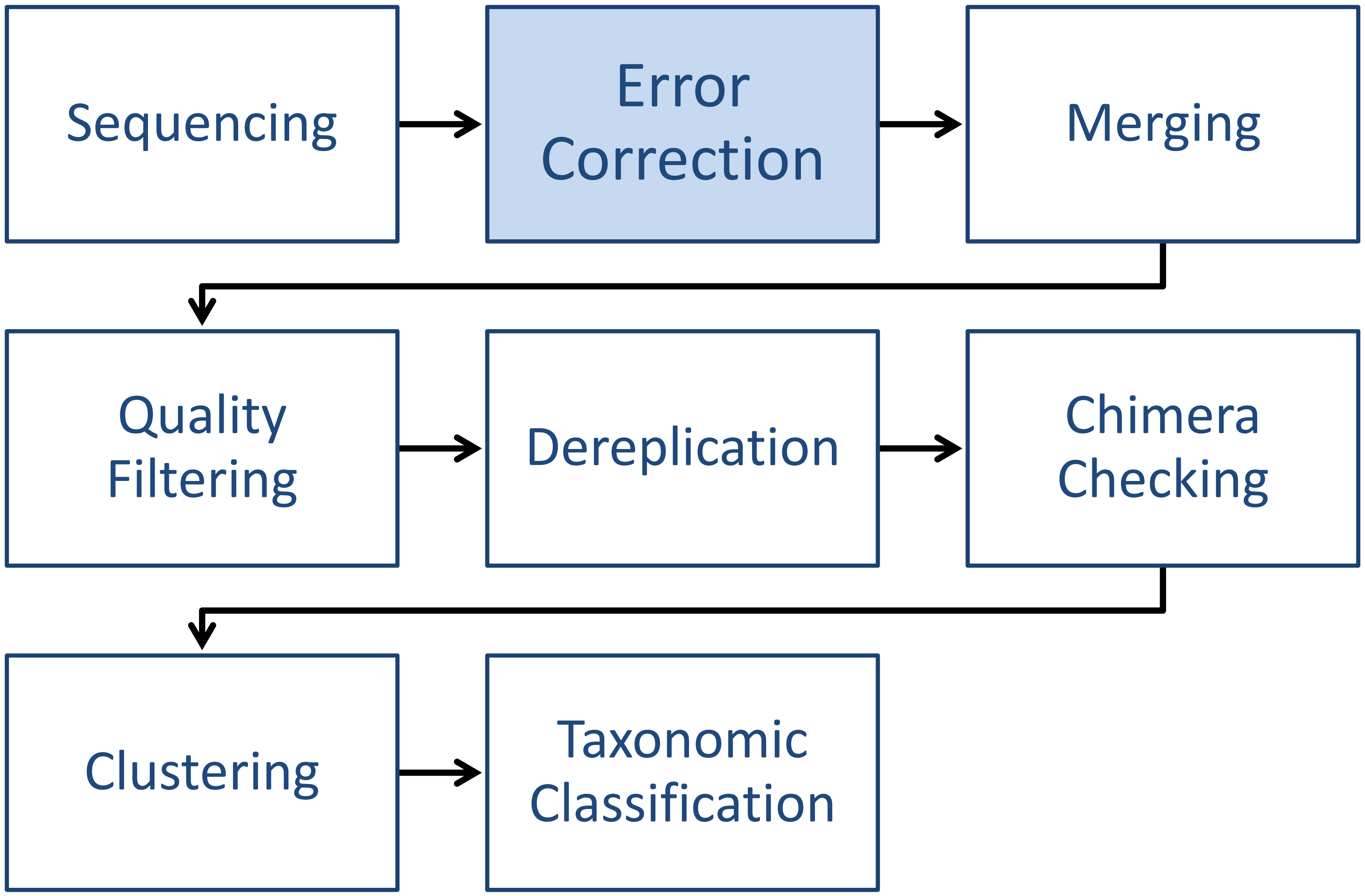
NGS Experimental Design

- Ecological studies require >1 sample or they are hopelessly statistically under powered
  - To allow multiplexing of many samples for ecological studies short indices of known identity are added onto each sample, allowing for sequence identification in downstream analysis
- Method of adaptor/barcode attachment can vary
  - PCR versus ligation approaches have different biases



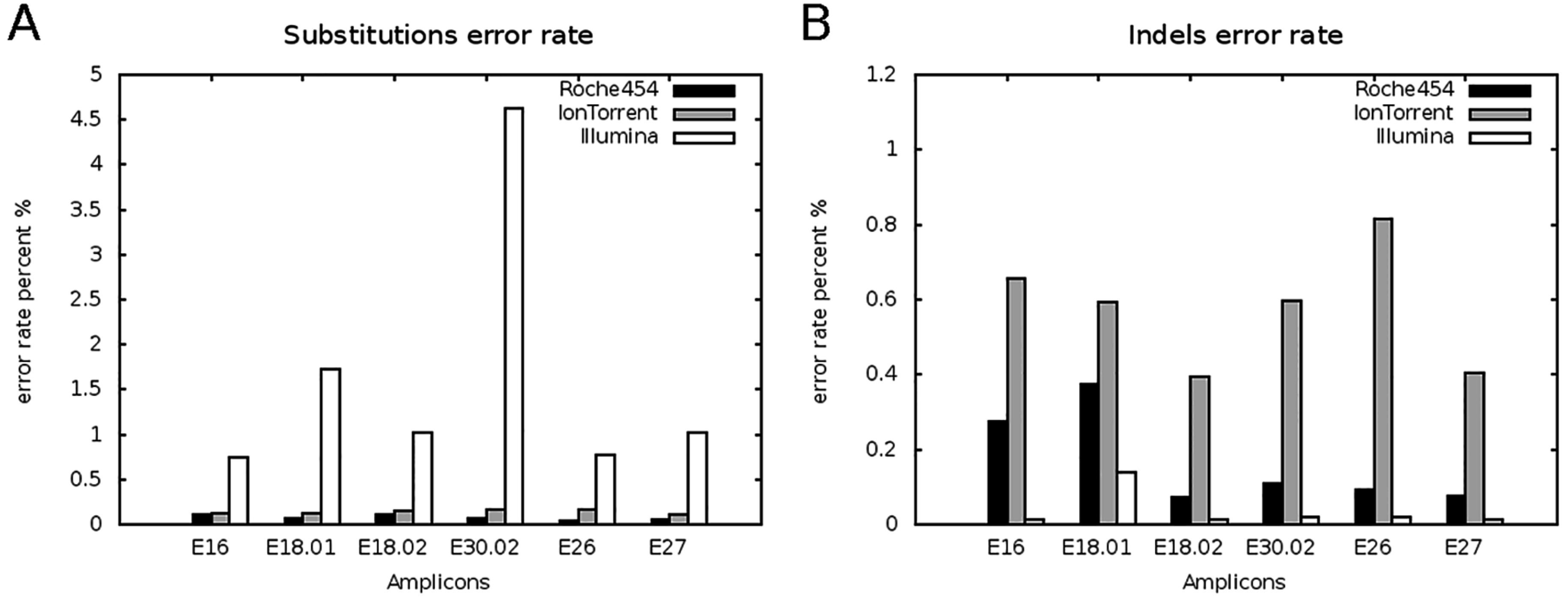
NGS Experimental Design

- Think before you sequence!
  - What's your question?
    - best gene region?
    - sequencing depth
  - Barcode selection
    - Dimerization
    - Colour balance (illumina)
    - Error correction
  - Internal standards


Sequencing
Error Correction
Merging
Quality Filtering
Dereplication
Chimera Checking
Clustering
Taxonomic Classification

Error Correction

- Primary error type in illumina sequencing is substitution miscalls
  - Substitutions caused by incorrect incorporation
  - Indels caused by polymerase slippage



Fuellgrabe et al. 2015

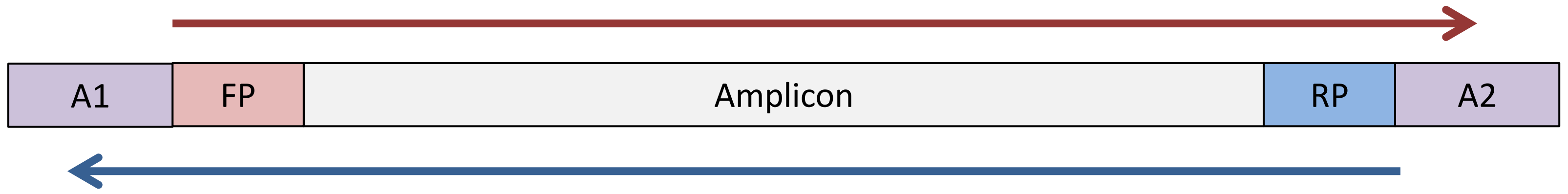
Error Correction

- Strategies for error correction:
  1. Contig building (mothur)
  2. K-mer correction (bayeshammer, trowel, DUDE-seq,quake)
  3. Machine-learning classification and clustering (IPED)

Error Correction: Contig Building

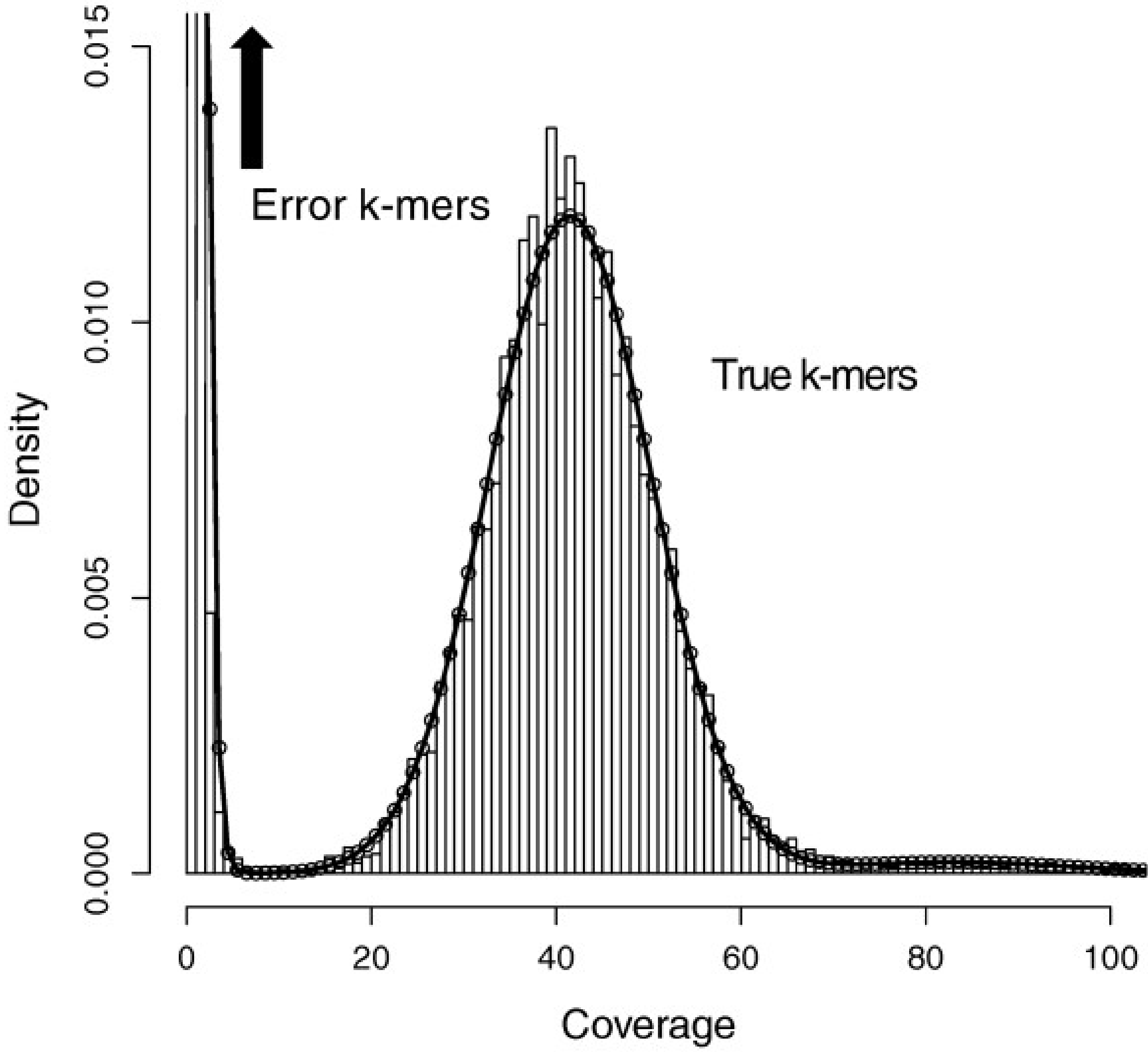
- Depends on complete coverage of the sequence of interest by both the forward and reverse reads
- Assumes that incorrect base incorporation typically results in a low quality score

Qual R1	35	35	35	35	35	35	35	35	35	35	35	35	35	5	35	35	35	35	35	35	35	35	35
R1	A	T	C	G	A	T	C	G	A	T	C	G	A	G	C	G	A	T	C	G	A	T	C
R2	A	T	C	G	A	A	C	G	A	T	C	G	A	T	C	G	A	T	C	G	A	T	C
Qual R2	35	35	35	35	35	10	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35
Corrected	A	T	C	G	A	T	C	G	A	T	C	G	A	T	C	G	A	T	C	G	A	T	C

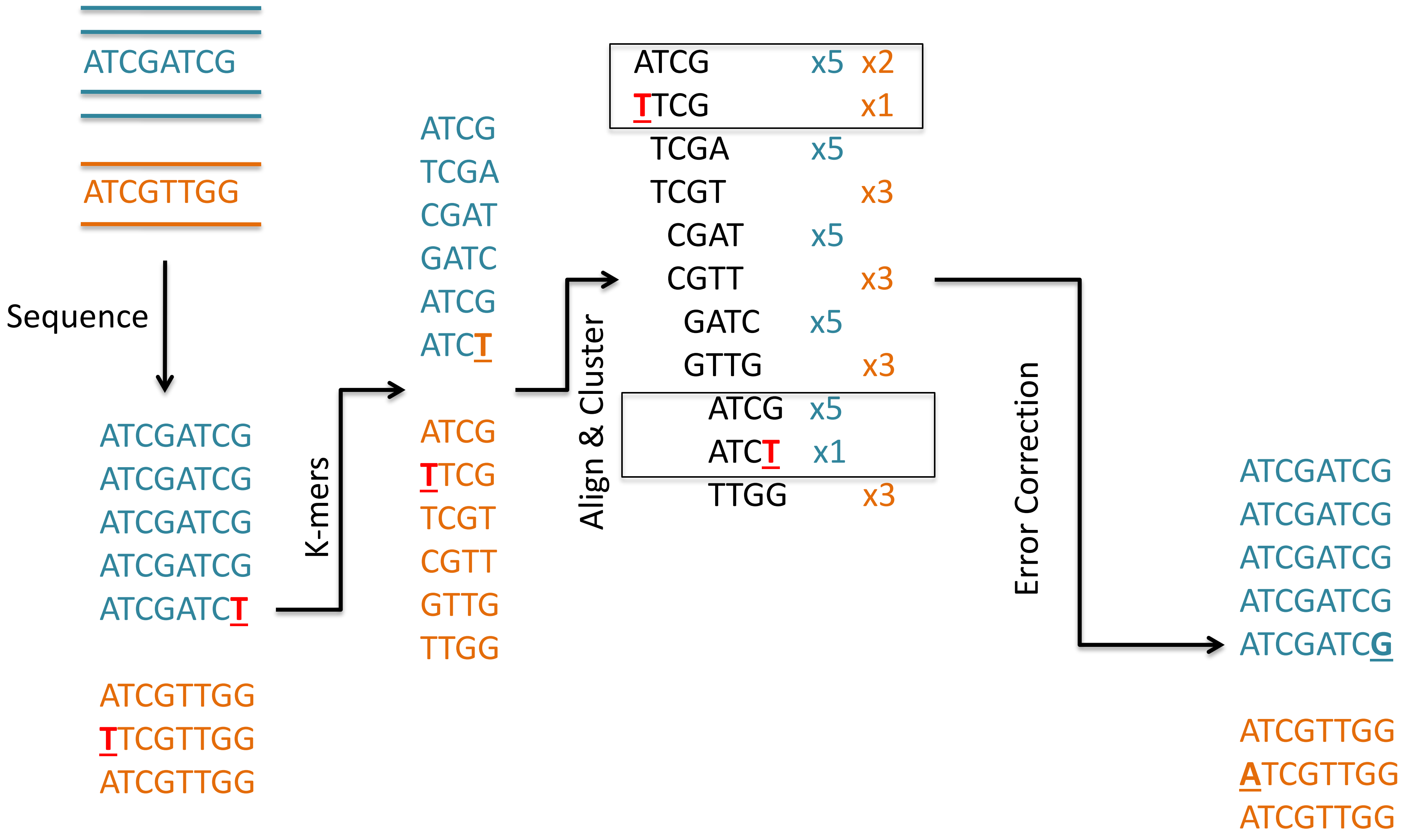


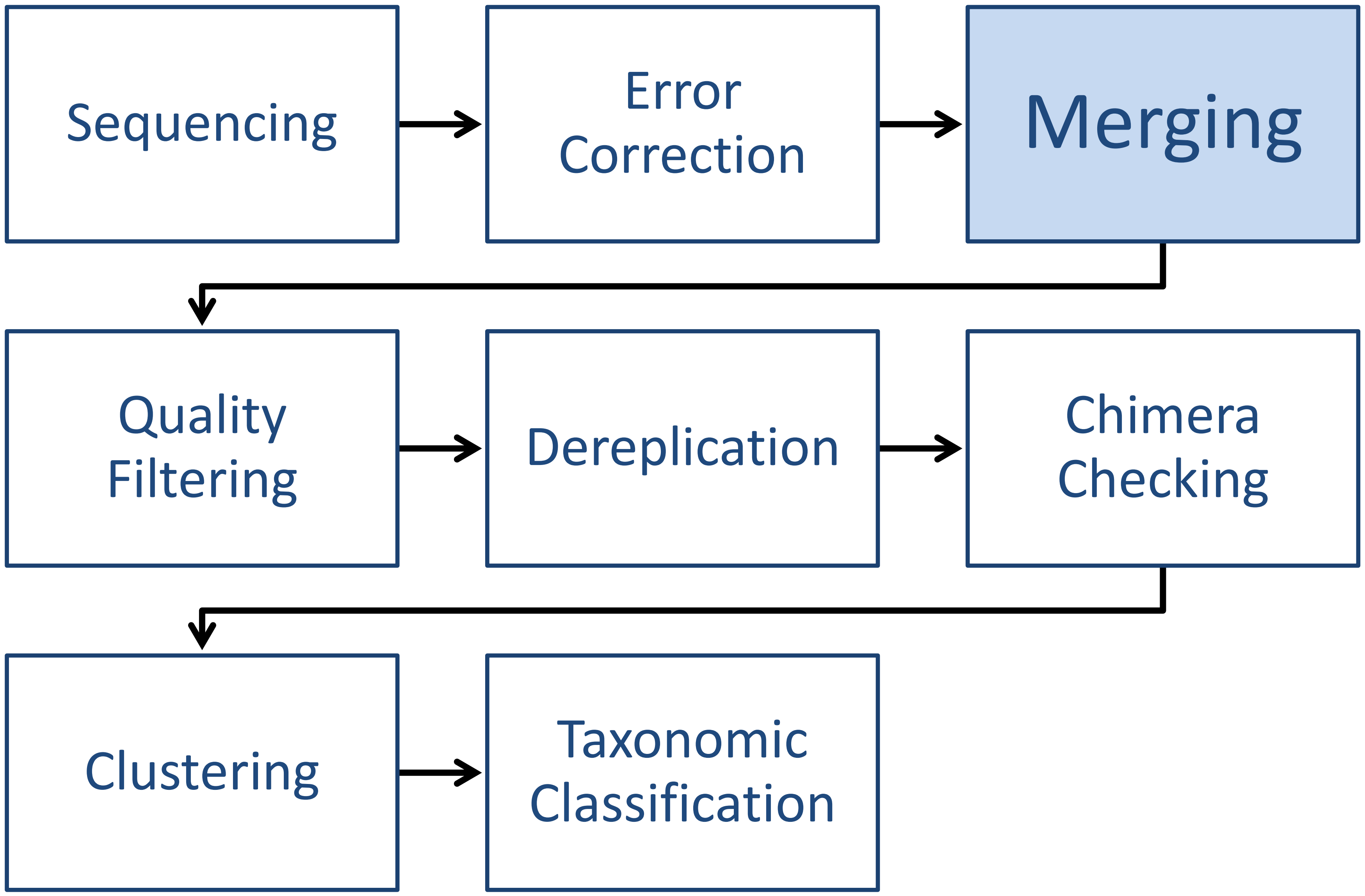
Error Correction: k-mers

- Divides sequences into bite-sized words (***k-mers***) and uses probabilistic methods to identify errors
- Built on the assumptions that errors are infrequent and occur at random



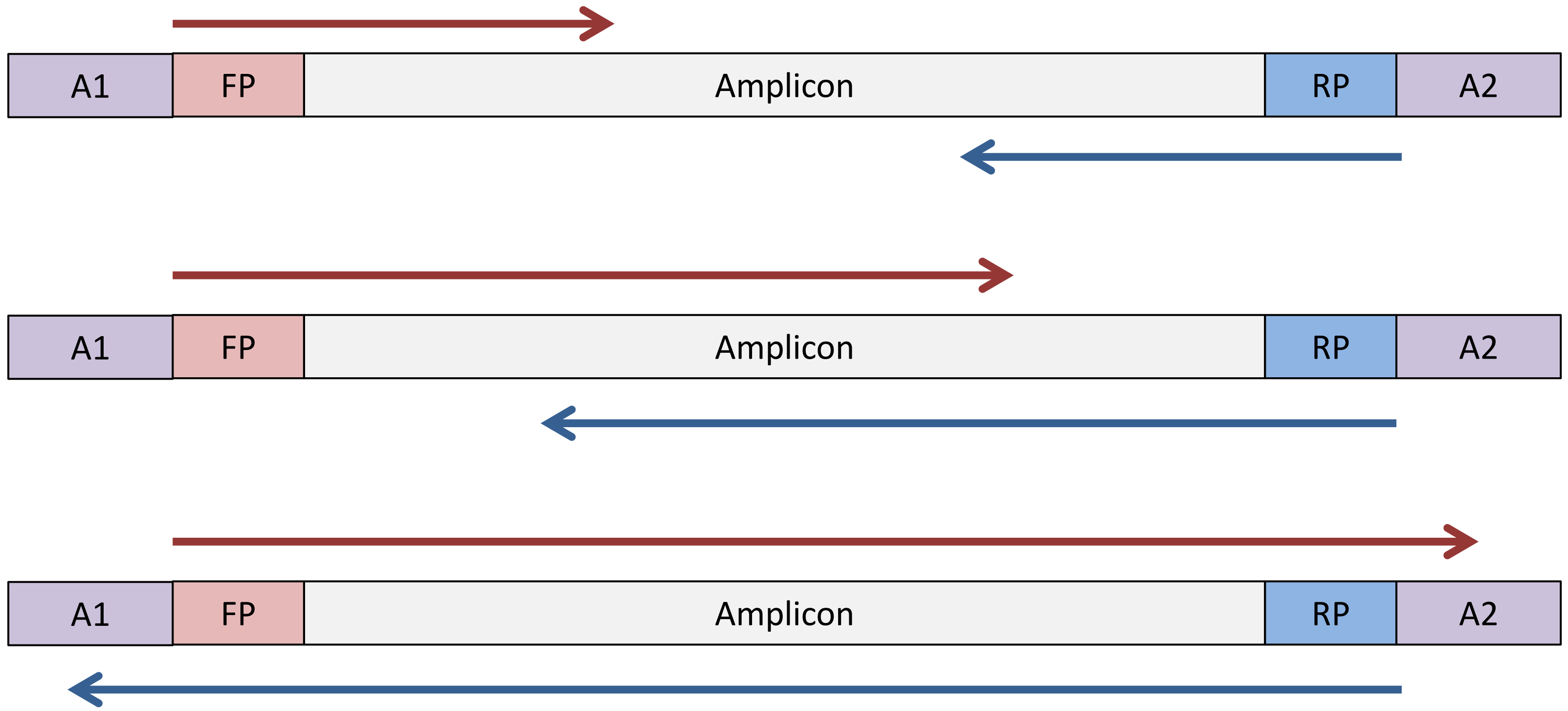
Error Correction: k-mers




Sequencing
Error Correction
Merging
Quality Filtering
Dereplication
Chimera Checking
Clustering
Taxonomic Classification

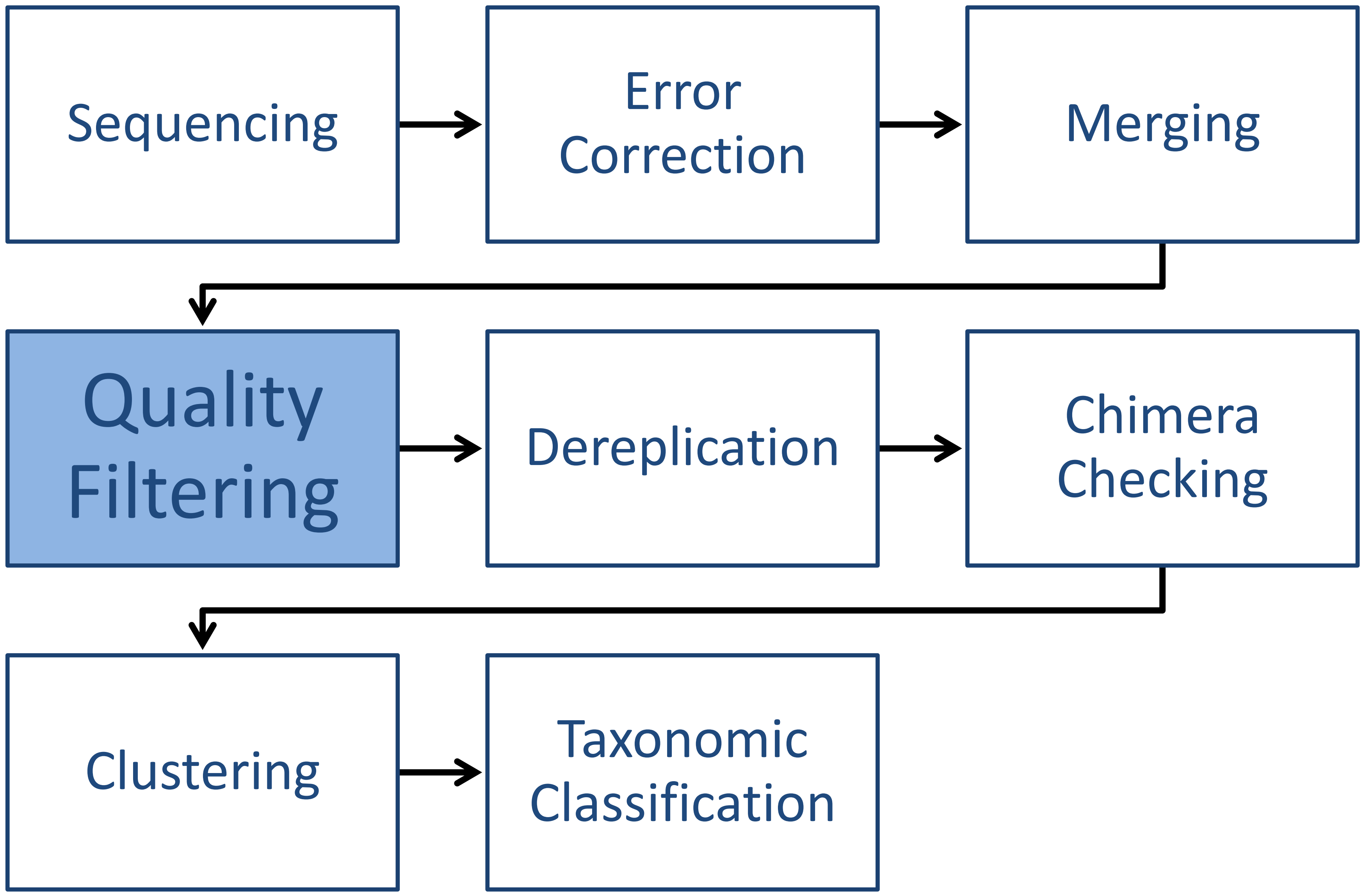
Paired End Read Merging

- Many algorithms to merge paired end reads:
  - FLASH, COPE, iTag, BIPES, Shera, PEAR, PandaSeq, FastqJoin, SeqPrep



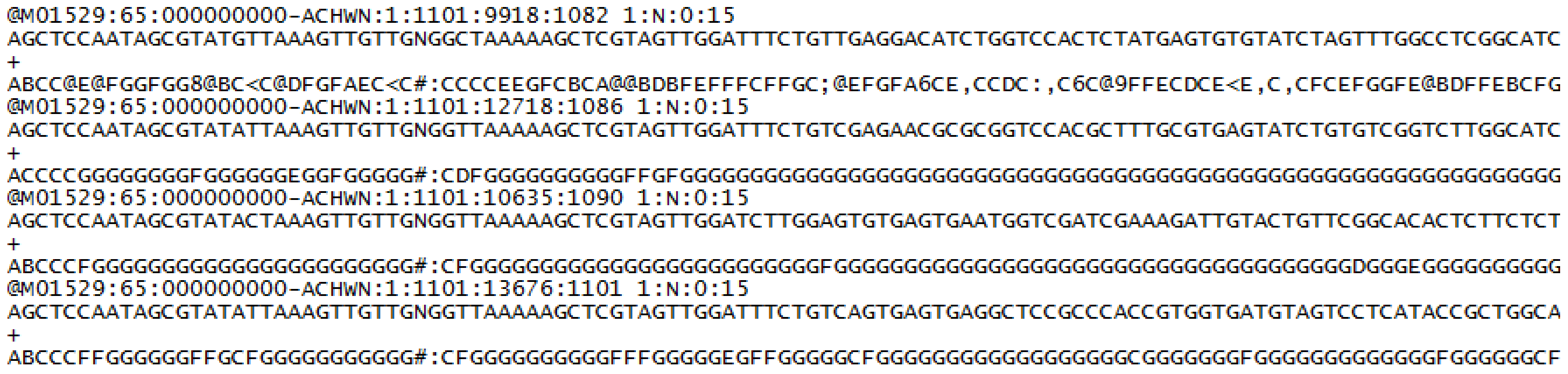
Paired End Merging

- Different approaches
  - Maximize overlap length to matches ratio (FLASH, COPE)
  - Maximize matches in overlap (fastq-join)
  - Maximize probability of true sequence match given quality scores and matches (PANDAseq)
  - Maximize assembly score of overlap (PEAR)


Sequencing
Error Correction
Merging
Quality Filtering
Dereplication
Chimera Checking
Clustering
Taxonomic Classification

Quality Filtering

- Quality filtering based on:
  - length,
  - mismatches to known sequences
  - ambiguous bases
  - PHRED quality scores



```
@M01529:65:000000000-ACHWN:1:1101:9918:1082 1:N:0:15
AGCTCCAATAGCGTATGTTAAAGTTGTTGNGGCTAAAAAGCTCGTAGTTGGATTTCTGTTGAGGACATCTGGTCCACTCTATGAGTGTGTATCTAGTTTGGCCTCGGCATC
+
ABCC@E@FGGFGG8@BC<C@DFGFAEC<C#:CCCCEEGFCBCA@@BDBFEFFFCFFGC;@EFGFA6CE,CCDC:,C6C@9FFECDCE<E,C,CFCEFGGFE@BDFFEBCFG
@M01529:65:000000000-ACHWN:1:1101:12718:1086 1:N:0:15
AGCTCCAATAGCGTATATTAAAGTTGTTGNGGTTAAAAAGCTCGTAGTTGGATTTCTGTCGAGAACGCGCGGTCCACGCTTTGCGTGAGTATCTGTGTCGGTCTTGGCATC
+
ACCCCGGGGGGGGFGGGGGGEGGFGGGGG#:CDFGGGGGGGGGGFFGFGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
@M01529:65:000000000-ACHWN:1:1101:10635:1090 1:N:0:15
AGCTCCAATAGCGTATACTAAAGTTGTTGNGGTTAAAAAGCTCGTAGTTGGATCTTGGAGTGTGAGTGAATGGTCGATCGAAAGATTGTACTGTTCGGCACACTCTTCTCT
+
ABCCCFGGGGGGGGGGGGGGGGGGGGGGG#:CFGGGGGGGGGGGGGGGGGGGGGGGGFGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGDGGGEGGGGGGGGGG
@M01529:65:000000000-ACHWN:1:1101:13676:1101 1:N:0:15
AGCTCCAATAGCGTATATTAAAGTTGTTGNGGTTAAAAAGCTCGTAGTTGGATTTCTGTCAGTGAGTGAGGCTCCGCCCACCGTGGTGATGTAGTCCTCATACCGCTGGCA
+
ABCCCFFGGGGGGFFGCFGGGGGGGGGGG#:CFGGGGGGGGGGFFFGGGGGEGFFGGGGGCFGGGGGGGGGGGGGGGGGGCGGGGGGGFGGGGGGGGGGGGGFGGGGGGCF
```

Quality Filtering

- Length filtering – uses *a priori* knowledge about your amplicons
- Ambiguous bases – often reflect incomplete fluorophore flushing in the sequencing cycle or mismatches from the read merging step
- Mismatches to known sequences (primers, tags) can be used as an overall indication of read quality

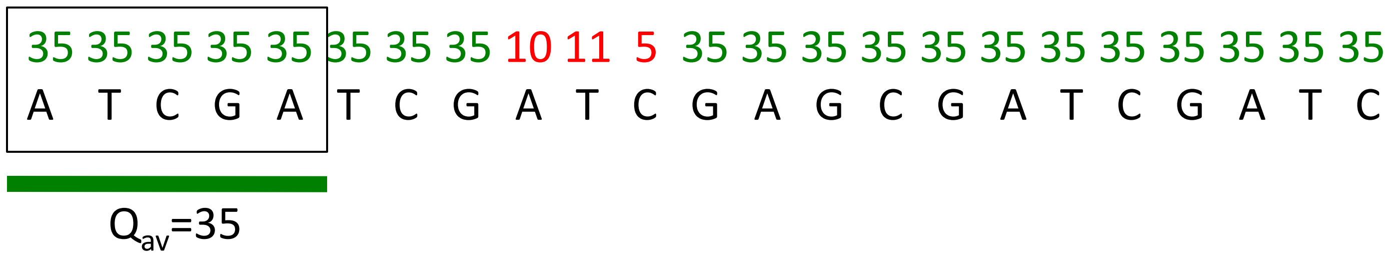
Quality Filtering: PHRED Quality Scores

$$Q = -10 \log_{10} P$$

- Quality scores are calculated based on the probability of a basecall error
- Limited use for specific error detection because many substitution errors have high quality scores
- Useful for pairing, contig-based error correction, identifying poor sequencing

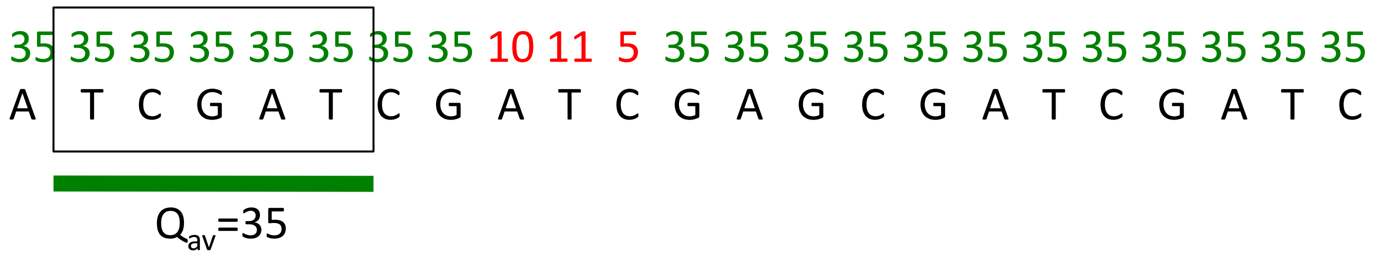
Quality Filtering: PHRED Quality Scores

- Filtering based on:
  - Minimum quality score
  - Average quality score
  - Truncation from poor quality score
  - Average quality across a sliding window
- If filtering after merging, it’s important to consider how the quality scores for the overlapping region were calculated



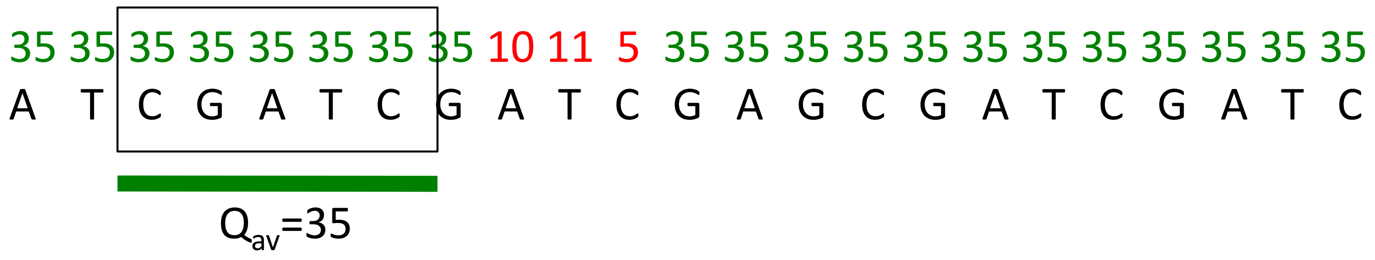
Quality Filtering: PHRED Quality Scores

- Filtering based on:
    - Minimum quality score
    - Average quality score
    - Truncation from poor quality score
    - Average quality across a sliding window
- If filtering after merging, it’s important to consider how the quality scores for the overlapping region were calculated



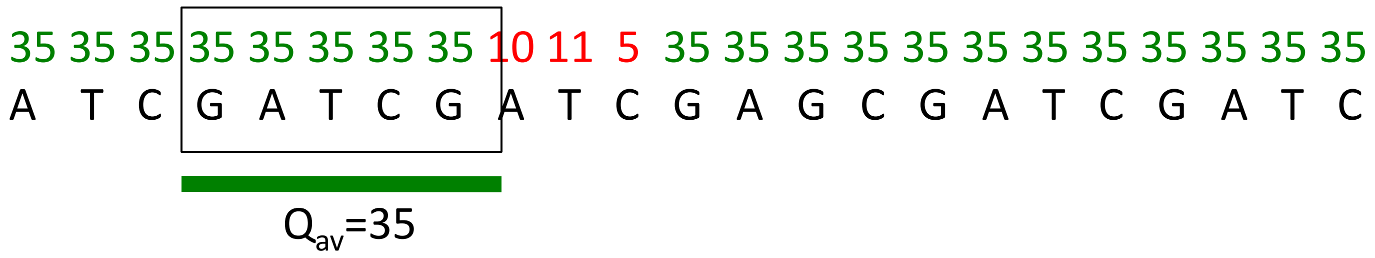
Quality Filtering: PHRED Quality Scores

- Filtering based on:
  - Minimum quality score
  - Average quality score
  - Truncation from poor quality score
  - Average quality across a sliding window
- If filtering after merging, it’s important to consider how the quality scores for the overlapping region were calculated



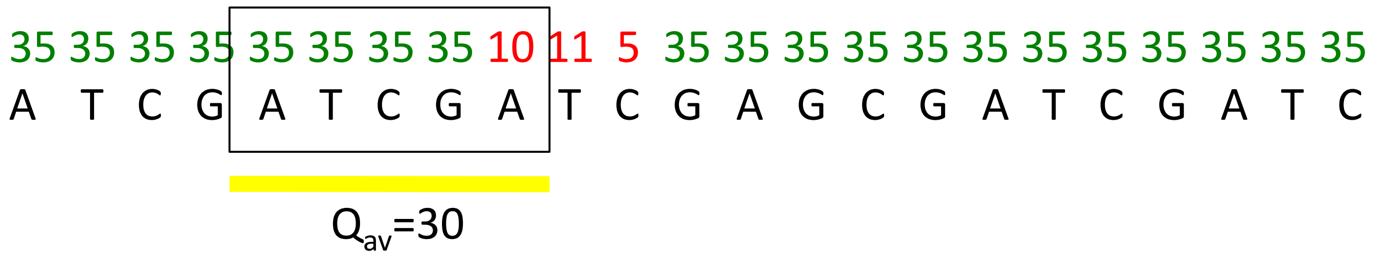
Quality Filtering: PHRED Quality Scores

- Filtering based on:
  - Minimum quality score
  - Average quality score
  - Truncation from poor quality score
  - Average quality across a sliding window
- If filtering after merging, it's important to consider how the quality scores for the overlapping region were calculated



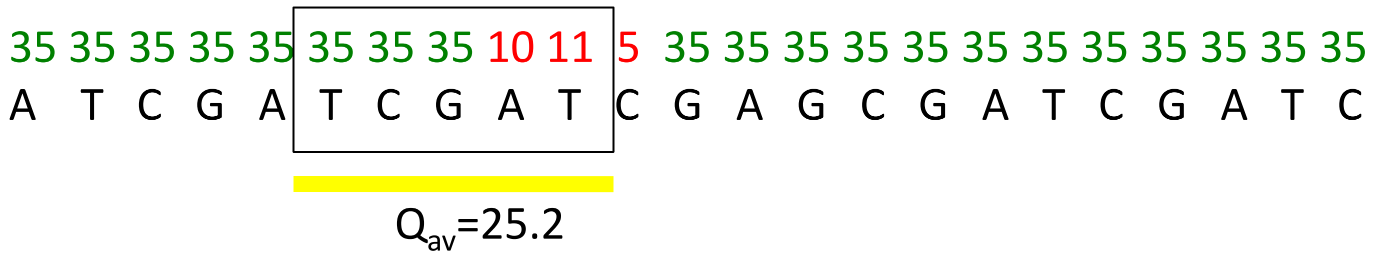
Quality Filtering: PHRED Quality Scores

- Filtering based on:
  - Minimum quality score
  - Average quality score
  - Truncation from poor quality score
  - Average quality across a sliding window
- If filtering after merging, it’s important to consider how the quality scores for the overlapping region were calculated



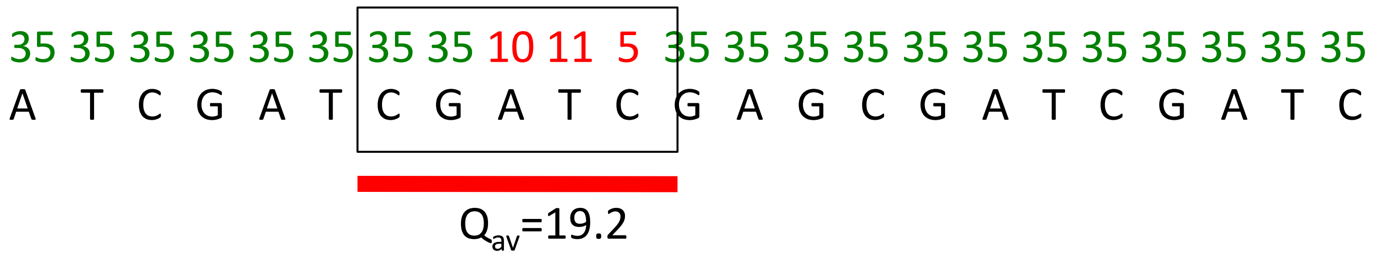
Quality Filtering: PHRED Quality Scores

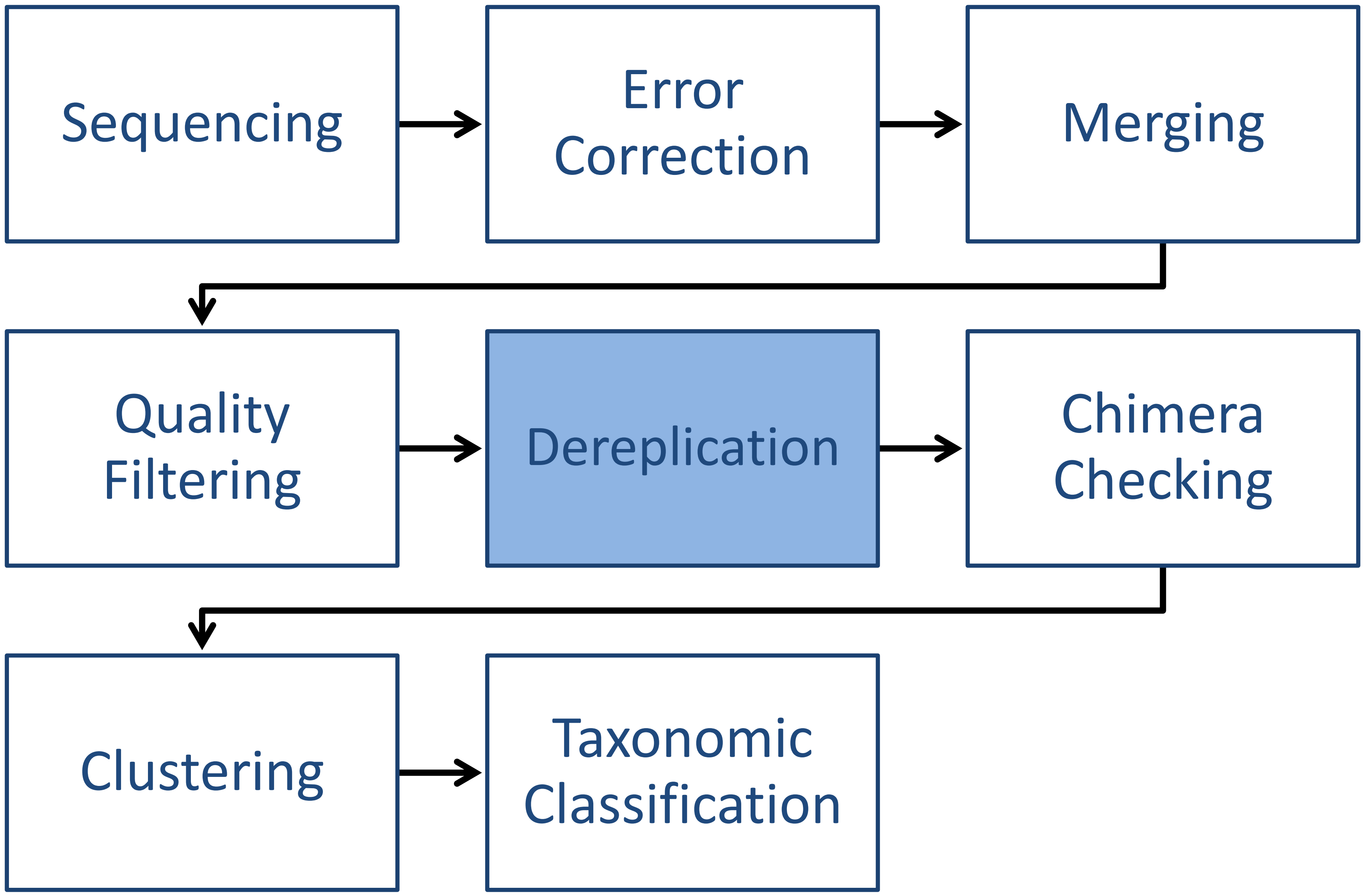
- Filtering based on:
  - Minimum quality score
  - Average quality score
  - Truncation from poor quality score
  - Average quality across a sliding window
- If filtering after merging, it’s important to consider how the quality scores for the overlapping region were calculated



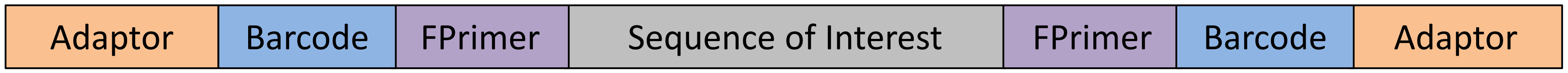
Quality Filtering: PHRED Quality Scores

- Filtering based on:
  - Minimum quality score
  - Average quality score
  - Truncation from poor quality score
  - Average quality across a sliding window
- If filtering after merging, it’s important to consider how the quality scores for the overlapping region were calculated




Sequencing
Error Correction
Merging
Quality Filtering
Dereplication
Chimera Checking
Clustering
Taxonomic Classification

Dereplication



- Separate and label reads based on known unique ’tags’ that are incorporated into the amplicons for each sample
- Sometimes done at the sequencing facility
- Algorithms search for barcode sequences and bin sequences accordingly
  - Most algorithms include a mismatch parameter to maximize assignment despite sequencing errors

Dereplication

- Most dereplication algorithms include a mismatch parameter to maximize assignment despite sequencing errors
  - Carefully consider error-correcting power of barcodes before allowing mismatches!
  - Mismatch parameter MUST be < barcode error correction

Barcodes with 1 bp Error Correction
Mismatch =1

Tag 1 ATCATC
Tag 2 $\boxed{\text{ATCATT}}$
Tag 3 ATCGTT

$\boxed{\text{ATCATT}}$GGCCTTAATTCCGG

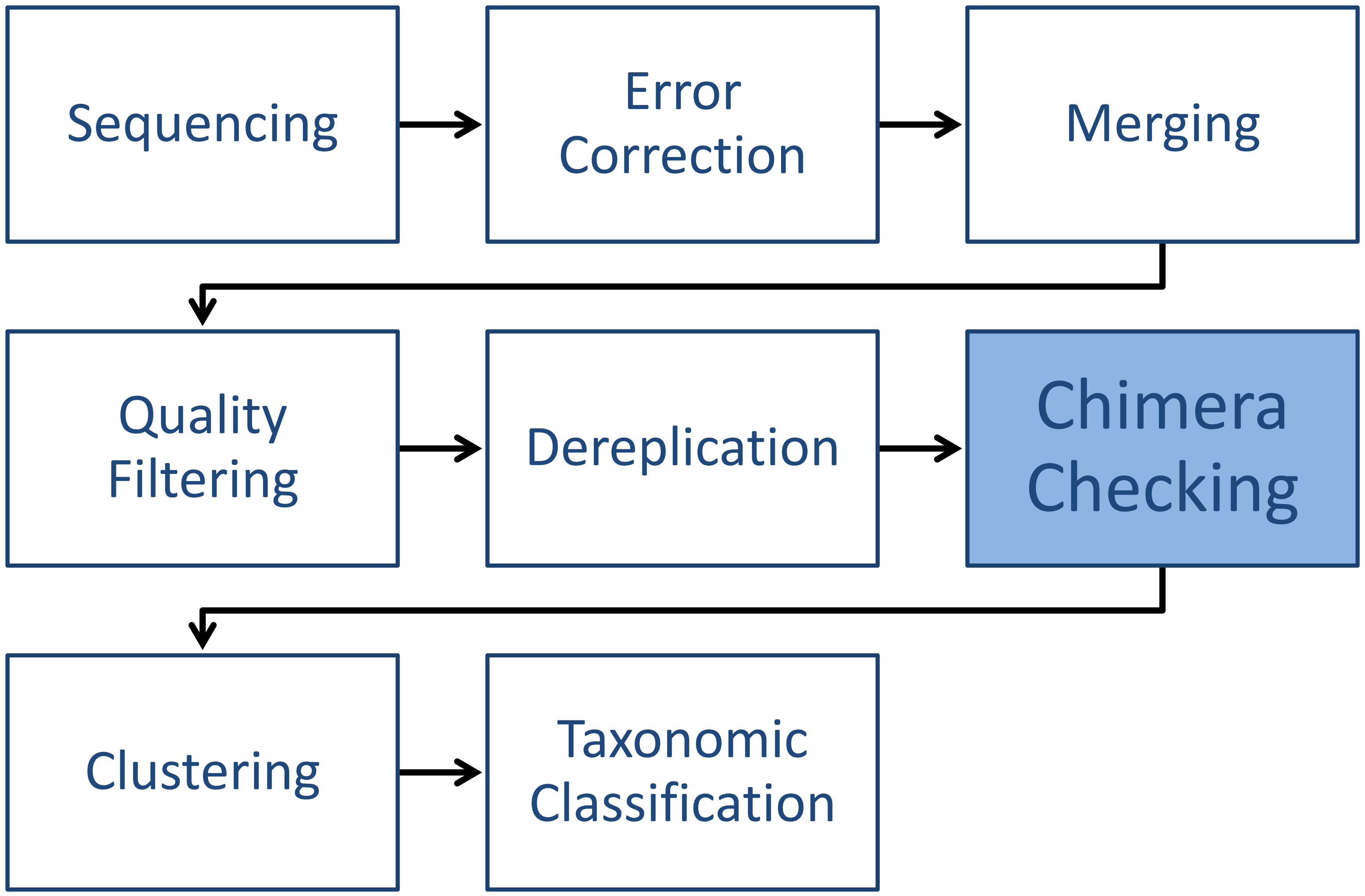
Incorrectly binned as Tag 1

Barcodes with 2 bp Error Correction
Mismatch =1

Tag 1 ATCCTC
Tag 2 $\boxed{\text{ATCATT}}$
Tag 3 ATCGTG

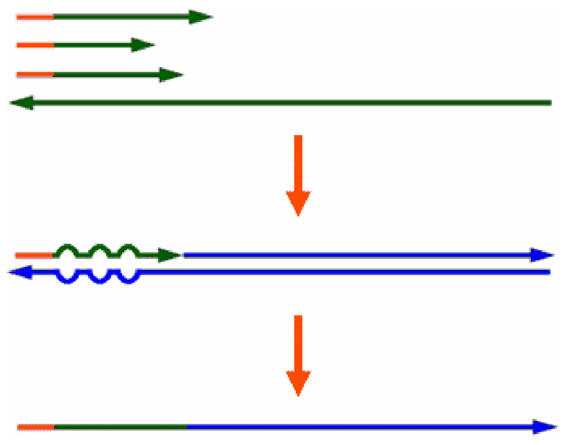
$\boxed{\text{ATCATT}}$GGCCTTAATTCCGG

Correctly binned as Tag 2


Sequencing
Error Correction
Merging
Quality Filtering
Dereplication
Chimera Checking
Clustering
Taxonomic Classification

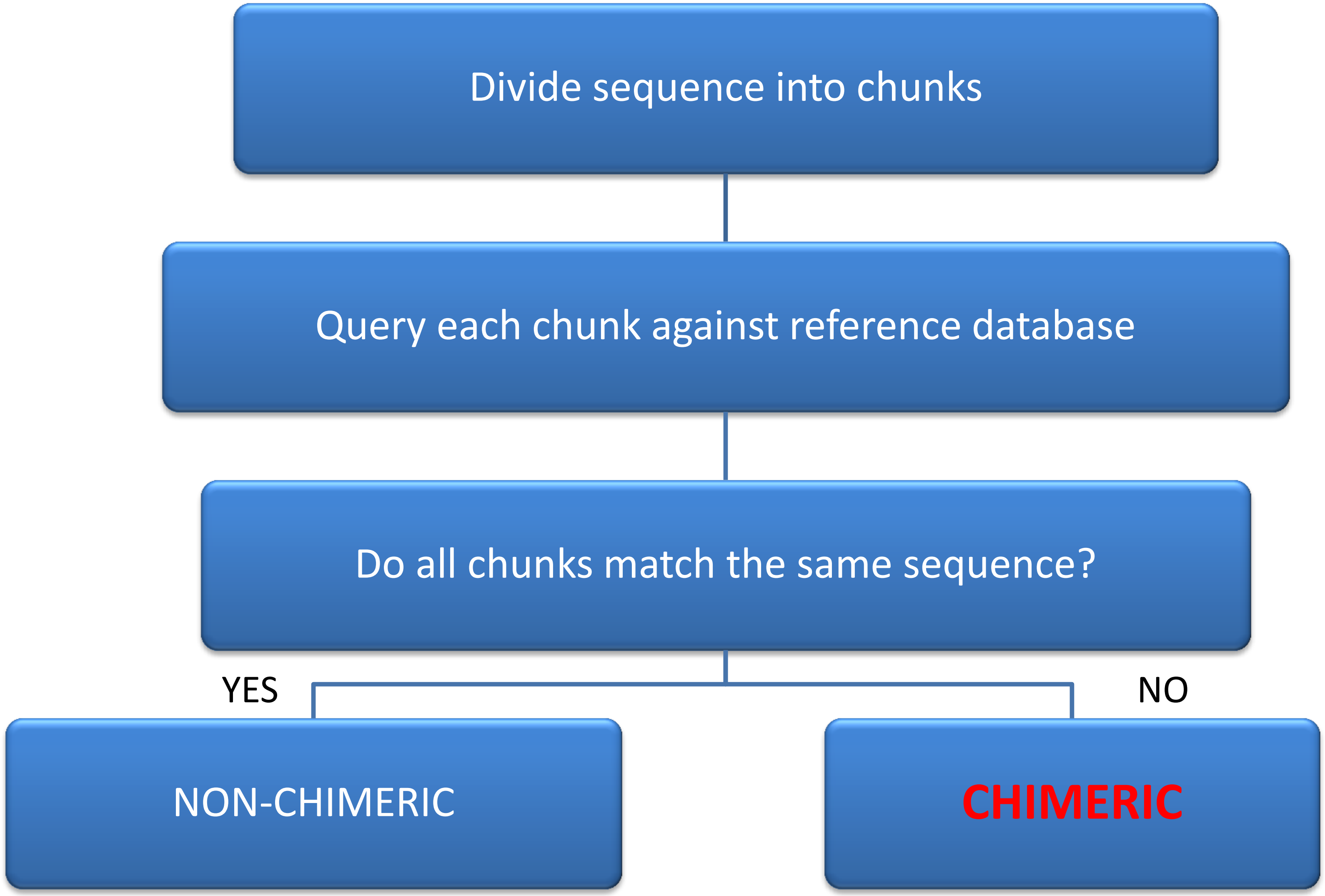
Chimera Detection

- Chimeras are formed during PCR when aborted fragments act as primers during the next cycle
- In environmental sequencing this leads to inflated estimates of diversity

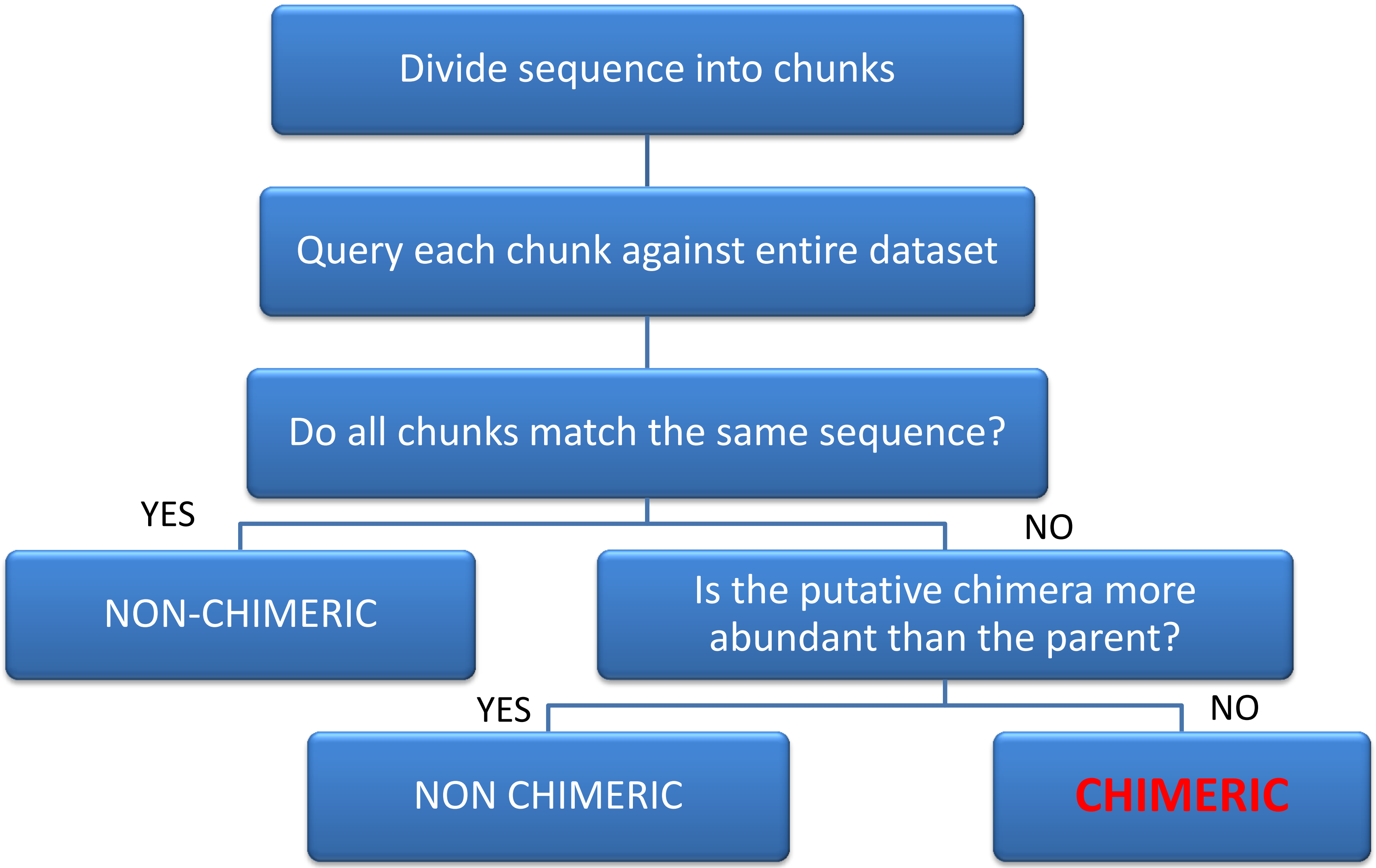


- Chimera detection can be performed in two ways
    - reference based (using a known set of non-chimeric sequences as putative parents)
    - *de novo* based (probabilistic method using sequence similarity and abundances to identify putative chimeras)

Reference-Based Chimera detection

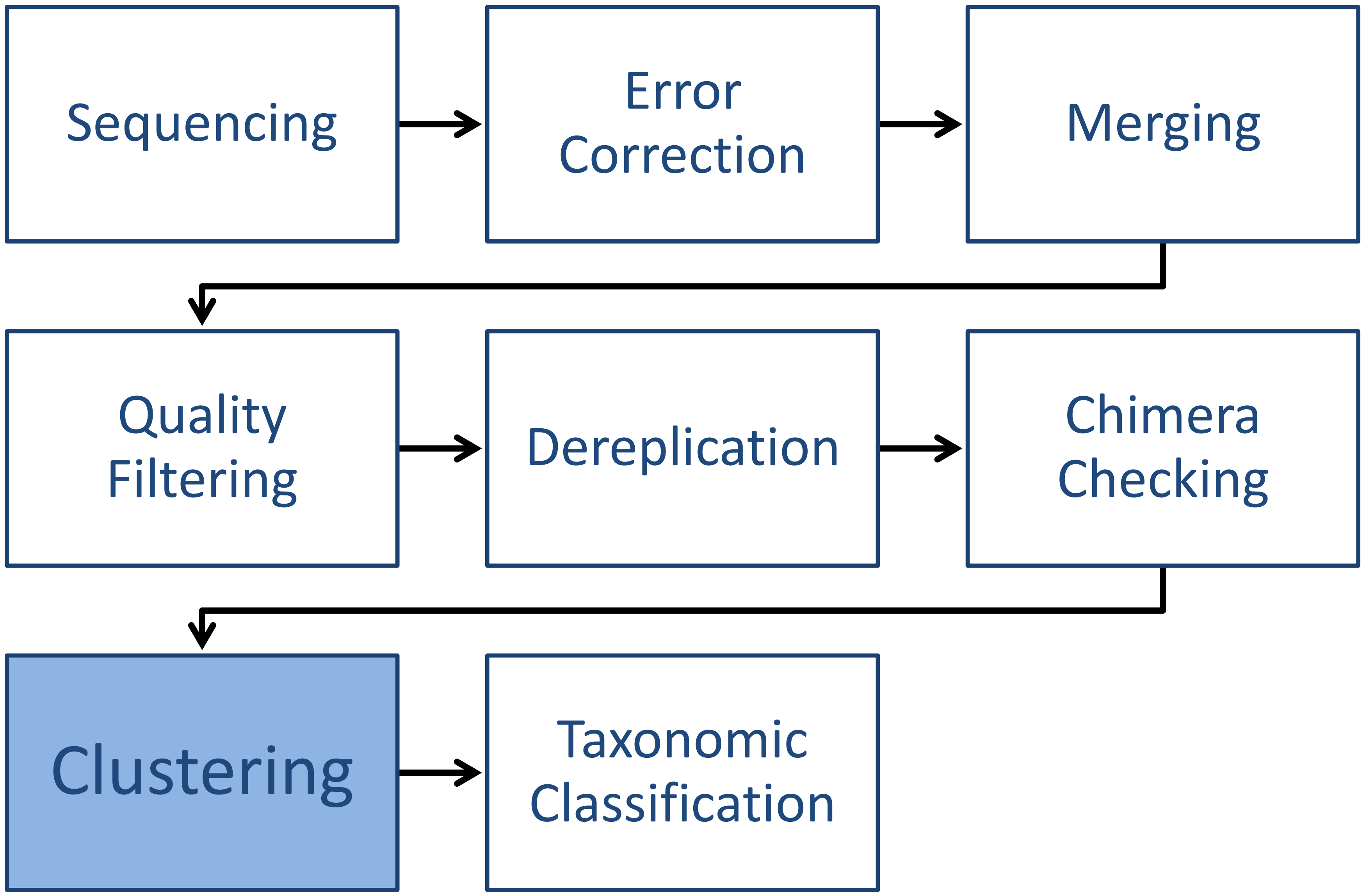

Divide sequence into chunks
Query each chunk against reference database
Do all chunks match the same sequence?
YES
NO
NON-CHIMERIC
CHIMERIC

De-Novo Chimera detection


Divide sequence into chunks
Query each chunk against entire dataset
Do all chunks match the same sequence?
YES
NO
NON-CHIMERIC
Is the putative chimera more abundant than the parent?
YES
NO
NON CHIMERIC
CHIMERIC

Chimera Detection

Program	Type	Method	Alignment Required
Bellerophon	de novo	partial treeing analysis allows detection of chimeric branching patterns	yes
c-code	reference based	aligned pairwise identity of query fragments versus reference sequences	no
pintail	reference based		yes
perseus	De novo	Fragments a query sequence and uses abundance/similarity to evaluate likelihood of other sequences in dataset as chimeric parents	no
ChimeraSlayer		30% of sequence from beginning and end are used to identify putative parents from a database, and chimeras are identified as having greater sequence homology than an in silico chimera generated from the putative parents	yes
Uchime	De novo or reference based	Query is divided into fragments that are searched against a reference database, the more abundant sequences in the dataset, and scores are assigned to a 3-way alignment of parents and query	no
usearch61	De novo or reference based	Similar to uchime	no
Blast_ fragments	Reference based	Queries are fragmented and compared to a reference database using the BLAST algorithm	no

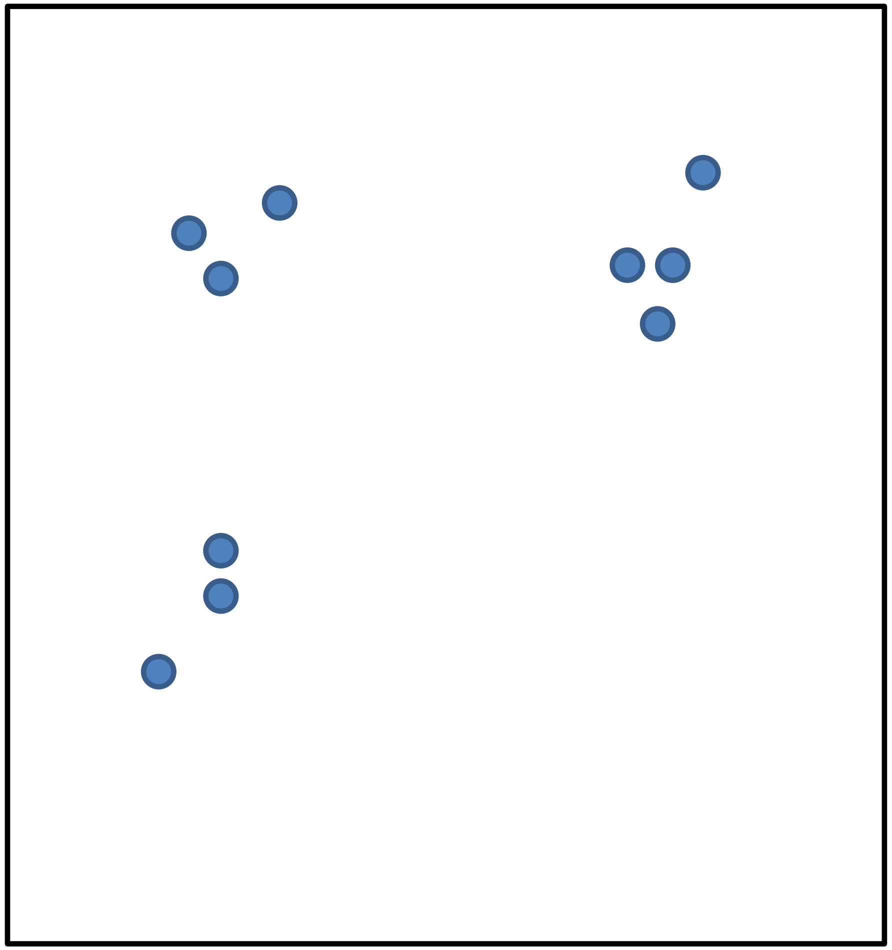

Sequencing
Error Correction
Merging
Quality Filtering
Dereplication
Chimera Checking
Clustering
Taxonomic Classification

Clustering

- Intraspecific genetic variation and sequencing errors may mean that unique sequence variants do not represent meaningful biological units
- Clustering sets an arbitrary threshold for grouping sequences into ‘species’
- Primary methods are heuristic clustering and hierarchical clustering

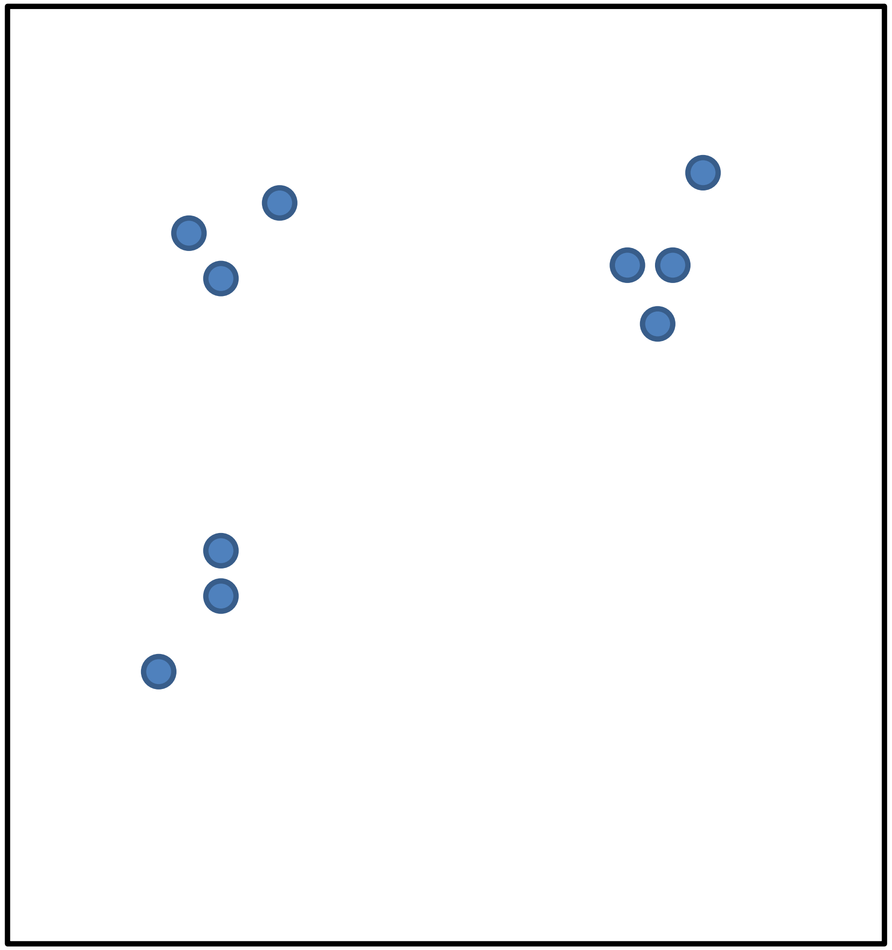
Hierarchical Clustering

- Begins by considering all unique sequences to be individual OTUs
  - Iteratively merges the 2 most similar OTUs into one, so long as the distance between those OTUs is within a user-set threshold



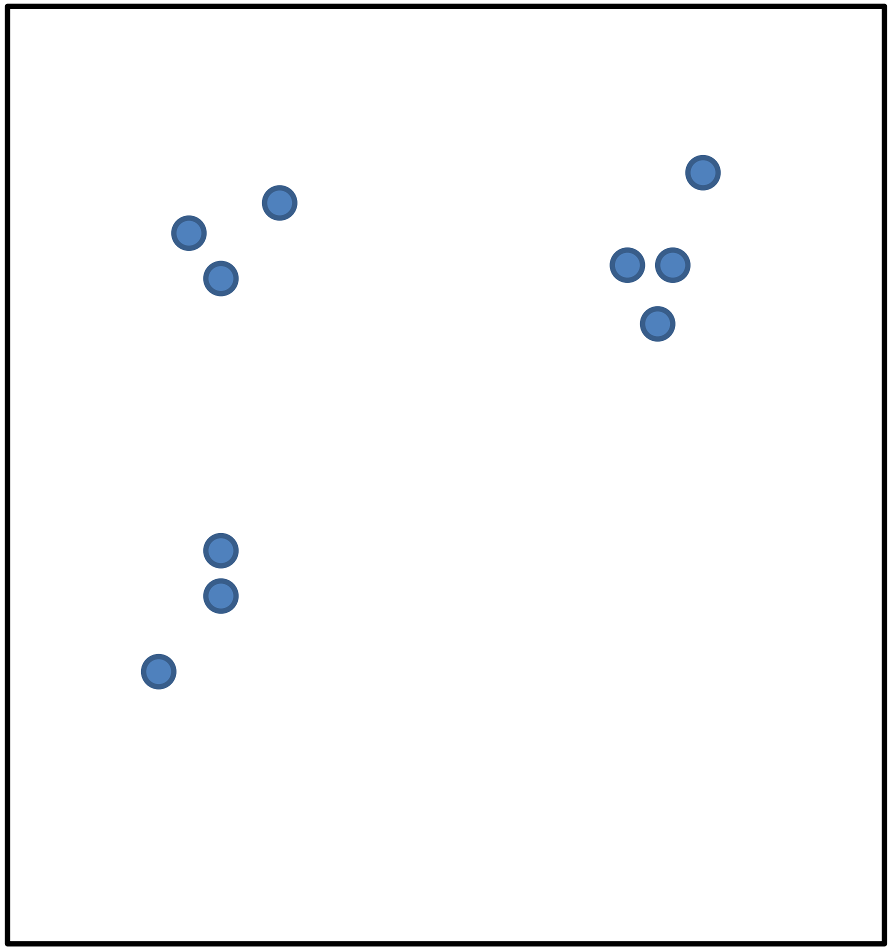
Hierarchical Clustering

- Begins by considering all unique sequences to be individual OTUs
  - Iteratively merges the 2 most similar OTUs into one, so long as the distance between those OTUs is within a user-set threshold



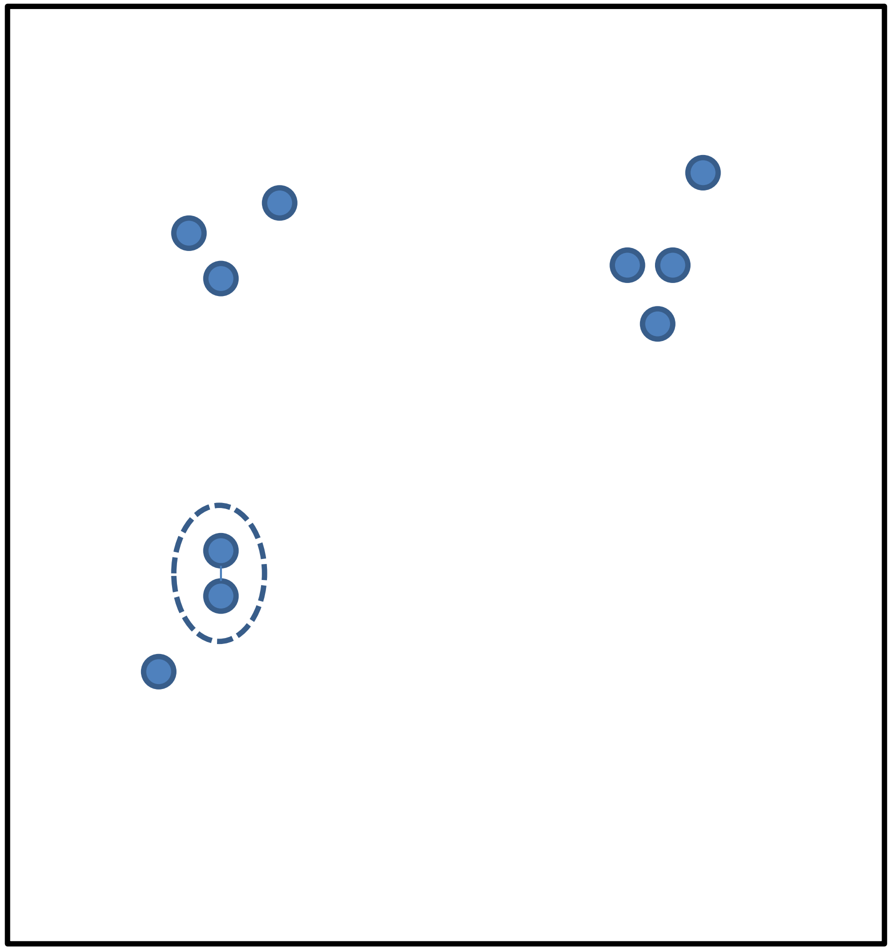
Hierarchical Clustering

- Begins by considering all unique sequences to be individual OTUs
  - Iteratively merges the 2 most similar OTUs into one, so long as the distance between those OTUs is within a user-set threshold



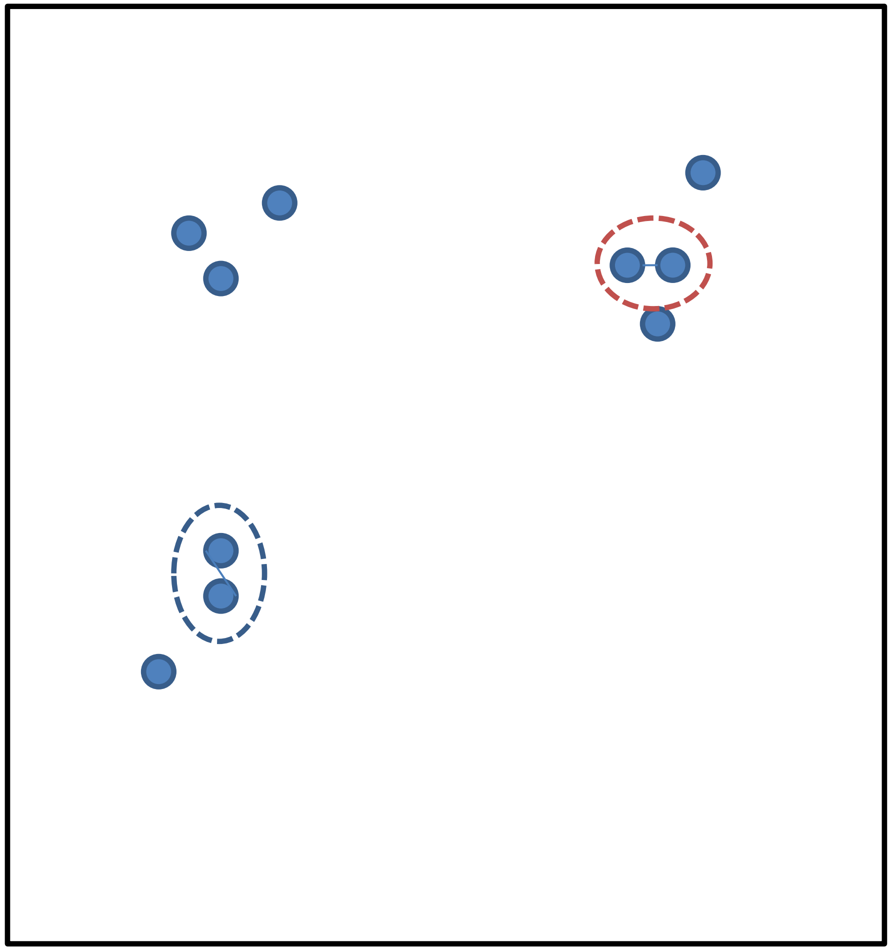
Hierarchical Clustering

- Begins by considering all unique sequences to be individual OTUs
  - Iteratively merges the 2 most similar OTUs into one, so long as the distance between those OTUs is within a user-set threshold



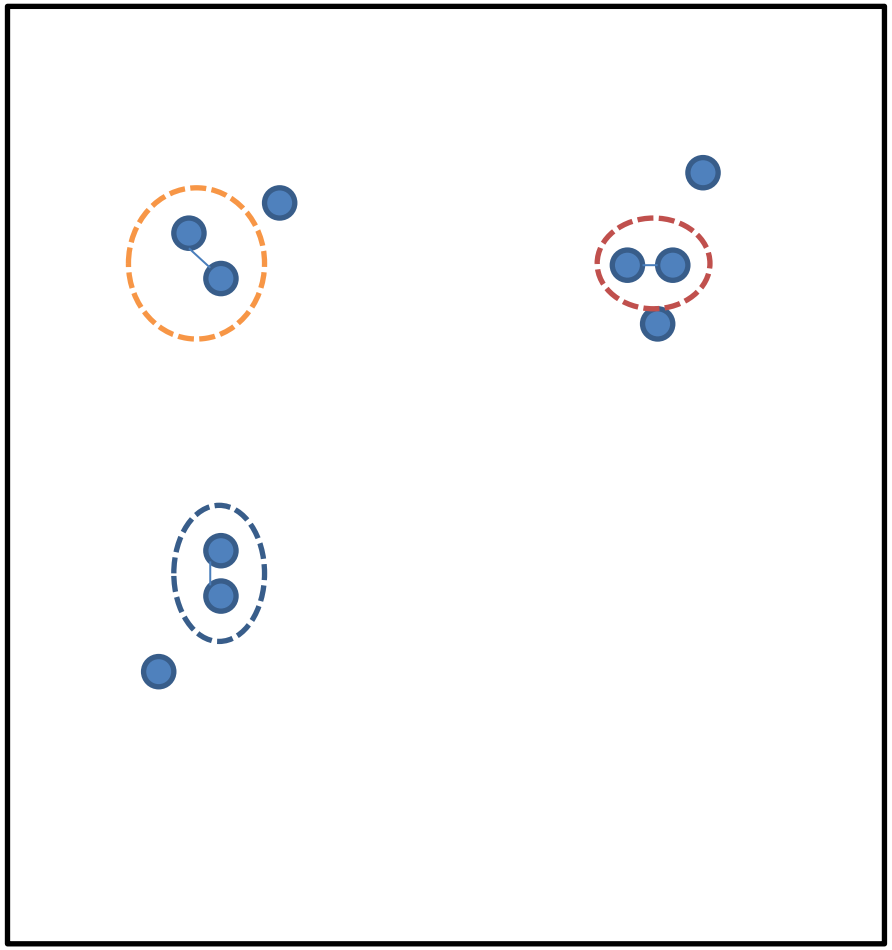
Hierarchical Clustering

- Begins by considering all unique sequences to be individual OTUs
  - Iteratively merges the 2 most similar OTUs into one, so long as the distance between those OTUs is within a user-set threshold



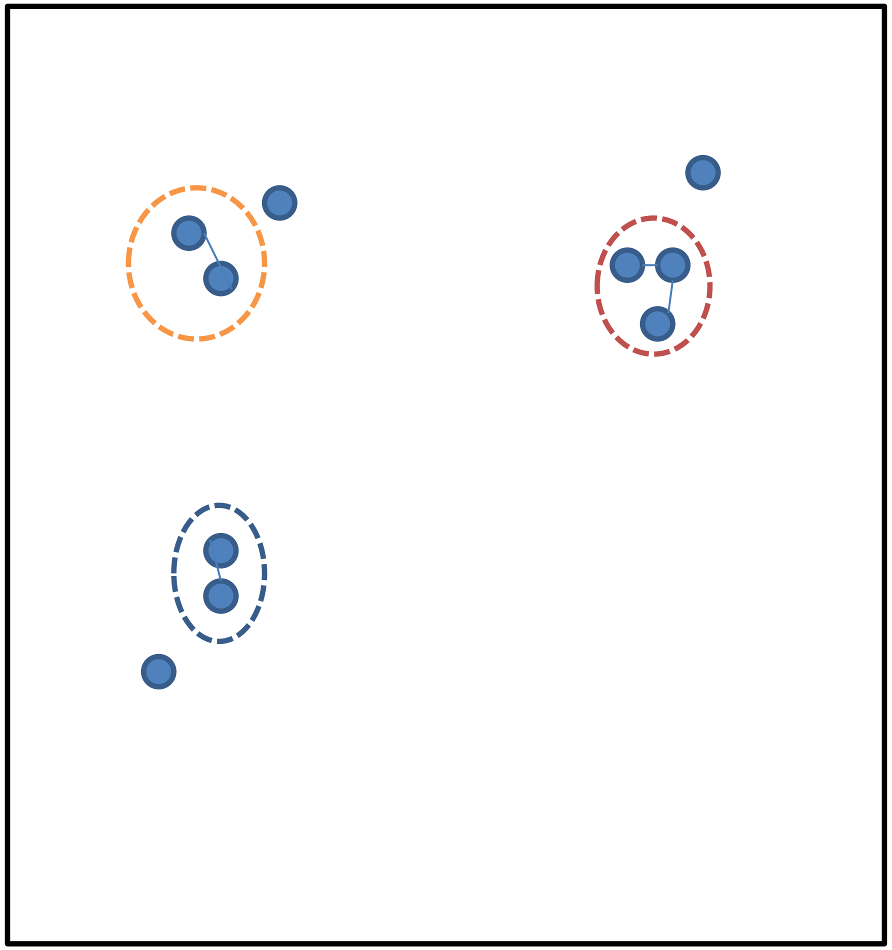
Hierarchical Clustering

- Begins by considering all unique sequences to be individual OTUs
  - Iteratively merges the 2 most similar OTUs into one, so long as the distance between those OTUs is within a user-set threshold



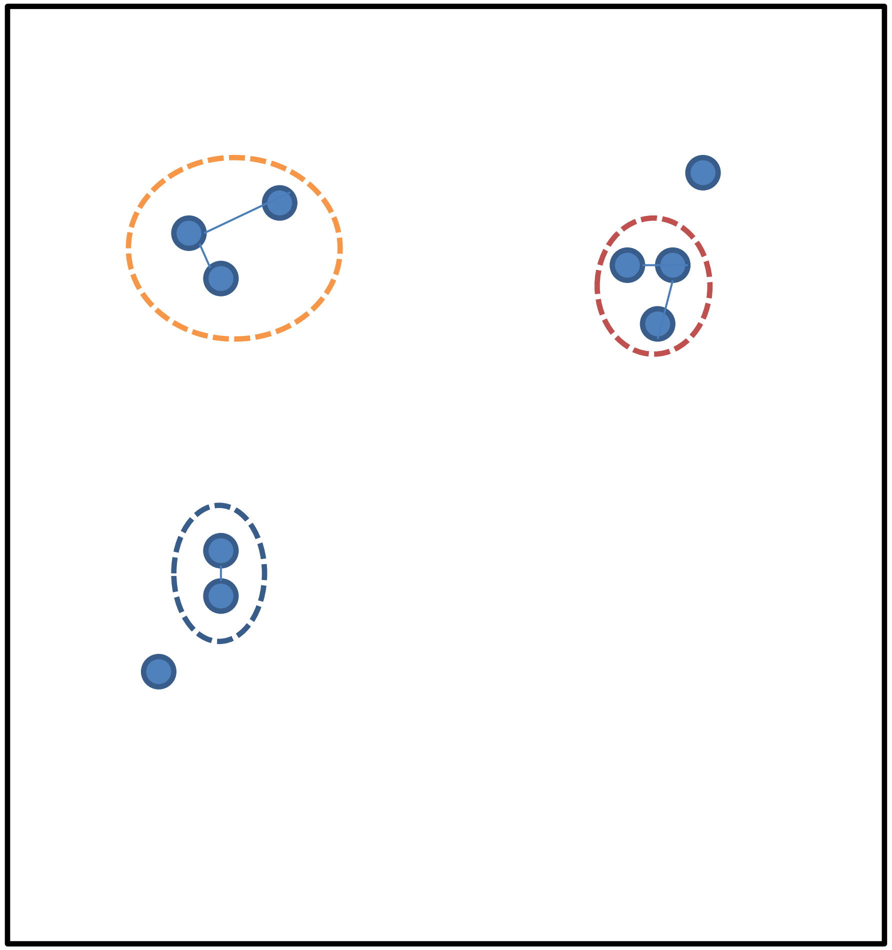
Hierarchical Clustering

- Begins by considering all unique sequences to be individual OTUs
  - Iteratively merges the 2 most similar OTUs into one, so long as the distance between those OTUs is within a user-set threshold



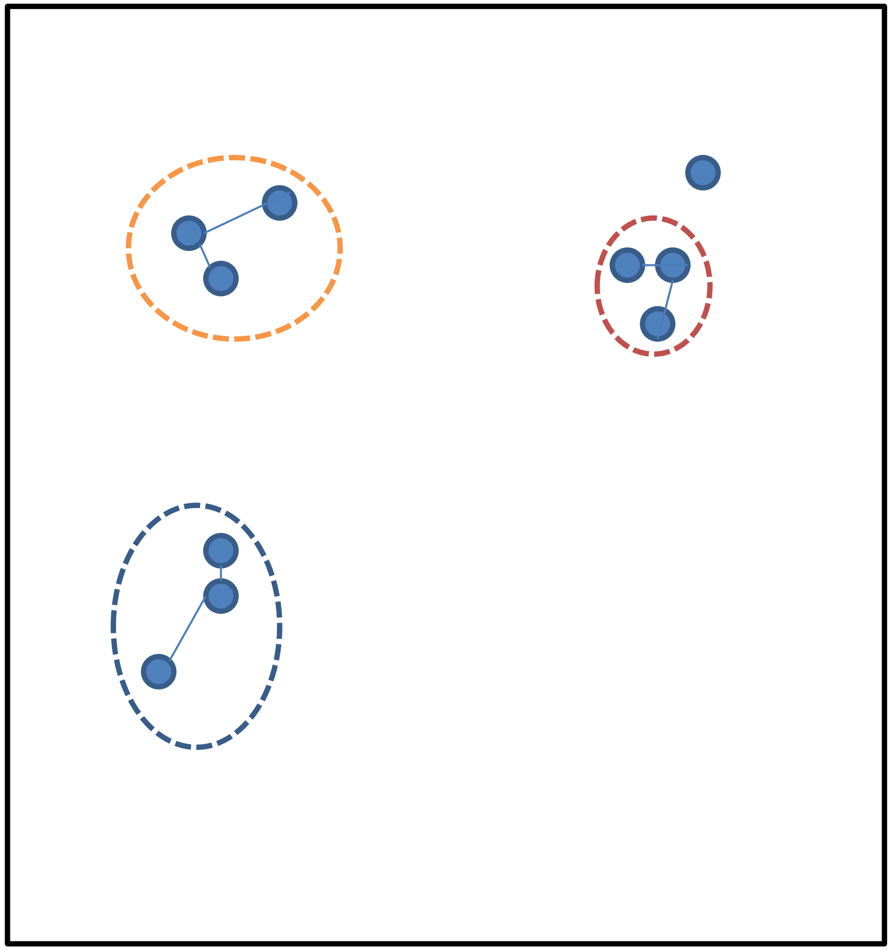
Hierarchical Clustering

- Begins by considering all unique sequences to be individual OTUs
  - Iteratively merges the 2 most similar OTUs into one, so long as the distance between those OTUs is within a user-set threshold



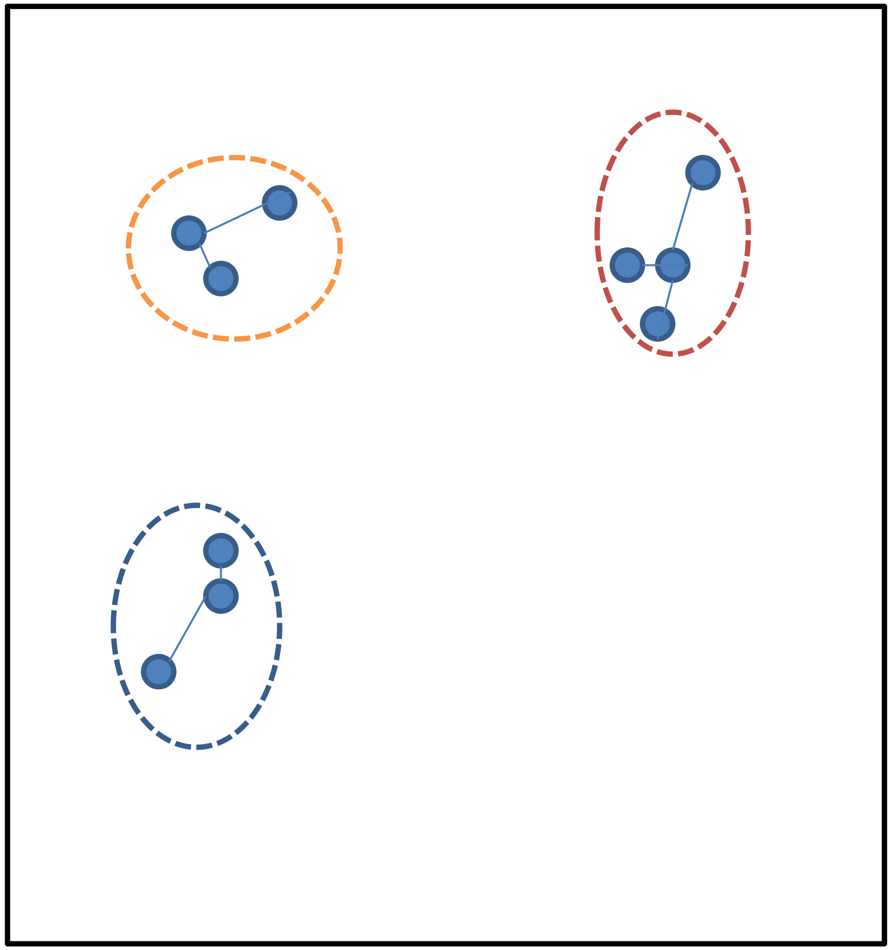
Hierarchical Clustering

- Begins by considering all unique sequences to be individual OTUs
  - Iteratively merges the 2 most similar OTUs into one, so long as the distance between those OTUs is within a user-set threshold



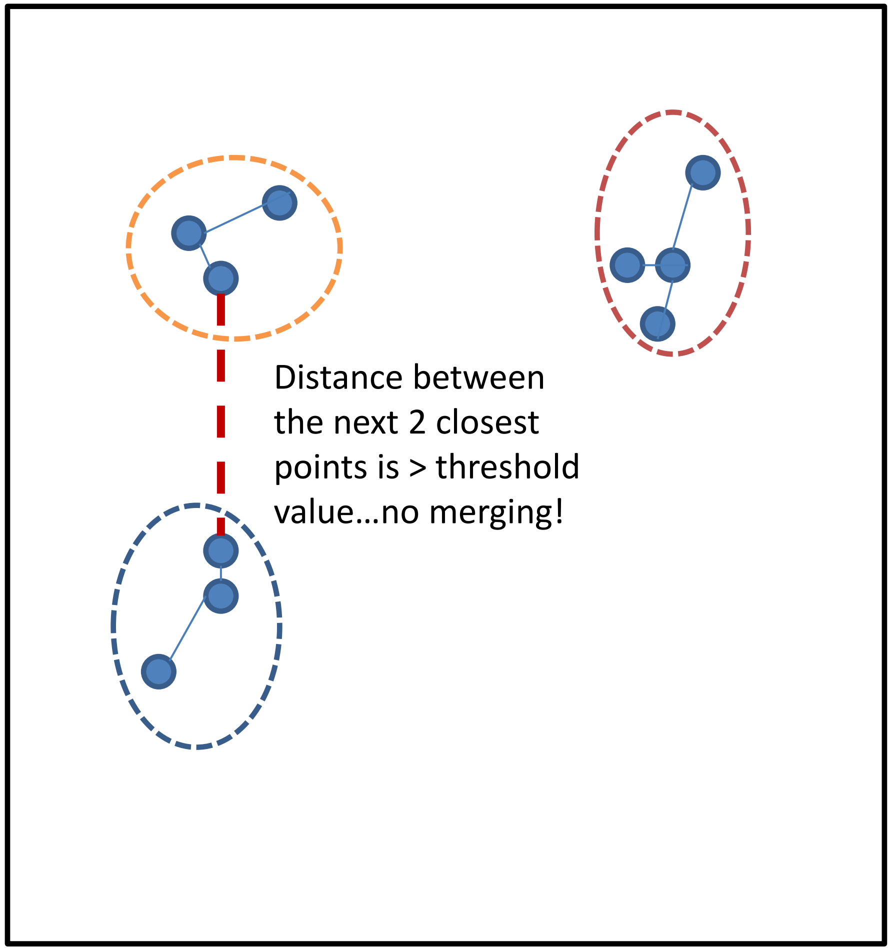
Hierarchical Clustering

- Begins by considering all unique sequences to be individual OTUs
  - Iteratively merges the 2 most similar OTUs into one, so long as the distance between those OTUs is within a user-set threshold



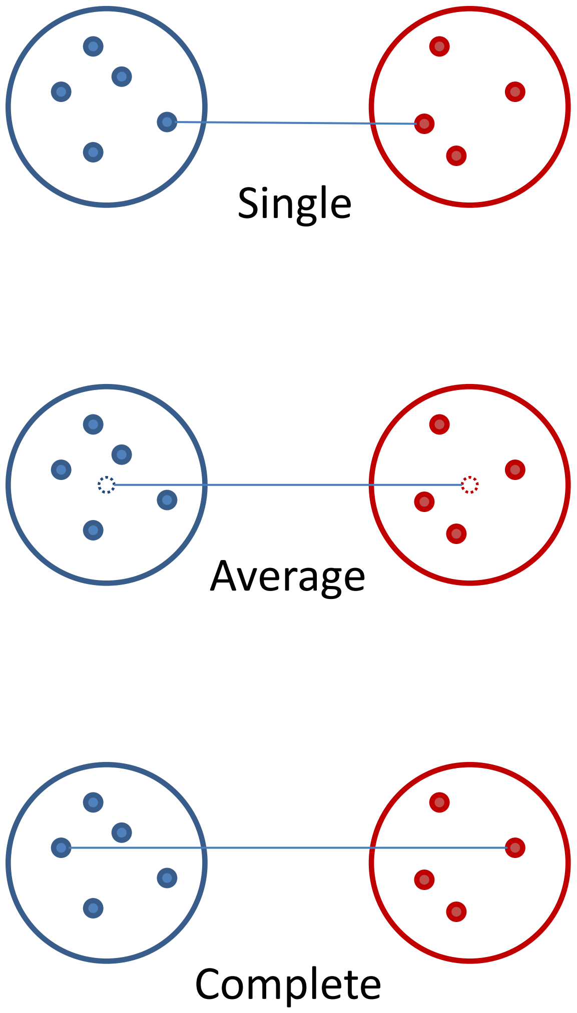
Hierarchical Clustering

- Begins by considering all unique sequences to be individual OTUs
  - Iteratively merges the 2 most similar OTUs into one, so long as the distance between those OTUs is within a user-set threshold


Distance between the next 2 closest points is > threshold value…no merging!

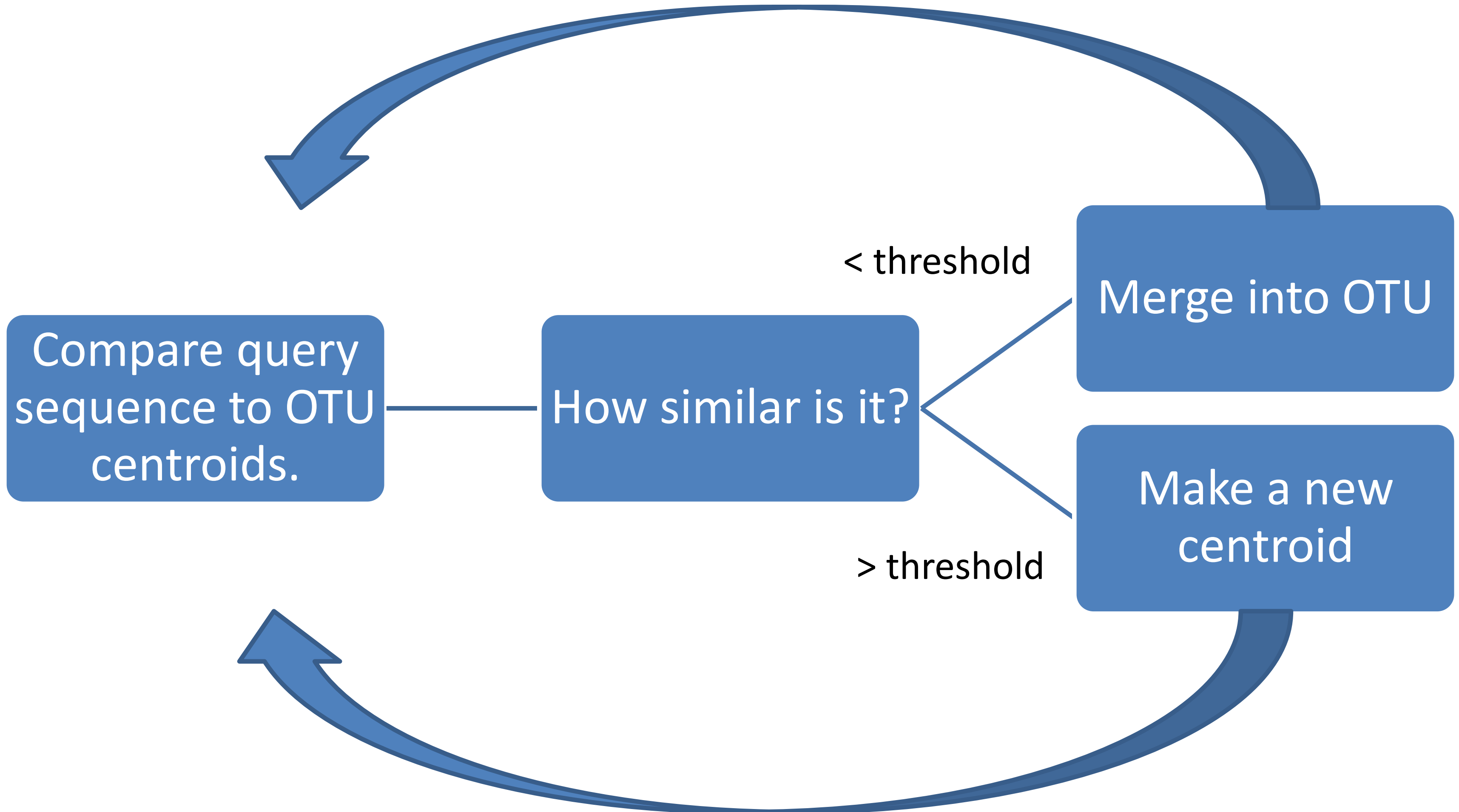
Hierarchical Clustering

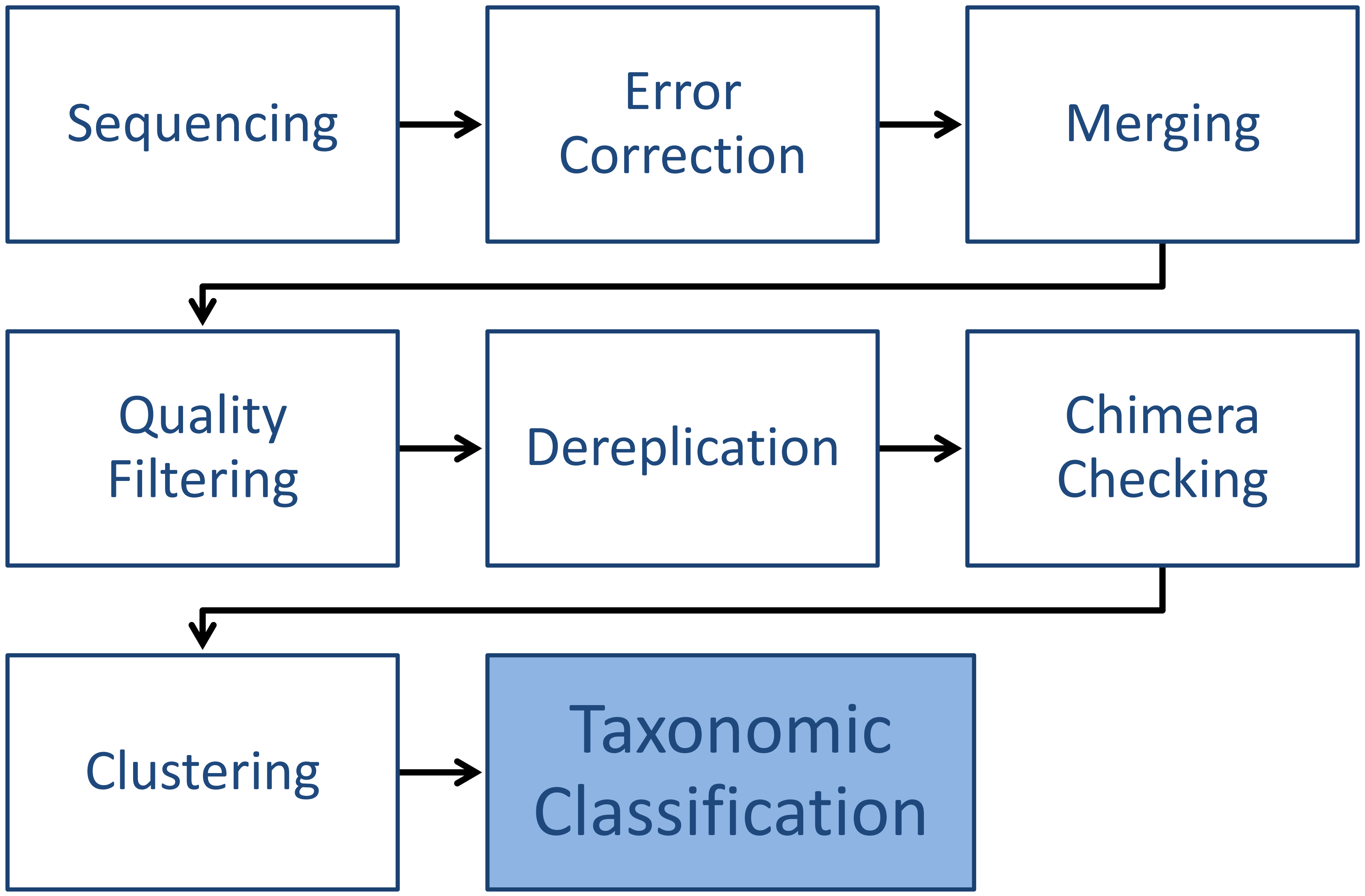
- Sub-types of hierarchical clustering are based on how the distance between OTUs is measured
  - Single linkage (distance between the point and the closest member of the OTU
  - Average linkage(distance between the point and the middle (centroid) of the OTU
  - Complete linkage (distance between the point and the farthest member of the OTU
- Hierarchical clustering is implemented in mothur, qiime, and swarm



Heuristic Clustering

- Includes algorithms like cd-hit, usearch, uclust, sumaclust
- Based on pairwise comparisons between sequences




Sequencing
Error Correction
Merging
Quality Filtering
Dereplication
Chimera Checking
Clustering
Taxonomic Classification

Taxonomic Classification

- OTU representative sequences are classified against sequence databases (ex/ GenBank, greengenes, SILVA, UNITE)
  - Primary approaches are BLAST and the RDP Bayesian Classifier
  - Results will only ever be as good as your database