Lecture course on environmental DNA metabarcoding using Claident and R: From nucleotide sequence data processing to ecological analyses

Akifumi S. Tanabe

田辺晶史

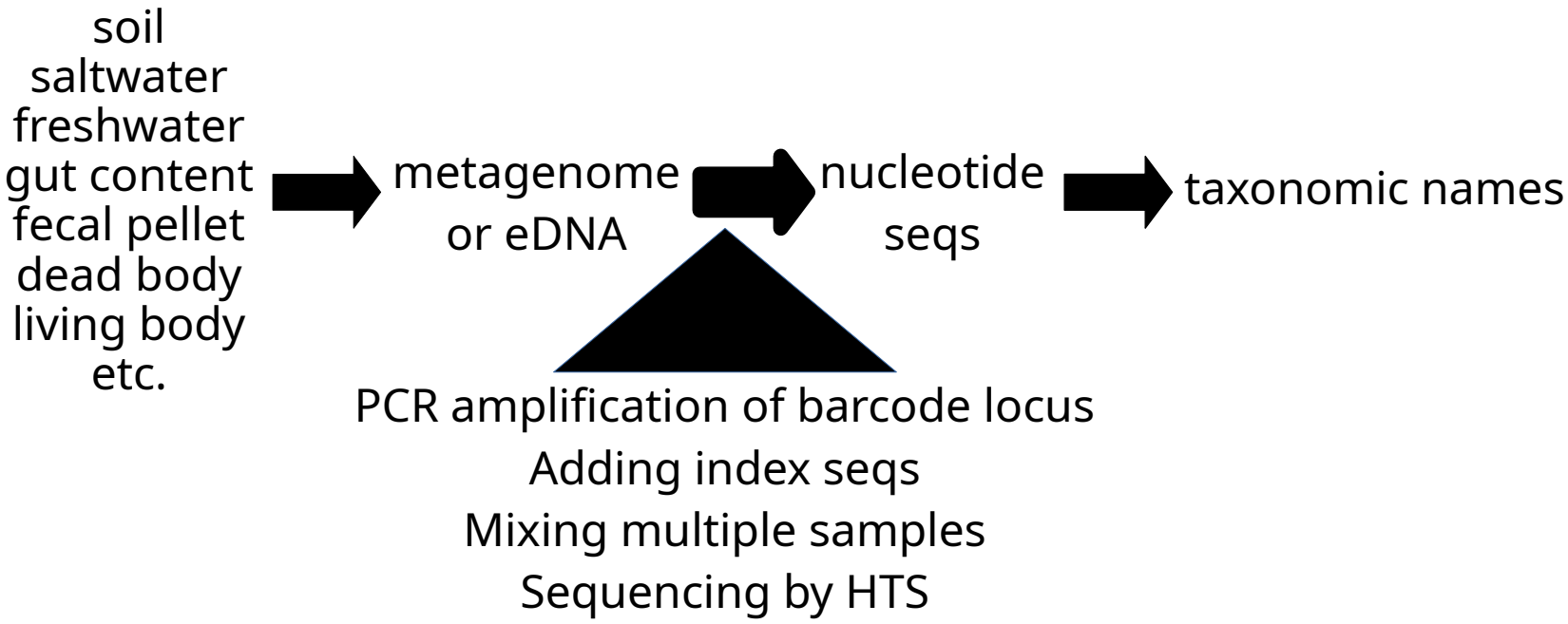
ClaidentとRによる環境DNAメタバーコーディング分析講座：塩基配列データ処理から生態学的分析まで

Workflow of metabarcoding

soil
saltwater
freshwater
gut content
fecal pellet
dead body
living body
etc.

→ metagenome or eDNA → nucleotide seqs → taxonomic names

Molecular laboratory processes of metabarcoding



Computational processes of metabarcoding

soil
saltwater
freshwater
gut content
fecal pellet
dead body
living body
etc.

→ metagenome or eDNA → nucleotide seqs → taxonomic names

Demultiplexing
Quality-trimming
Quality-filtering
Denoising
Chimera removal
Decontamination
Clustering
Taxonomic assignment

Claident

https://www.claident.org/

Single-end sequence data analysis in Claident

1. Demultiplexing by clsplitseq
2. Evaluate sequence quality by VSEARCH via clcalcfastqstatv
3. Quality-trimming&filtering by VSEARCH via clfilterseqv
4. Denoising by DADA2 via cldenoiseseqd
5. Removing chimeras by UCHIME3 via clremovechimev
6. Removing contaminants by clremovecontam
7. Additional clustering by VSEARCH via clclassseqv (Optional)
8. Assigning taxonomy by clmakecachedb,clidentseq,classigntax
9. Additional taxonomy processing by clmergeassign,clfillassign
10. Summarizing results by clsumclass,clsumtaxa

Overlapped paired-end sequence data analysis in Claident

1. Demultiplexing by clsplitseq
2. Concatenating pairs by VSEARCH via clconcatpairv
3. Quality-filtering by VSEARCH via clfilterseqv
4. Denoising by DADA2 via cldenoiseseqd
5. Removing chimeras by UCHIME3 via clremovechimev
6. Removing contaminants by clremovecontam
7. Additional clustering by VSEARCH via clclassseqv (Optional)
8. Assigning taxonomy by clmakecachedb,clidentseq,classigntax
9. Additional taxonomy processing by clmergeassign,clfillassign
10. Summarizing results by clsumclass,clsumtaxa

Non-overlapped paired-end sequence data analysis in Claident

1. Demultiplexing by clsplitseq
2. Evaluate sequence quality by VSEARCH via clcalcfastqstatv x2
3. Quality-filtering by VSEARCH via clfilterseqv x2
4. Joining pairs by VSEARCH via clconcatpairv
5. Denoising by DADA2 via cldenoiseseqd
6. Removing chimeras by UCHIME3 via clremovechimev
7. Removing contaminants by clremovecontam
8. Additional clustering by VSEARCH via clclassseqv (Optional)
9. Dividing pairs by cldivseq
10. Assigning taxonomy by clmakecachedb,clidentseq,classigntax x2
11. Additional taxonomy processing by clmergeassign,clfillassign
12. Summarizing results by clsumclass,clsumtaxa

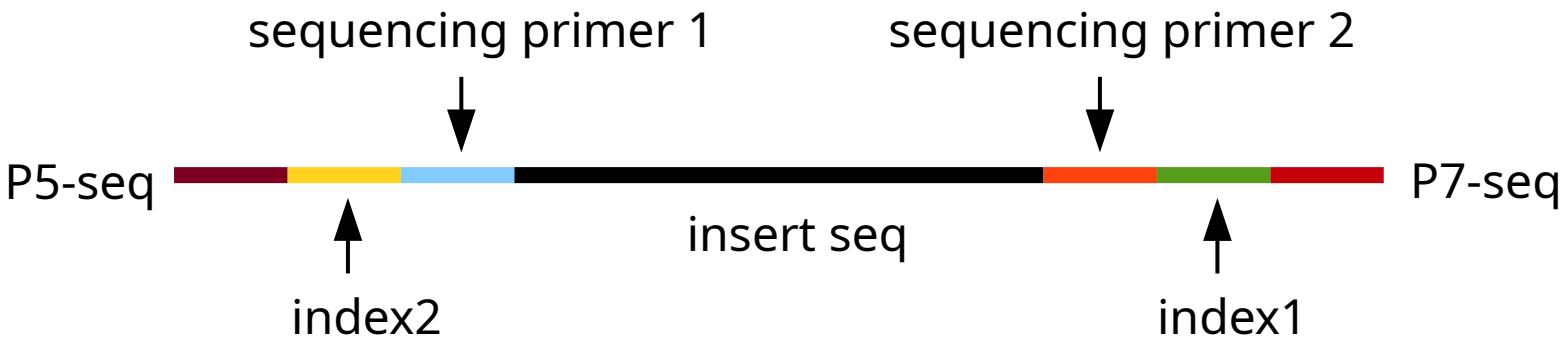
Analysis demonstration of overlapped paired-end data using Claident

- Prerequisites to run Claident
  - Debian/Ubuntu/Linux Mint, RedHat/CentOS
  - Claident+BLASTDB+TaxonomyDB+UCHIMEDB
  - Code from https://github.com/astanabe/ClaidentTutorial
- Prerequisites to learn about analyses using Claident and R
  - Code from https://github.com/astanabe/ClaidentTutorial
    - This includes simulated data and all results

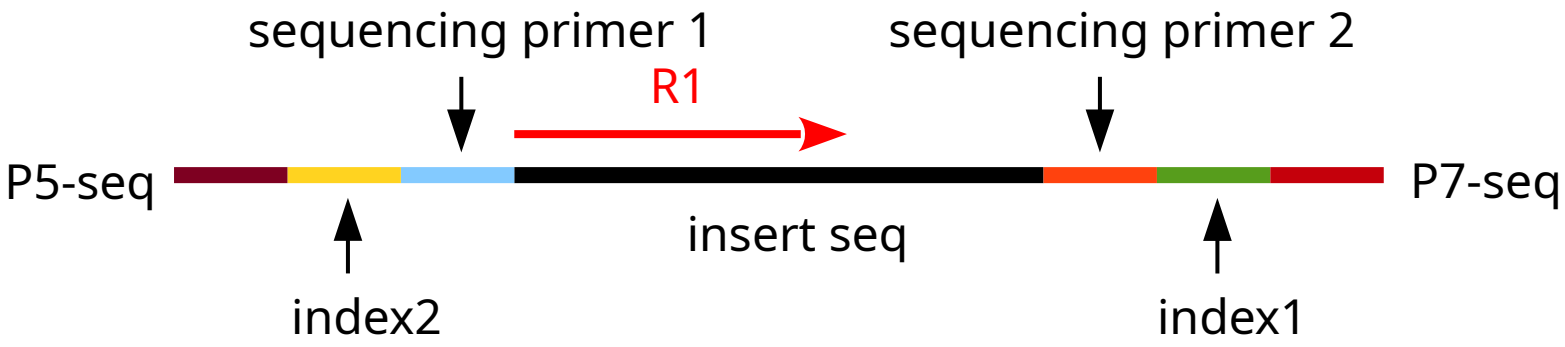
Chapter 0: Simulated data creation

1. Download complete mitogenome seq data of fishes
2. Extract 12S rRNA region
3. Run *in silico* PCR using MiFish-U primer by ecoPCR and obtain amplicons
4. Cluster amplicon seqs and pick representative seqs
5. Randomly pick 50 seqs from all repseqs (1st sample)
6. Randomly pick 40 seqs from previous sample and randomly pick 10 seqs from all repseqs except for previous sample seqs (2nd-20th sample)
7. Pick all sequences from all 1st-20th samples for blank (1st-4th blank)
8. Generate 500 paired-end seqs for each picked seqs by ART for samples
9. Generate 50 paired-end seqs for each picked seqs by ART for blanks
10. Generate dual index seqs based on given fasta files

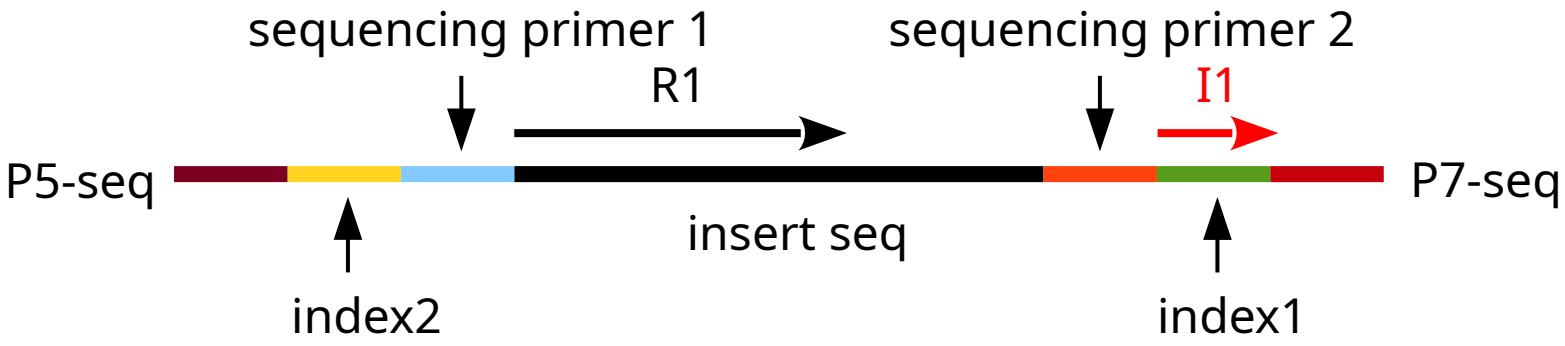
Interlude: The structure of Illumina dual-index library



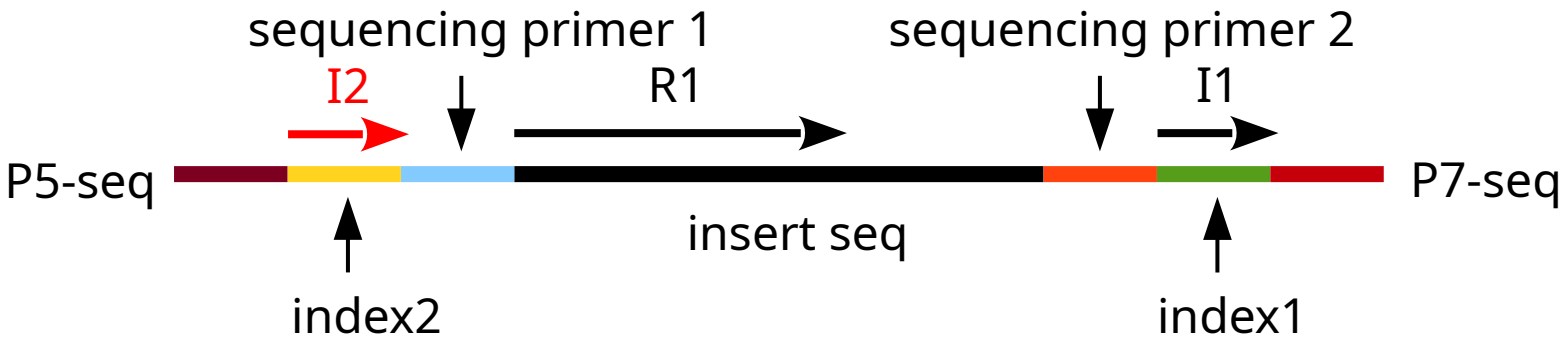
Interlude: The read order and strand of Illumina dual-index library



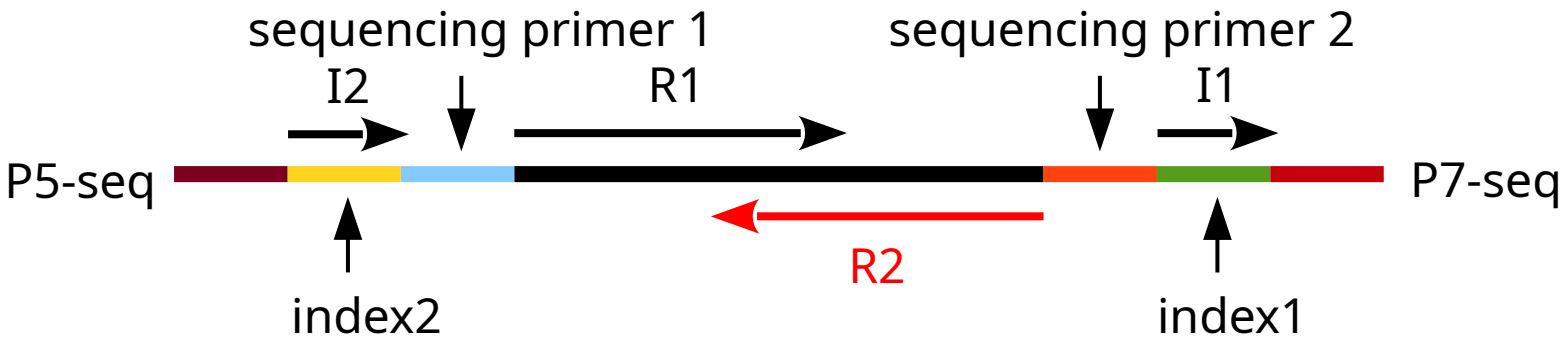
Interlude: The read order and strand of Illumina dual-index library



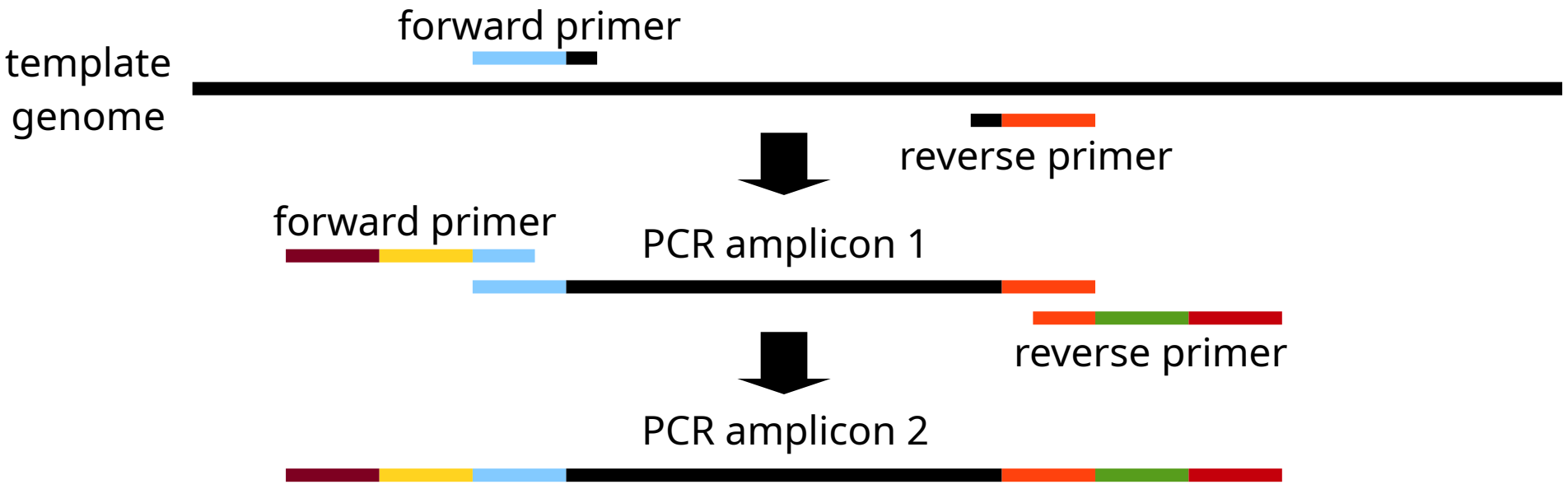
Interlude: The read order and strand of Illumina dual-index library



Interlude: The read order and strand of Illumina dual-index library



Interlude: Preparation of Illumina dual-index library



By 8 forward index primers and 12 reverse index primers,
96 samples can be distinguished (combinatorial dual-indexing).

Interlude: Dual-index design of simulated data

reverse index (index1)				
TTGCAGGT	Sample01	Sample07	not used	not used
CAAGGAAC	Sample02	Sample08	not used	not used
AGATCTGG	Sample03	Sample09	not used	not used
TCACACTT	Sample04	Sample10	not used	not used
GATCATGG	Sample05	Sample11	not used	not used
AGACATGA	Sample06	Sample12	not used	not used
GTGAGTTG	not used	not used	Sample13	Sample19
AGTCTGTT	not used	not used	Sample14	Sample20
AACCAACC	not used	not used	Sample15	Blank01
AGTGTGCA	not used	not used	Sample16	Blank02
CATGTCGA	not used	not used	Sample17	Blank03
CGAGACTT	not used	not used	Sample18	Blank04
	AACCTCTC	**GTGACTCT**	**GATCACCA**	**CTTCACAT**

forward index (index2)

Chapter 1: Demultiplexing

- Inputs
- Undemultiplexed_R1_001.fastq.xz
- Undemultiplexed_I1_001.fastq.xz
- Undemultiplexed_I2_001.fastq.xz
- Undemultiplexed_R2_001.fastq.xz

in 01_RawSequences

- index1.fasta
- index2.fasta
- forwardprimer.fasta
- reverseprimer.fasta

in top directory

- Outputs
- ClaidentTutorial__*__MiFish.forward.fastq.xz
- ClaidentTutorial__*__MiFish.reverse.fastq.xz
  - Sample*
  - Blank*
  - NNNNNNNN+NNNNNNNN

in 02a_DemultiplexedSequences

Launch Terminal

Chapter 2: Concatenating pairs

- Inputs
- ClaidentTutorial__*__MiFish.forward.fastq.xz
- ClaidentTutorial__*__MiFish.reverse.fastq.xz
  - Sample*
  - Blank*
  - NNNNNNNN+NNNNNNNN

  in 02a_DemultiplexedSequences

- Outputs
- ClaidentTutorial__*__MiFish.fastq.xz
  - Sample*
  - Blank*
  - NNNNNNNN+NNNNNNNN

  in 03_ConcatenatedSequences

Chapter 2: Concatenating pairs

Switch to Terminal

Chapter 3: Quality-filtering

- Inputs
- ClaidentTutorial__*__MiFish.fastq.xz
  - Sample*
  - Blank*
  - NNNNNNNN+NNNNNNNN

    in 03_ConcatenatedSequences

- Outputs
- ClaidentTutorial__*__MiFish.fastq.xz
  - Sample*
  - Blank*
  - NNNNNNNN+NNNNNNNN

    in 04_FilteredSequences

Chapter 3: Quality-filtering

Switch to Terminal

Chapter 4: Denoising

- Inputs
- ClaidentTutorial__*__MiFish.fastq.xz
  - Sample*
  - Blank*
  - NNNNNNNN+NNNNNNNN

in 04_FilteredSequences

- Outputs
- denoised.fasta
- denoised.otu.gz
- denoised.tsv
- plotErrors.pdf
- runDADA2.R

in 05_DenoisedSequences

Chapter 4: Denoising

Switch to Terminal

Interlude: Methods in DADA2

	observed number
ACCTCTCGATATCGAGATGAGGCT	10000
ACCTCTTGATATCGAGATGAGGCT	10
ACCTCTCGAAATCGAGATGAGGCT	7
ACCTCTGGATATCGAGATGAGGCT	200

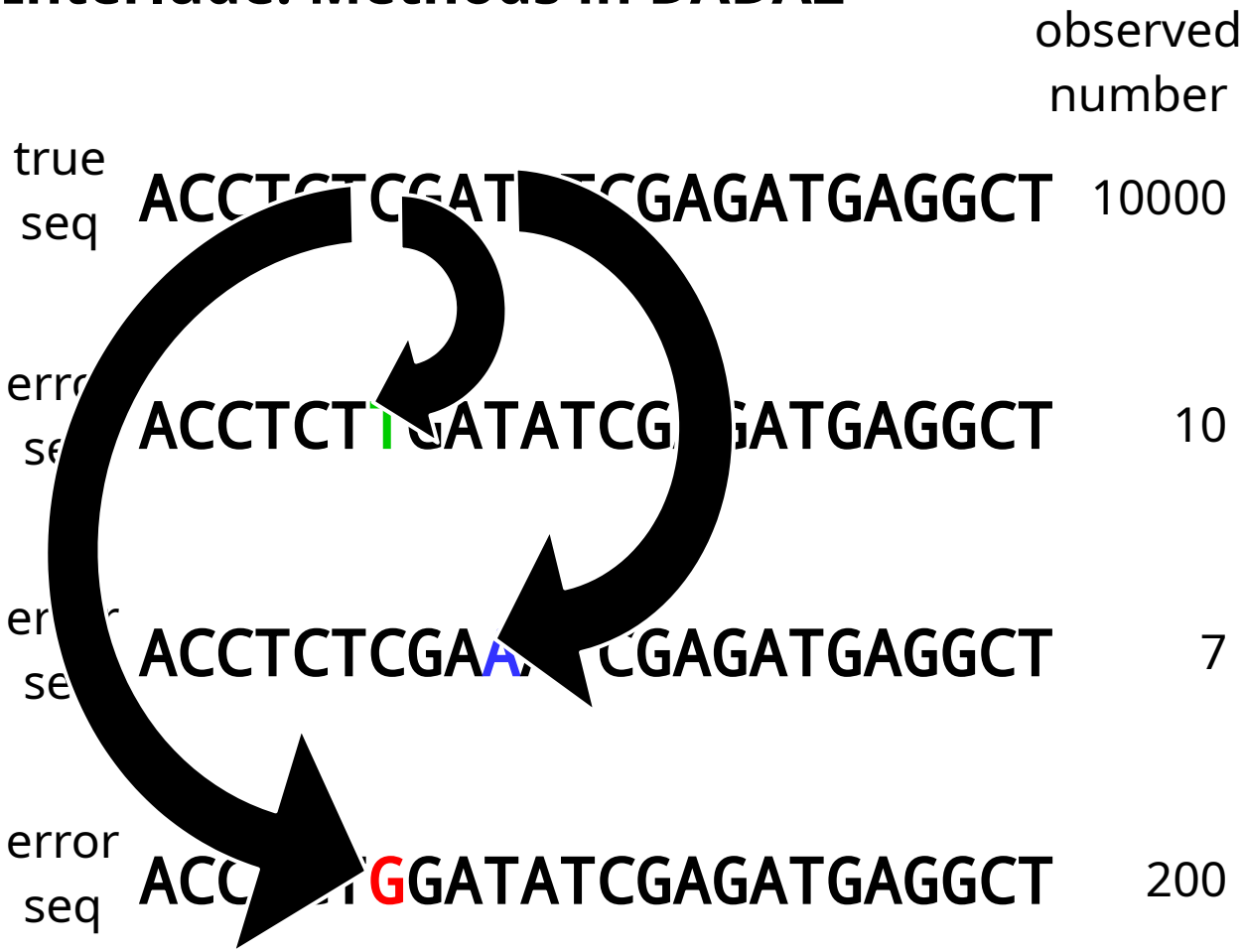
See also Callahan et al. (2016) https://doi.org/10.1038/nmeth.3869

Interlude: Methods in DADA2

		observed number
true seq	ACCTCTCGATATCGAGATGAGGCT	10000
error seq	ACCTCTTGATATCGAGATGAGGCT	10
error seq	ACCTCTCGAAATCGAGATGAGGCT	7
error seq	ACCTCTGGATATCGAGATGAGGCT	200

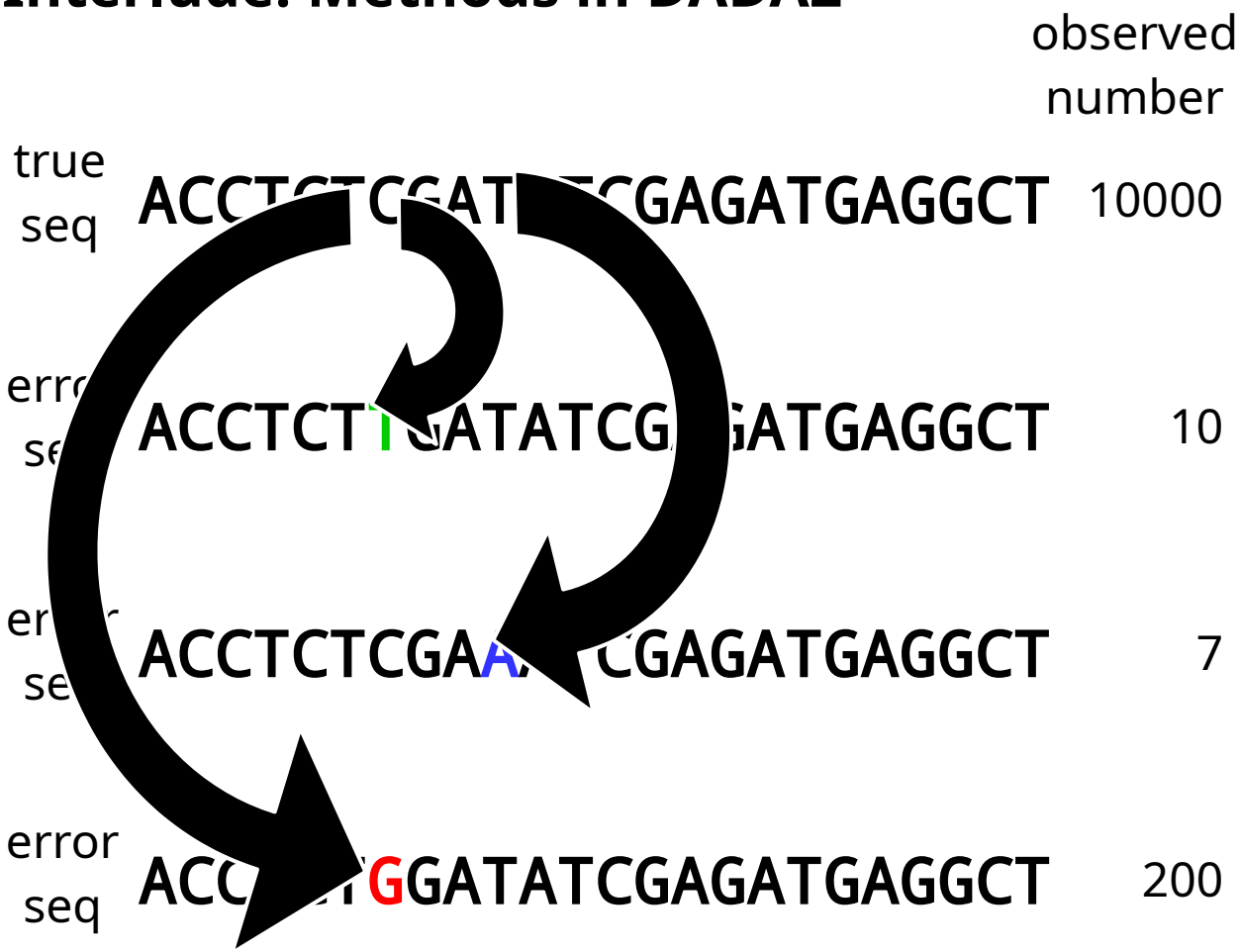
See also Callahan et al. (2016) https://doi.org/10.1038/nmeth.3869

Interlude: Methods in DADA2



See also Callahan et al. (2016) https://doi.org/10.1038/nmeth.3869

Interlude: Methods in DADA2

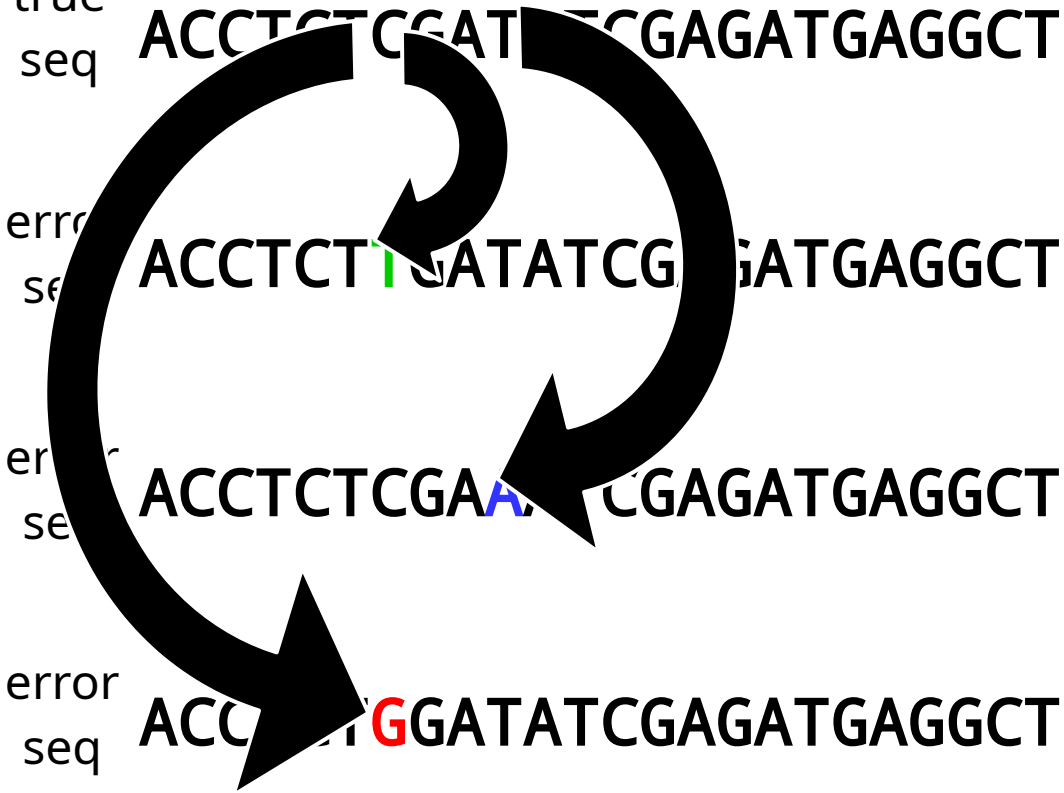


Error rate matrix

from \ to	A	C	G	T
A	R_{AA}	R_{AC}	R_{AG}	R_{AT}
C	R_{CA}	R_{CC}	R_{CG}	R_{CT}
G	R_{GA}	R_{GC}	R_{GG}	R_{GT}
T	R_{TA}	R_{TC}	R_{TG}	R_{TT}

See also Callahan et al. (2016) https://doi.org/10.1038/nmeth.3869

Interlude: Methods in DADA2



observed number

10000

10

7

200

abundance p-value

NA

0.3

0.5

0.001

Error rate matrix

from \ to	A	C	G	T
A	R_{AA}	R_{AC}	R_{AG}	R_{AT}
C	R_{CA}	R_{CC}	R_{CG}	R_{CT}
G	R_{GA}	R_{GC}	R_{GG}	R_{GT}
T	R_{TA}	R_{TC}	R_{TG}	R_{TT}

See also Callahan et al. (2016) https://doi.org/10.1038/nmeth.3869

Interlude: Methods in DADA2

		observed number	abundance p-value
true seq	ACCTCTCGATATCGAGATGAGGCT	10000	NA
error seq	ACCTCTTGATATCGAGATGAGGCT	10	0.3
error seq	ACCTCTCGAAATCGAGATGAGGCT	7	0.5
error seq	ACCTCTGGATATCGAGATGAGGCT	200	0.001

Error rate matrix

from \ to	A	C	G	T
A	R_{AA}	R_{AC}	R_{AG}	R_{AT}
C	R_{CA}	R_{CC}	R_{CG}	R_{CT}
G	R_{GA}	R_{GC}	R_{GG}	R_{GT}
T	R_{TA}	R_{TC}	R_{TG}	R_{TT}

Observed number cannot be explain as error

See also Callahan et al. (2016) https://doi.org/10.1038/nmeth.3869

Interlude: Methods in DADA2

		observed number	abundance p-value
true seq	ACCTCTCGATATCGAGATGAGGCT	10000	NA
error seq	ACCTCTTGATATCGAGATGAGGCT	10	0.3
error seq	ACCTCTCGAAATCGAGATGAGGCT	7	0.5
true seq	ACCTCTGGATATCGAGATGAGGCT	200	0.001

Error rate matrix

from \ to	A	C	G	T
A	R_{AA}	R_{AC}	R_{AG}	R_{AT}
C	R_{CA}	R_{CC}	R_{CG}	R_{CT}
G	R_{GA}	R_{GC}	R_{GG}	R_{GT}
T	R_{TA}	R_{TC}	R_{TG}	R_{TT}

Observed number cannot be explain as error

See also Callahan et al. (2016) https://doi.org/10.1038/nmeth.3869

Chapter 5: Chimera removal

- Inputs
- denoised.fasta
- denoised.otu.gz

in 05_DenoisedSequences

- Outputs
- nonchimeras.fasta
- nonchimeras.otu.gz
- nonchimeras.tsv
- *_borderline.fasta
- *_chimeras.fasta
- *_nonchimeras.fasta
- *_uchimealns.txt
- *_uchimeout.txt

in 06_NonchimericSequences

Chapter 5: Chimera removal

Switch to Terminal

Chapter 6: Removing index-hopped sequences

- Inputs
- nonchimeras.fasta
- nonchimeras.otu.gz

in 06_NonchimericSequences

- index1.fasta
- index2.fasta

in top directory

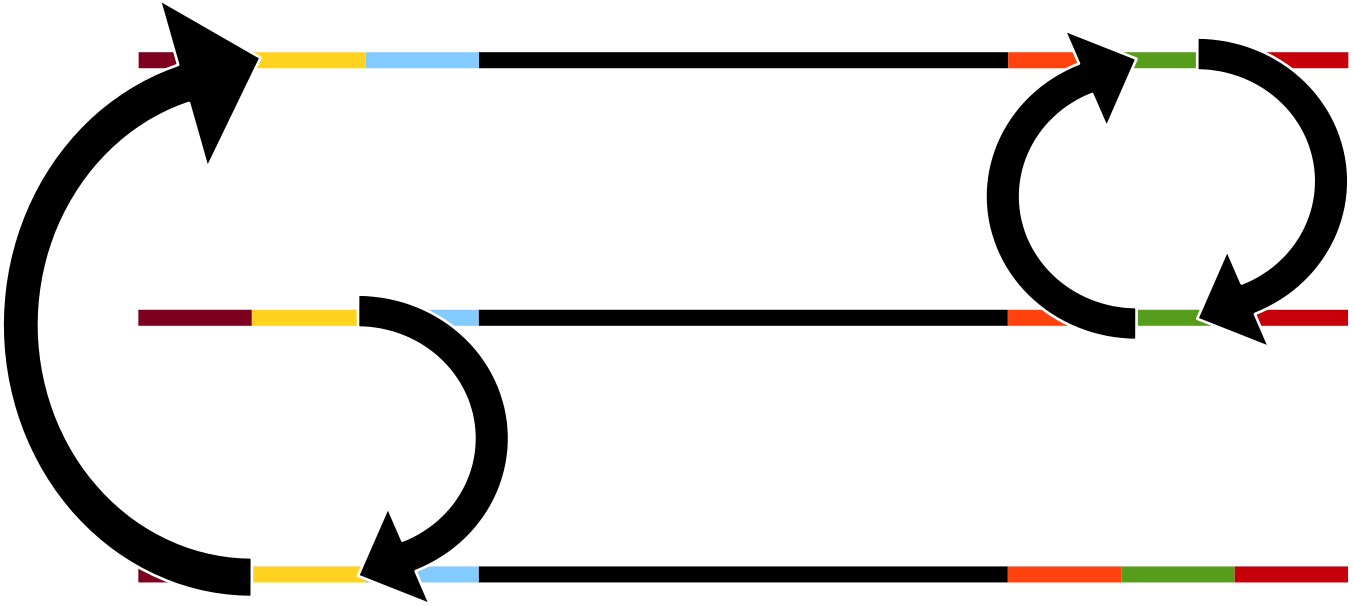
- Outputs
- decontaminated.fasta
- decontaminated.otu.gz
- decontaminated.tsv

in 07_NonhoppedSequences

Switch to Terminal

Interlude: Index can hop into another amplicon within a flowcell!

Index-hopping potentially causes sequence misassignments!
Especially in newer models! OMG!



Interlude: Detecting index-hopping using unused index combinations

reverse index (index1)

TTGCAGGT	Sample01	Sample07	not used	not used
CAAGGAAC	Sample02	Sample08	not used	not used
AGATCTGG	Sample03	Sample09	not used	not used
TCACACTT	Sample04	Sample10	not used	not used
GATCATGG	Sample05	Sample11	not used	not used
AGACATGA	Sample06	Sample12	not used	not used
GTGAGTTG	not used	not used	Sample13	Sample19
AGTCTGTT	not used	not used	Sample14	Sample20
AACCAACC	not used	not used	Sample15	Blank01
AGTGTGCA	not used	not used	Sample16	Blank02
CATGTCGA	not used	not used	Sample17	Blank03
CGAGACTT	not used	not used	Sample18	Blank04
	AACCTCTC	**GTGACTCT**	**GATCACCA**	**CTTCACAT**

forward index (index2)

1. Count abundances
2. Collect abundances of a sample + "not used"
3. Test whether sample abundance is outlier or not
4. If it's not outlier, it's determined as hopped

See also Esling et al. (2015) https://doi.org/10.1093/nar/gkv107

Chapter 7: Removing contaminant sequences

- Inputs
- decontaminated.fasta
- decontaminated.otu.gz

  in 07_NonhoppedSequences
- blanklist.txt

  in top directory

- Outputs
- decontaminated.fasta
- decontaminated.otu.gz
- decontaminated.tsv

  in 08_DecontaminatedSequences

Chapter 7: Removing contaminant sequences

Switch to Terminal

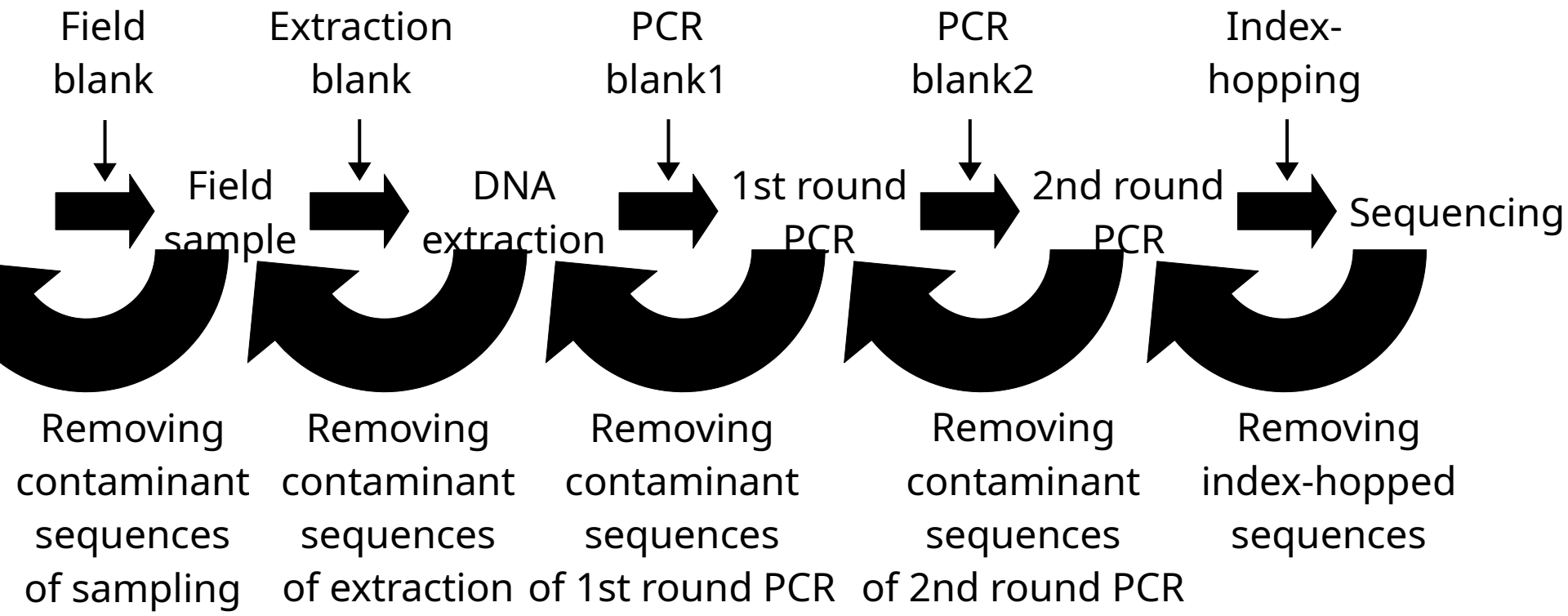
Interlude: Detecting contaminants using blank samples

1. Count abundances
2. Collect abundances of a sample + associated blanks
3. Test whether sample abundance is outlier or not
4. If it's not outlier, it's determined as contaminant

See also Esling et al. (2015) https://doi.org/10.1093/nar/gkv107

Interlude: Multistep contamination and multistep decontamination

My recommendation is index-hopping removal + the other contaminant removal. However, the best practice has been still unknown.



Interlude: Study purpose and decontamination

- Non-decontaminated metabarcoding results contain contaminants
- Decontamination should be applied?
  - If you want to maximize detection power, NO. Decontamination potentially misidentify true sequence as contaminant
  - If you want to minimize misdetection, YES. Lack of decontamination may cause many misdetection
  - If you want to analyse community composition, UNKNOWN. Because abundances of contaminants may be low in many cases, their effects to analysis may be low. However, whether abundances of contaminants are really low or not IN YOUR DATA is unknown.

Chapter 8: Additional clustering

- Inputs
- decontaminated.fasta
- decontaminated.otu.gz

  in 08_DecontaminatedSequences

- Outputs
- clustered.fasta
- clustered.otu.gz
- clustered.tsv

  in 09_ClusteredSequences

Switch to Terminal

Chapter 9: Taxonomic assignment

- Inputs
- clustered.fasta

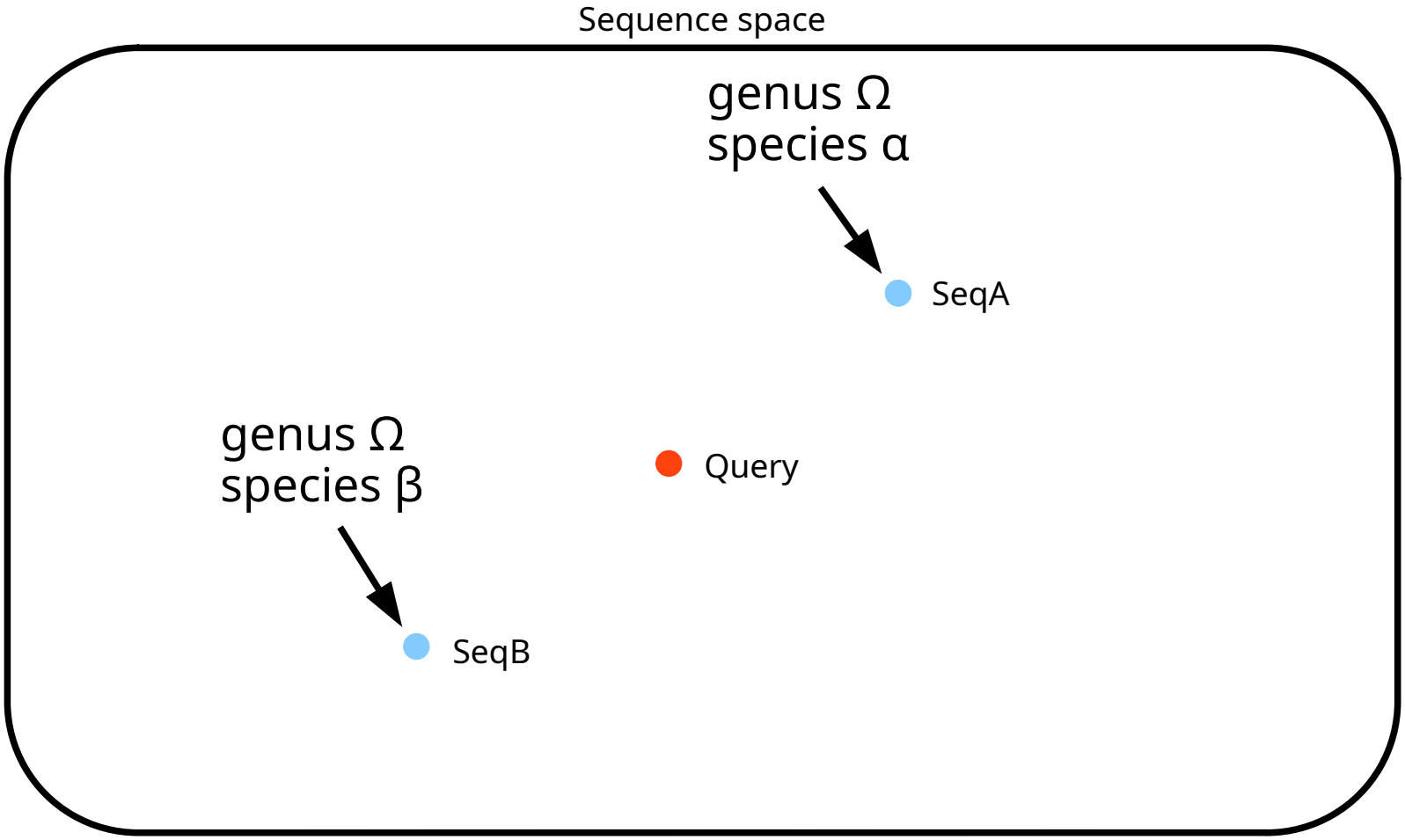
in 09_ClusteredSequences

- Outputs
- neighborhoods_1nn_*.txt
- neighborhoods_qc_*.txt
- taxonomy_1nn_*.tsv
- taxonomy_qc_*.tsv
- taxonomy_merged.tsv
- taxonomy_merged_filled.tsv

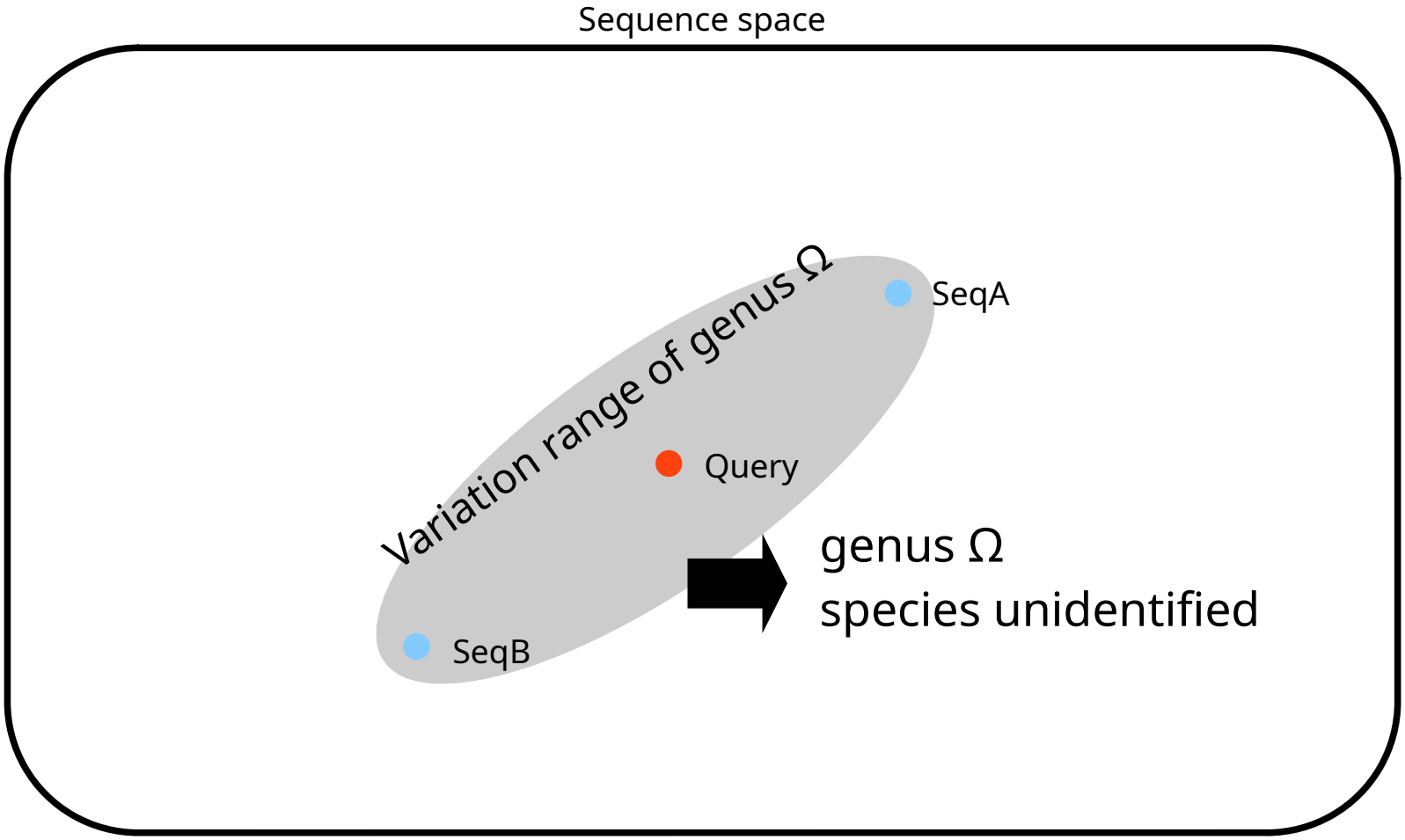
in 10_ClaidentResults

Switch to Terminal

Interlude: Query-centric auto-*k*-nearest neighbor method



Interlude: Query-centric auto-*k*-nearest neighbor method



Interlude: Which method should be used for taxonomic assignment?

- If reference database is imperfect (most cases), QCauto shows the best balance between less misidentification and less successful identification
- If reference database is perfect or nearly perfect, 1-NN is the best. However, whether the reference database is really perfect or not should not be known by anyone

Interlude: Ready-made reference databases

- Installed to INSTALLPATH/share/claident/blastdb
- overall_class, overall_order, overall_family
  - Subset of NCBI nt including class, order or family level identified seqs
- *_genus
  - Subset of overall_* including genus level identified seqs
- *_species_wsp
  - Subset of overall_* including species level identified seqs
- *_species
  - Subset of overall_* including species level identified seqs except for the seqs which have " sp." at the tail in species name
- *_species_wosp
  - Subset of overall_* including species level identified seqs except for the seqs which have " sp." in species name

Interlude: Taxonomic infomation reliability in reference databases

*_species_wosp＞*_species＞*_species_wsp＞*_genus＞*_family＞*_order＞*_class

- Because the seqs which only have higher level taxonomic info likely to be identified based on closest INSD seqs, such taxonomic info are less reliable
- Because the seqs identified as " sp." is not strictly identified or such species are undescribed, such taxonomic info are less reliable

Interlude: Which reference database should be used?

- overall_species_wosp is recommended in most cases because the seqs lacking lower level taxonomic info likely to be less reliable
- The other overall_* are recommended if you want to minimize "unidentified" in * level and can rolerate misidentification in lower level
- The others are recommended for screening or PCs lacking enough amount of memory

Interlude: Merging of taxonomy

- More reliable taxonomy should be preferred but less reliable taxonomy which reached to lower taxonomic level could be tolerated
- The best balance between reliability and identifiability can be achieved by merging taxonomy from overall_species_wosp and the other overall_*

Chapter 10: Making summary tables

- Inputs
- clustered.tsv
  in 09_ClusteredSequences
- taxonomy_merged_filled.tsv
  in 10_ClaidentResults

- Outputs
- sample_otu_matrix_fishes.tsv
- sample_species_matrix_fishes.tsv
- sample_top50species_nreads_fishes.tsv
- sample_top50family_nreads_fishes.tsv
- sample_species_nreads_fishes.tsv
- sample_family_nreads_fishes.tsv
  in 10_ClaidentResults

Chapter 10: Making summary tables

Switch to Terminal

Chapter 11: Plotting community structure

- Inputs
- sample_top50species_nreads_fishes.tsv
- sample_top50family_nreads_fishes.tsv
- sample_species_nreads_fishes.tsv
- sample_family_nreads_fishes.tsv

in 10_ClaidentResults

- Outputs
- barplottop50species.pdf
- barplottop50family.pdf
- heatmapspecies.pdf
- heatmapfamily.pdf

in 11_RAnalysisResults

Switch to Terminal

Chapter 12: Plotting sampling/sequencing coverage

- Inputs
- sample_species_matrix_fishes.tsv in 10_ClaidentResults

- Outputs
- specaccum.pdf
- rarecurve.pdf in 11_RAnalysisResults
- Community (data.frame) in R workspace

Chapter 12: Plotting sampling/sequencing coverage

Switch to Terminal

Chapter 12: Applying coverage-based rarefaction

- Inputs
- Community (data.frame)

in R workspace

- Outputs
- RarefiedCommunity (data.frame)
- BinaryRarefiedCommunity (d.f.)

in R workspace

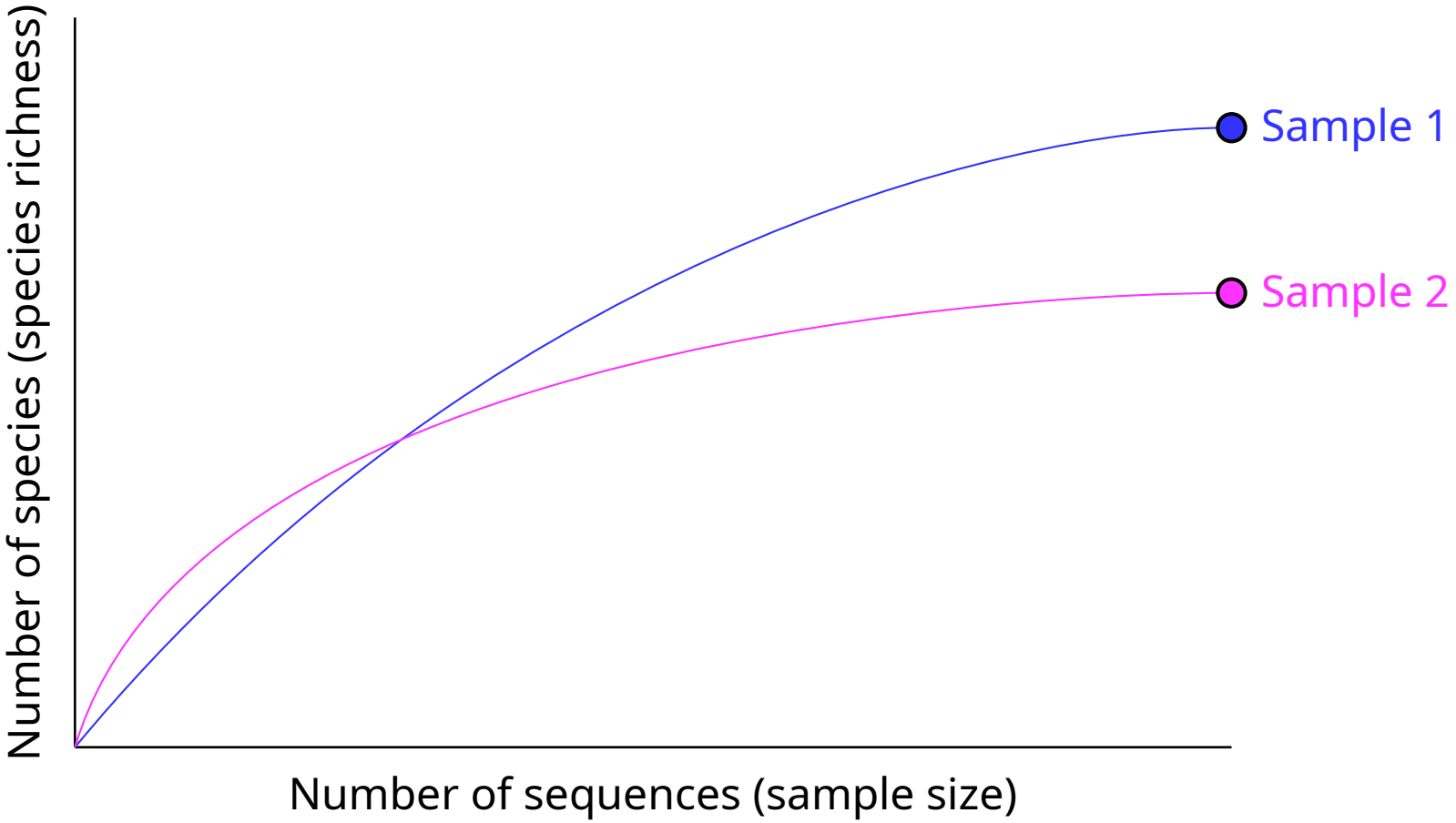
- RarefiedCommunity.tsv
- BinaryRarefiedCommunity.tsv

in 11_RAnalysisResults

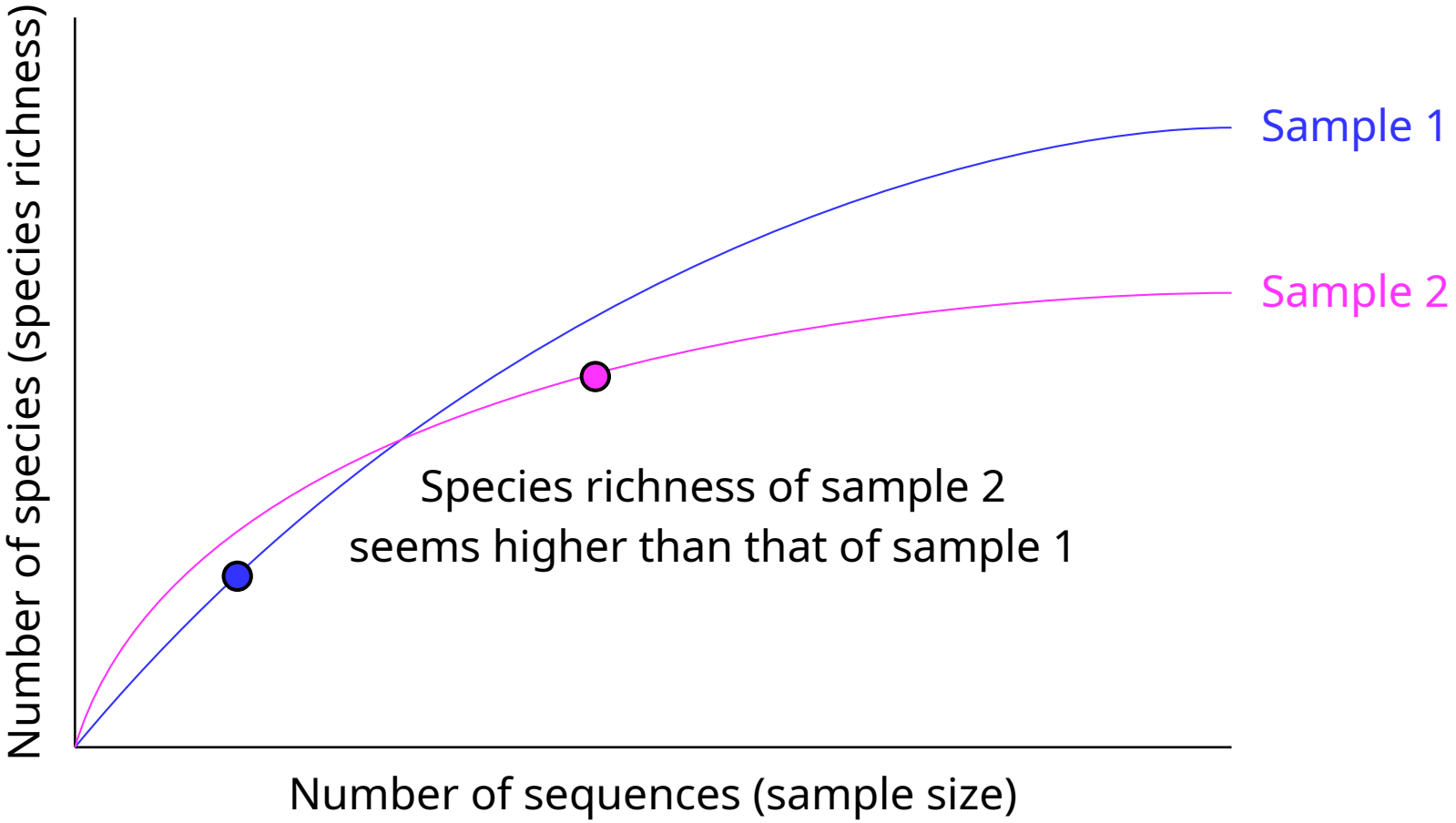
Chapter 12: Applying coverage-based rarefaction

Switch to Terminal

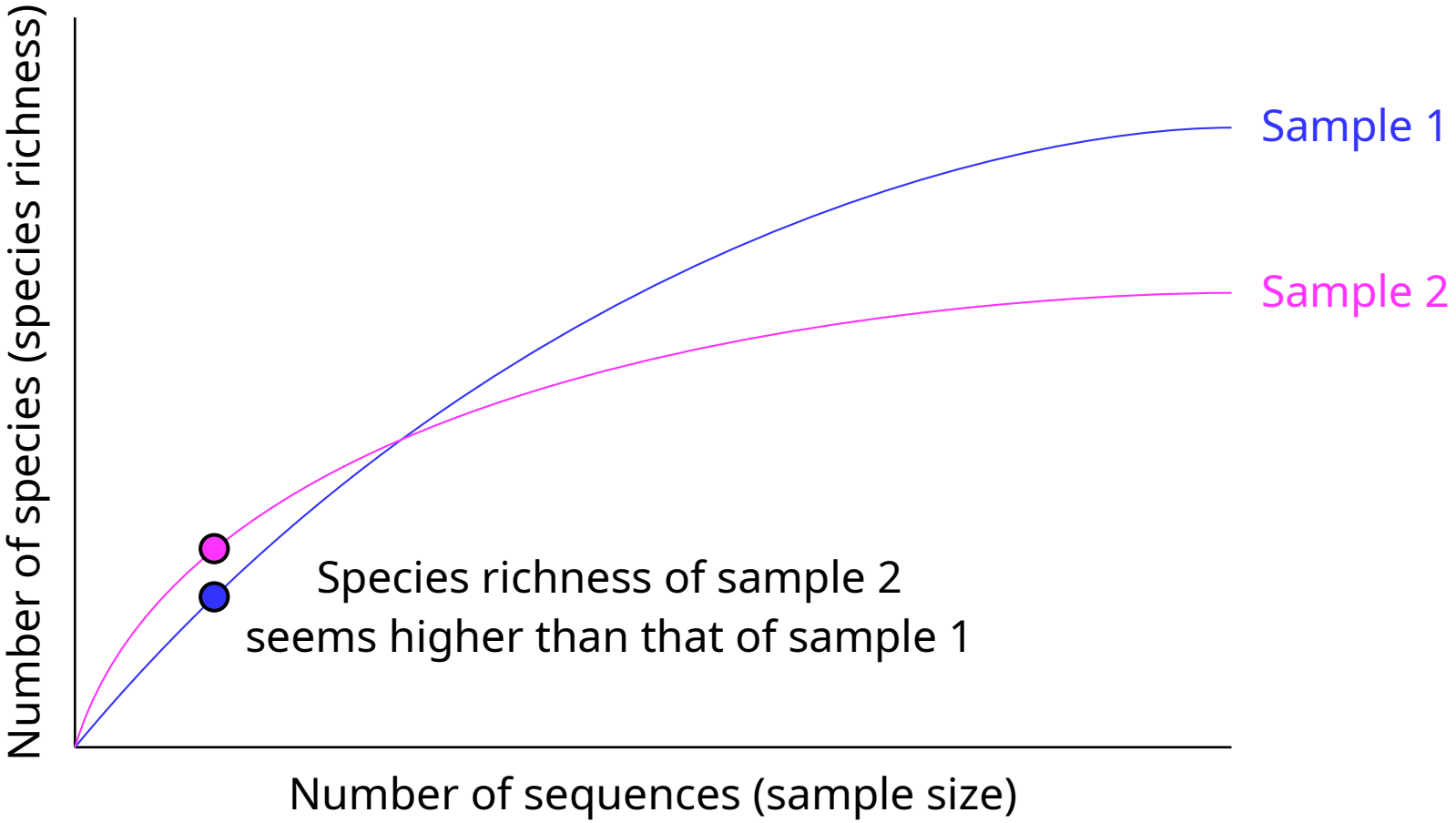
Interlude: Problem of unequal sequencing effort



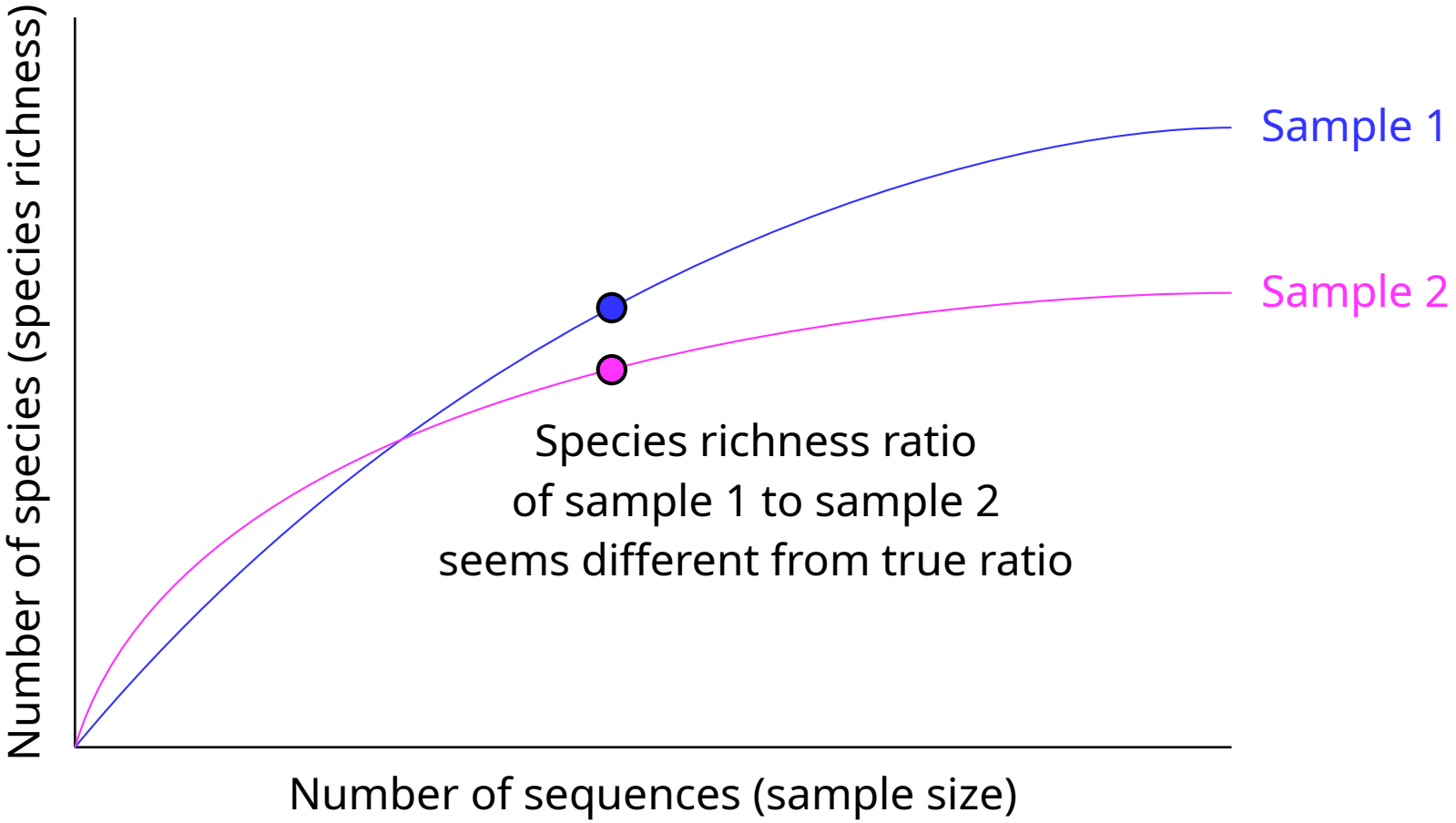
Interlude: Problem of unequal sequencing effort



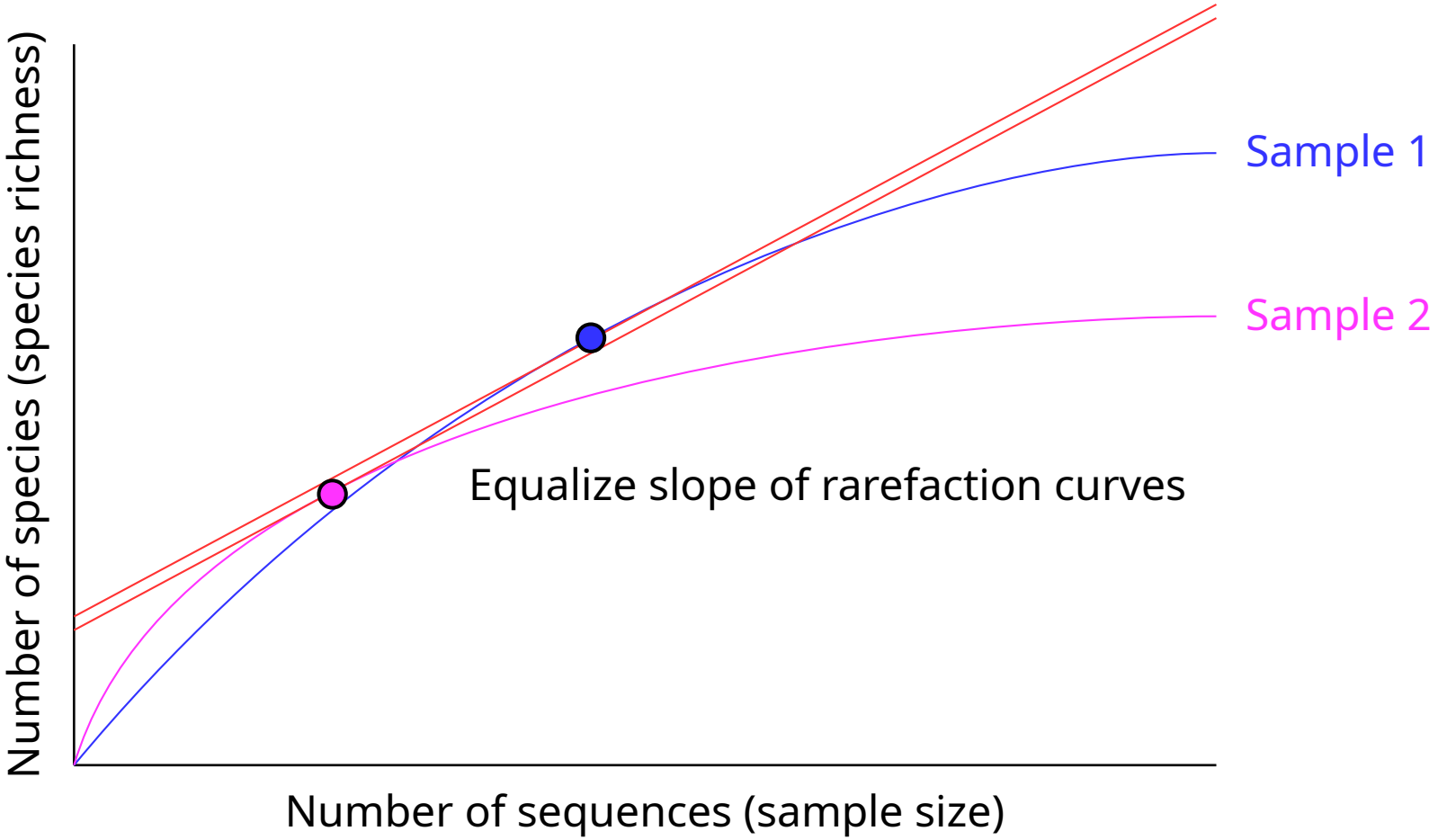
Interlude: Problem of unequal sequencing effort



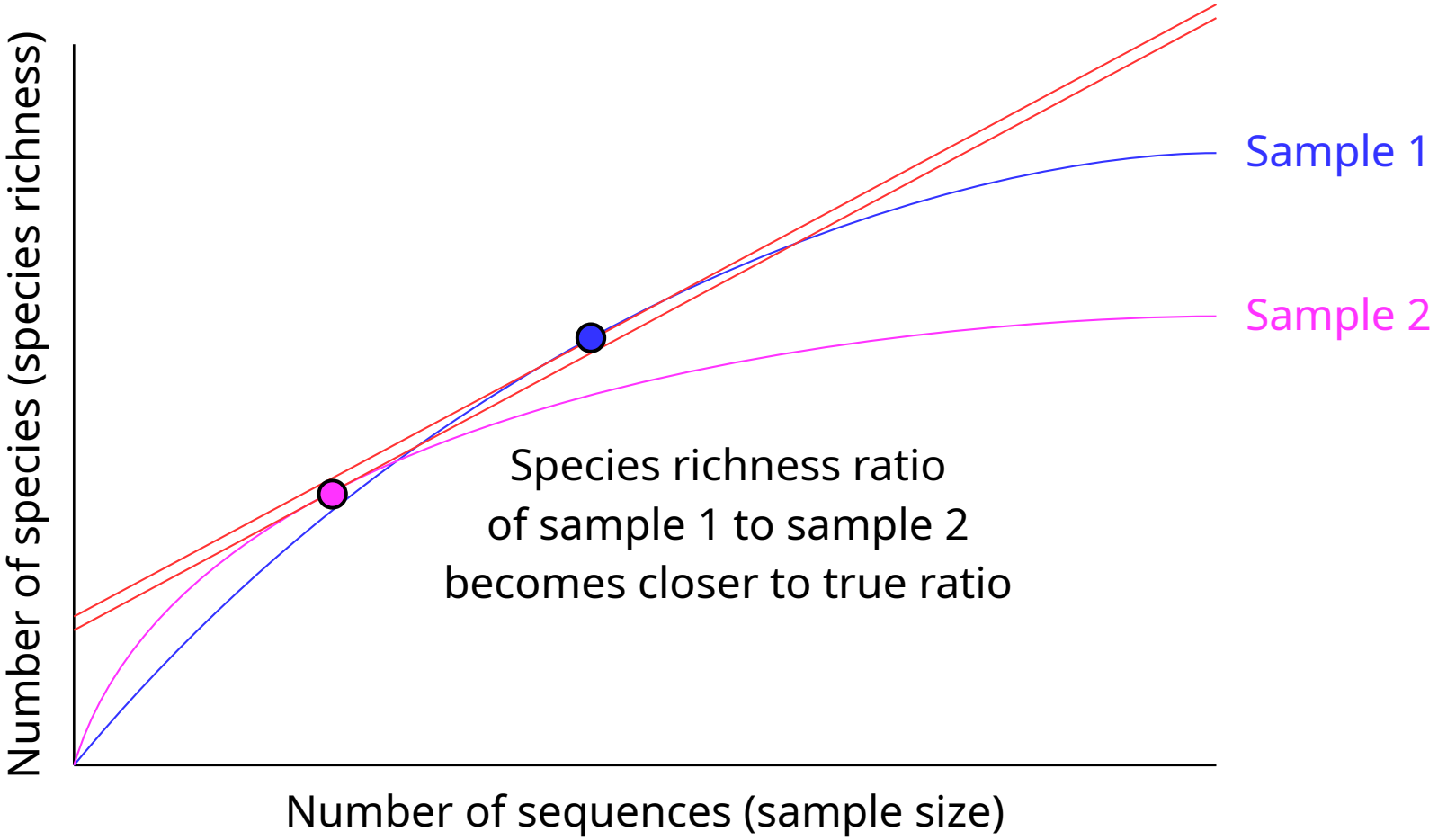
Interlude: Problem of unequal sequencing effort



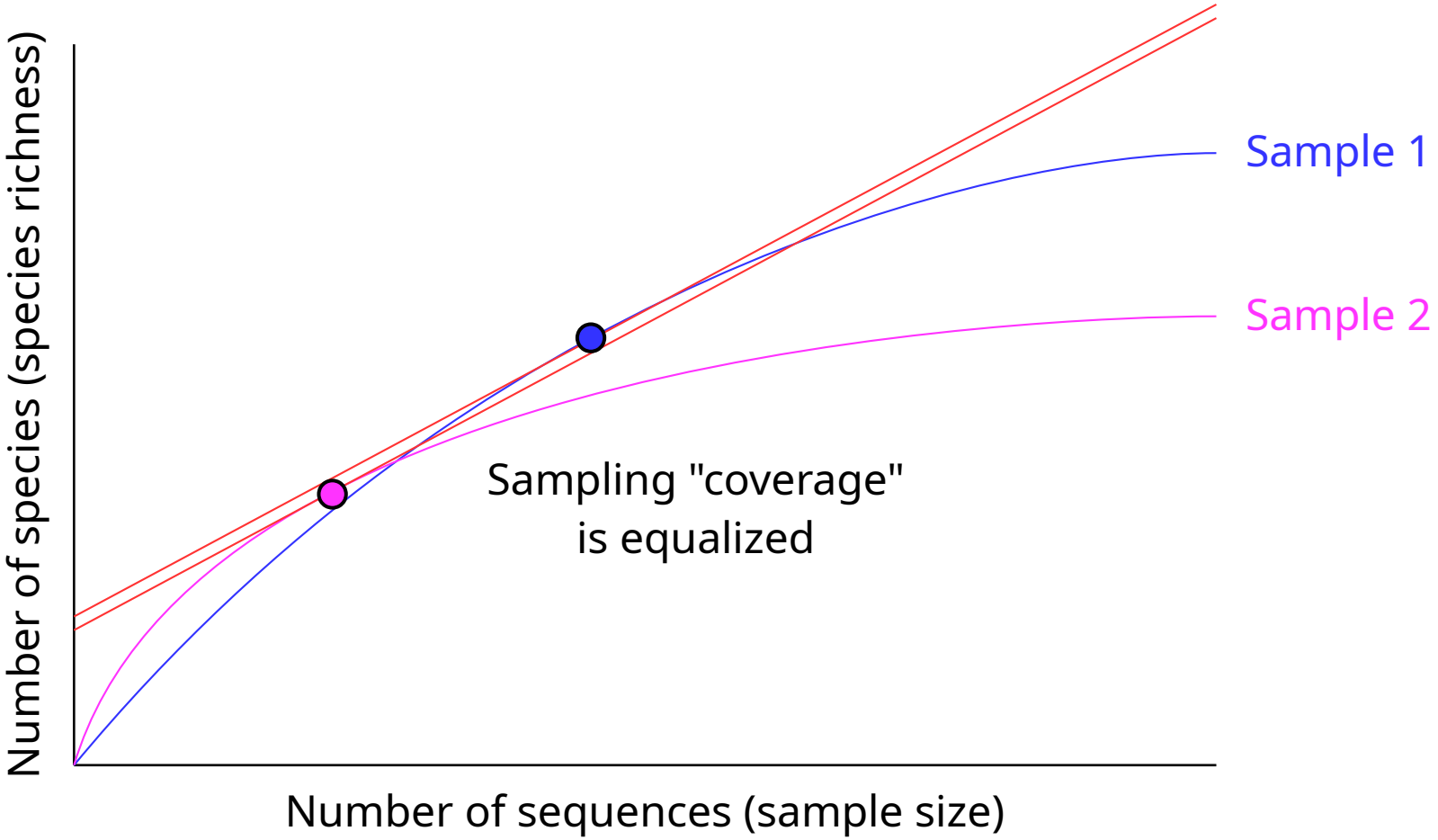
Interlude: Problem of unequal sequencing effort



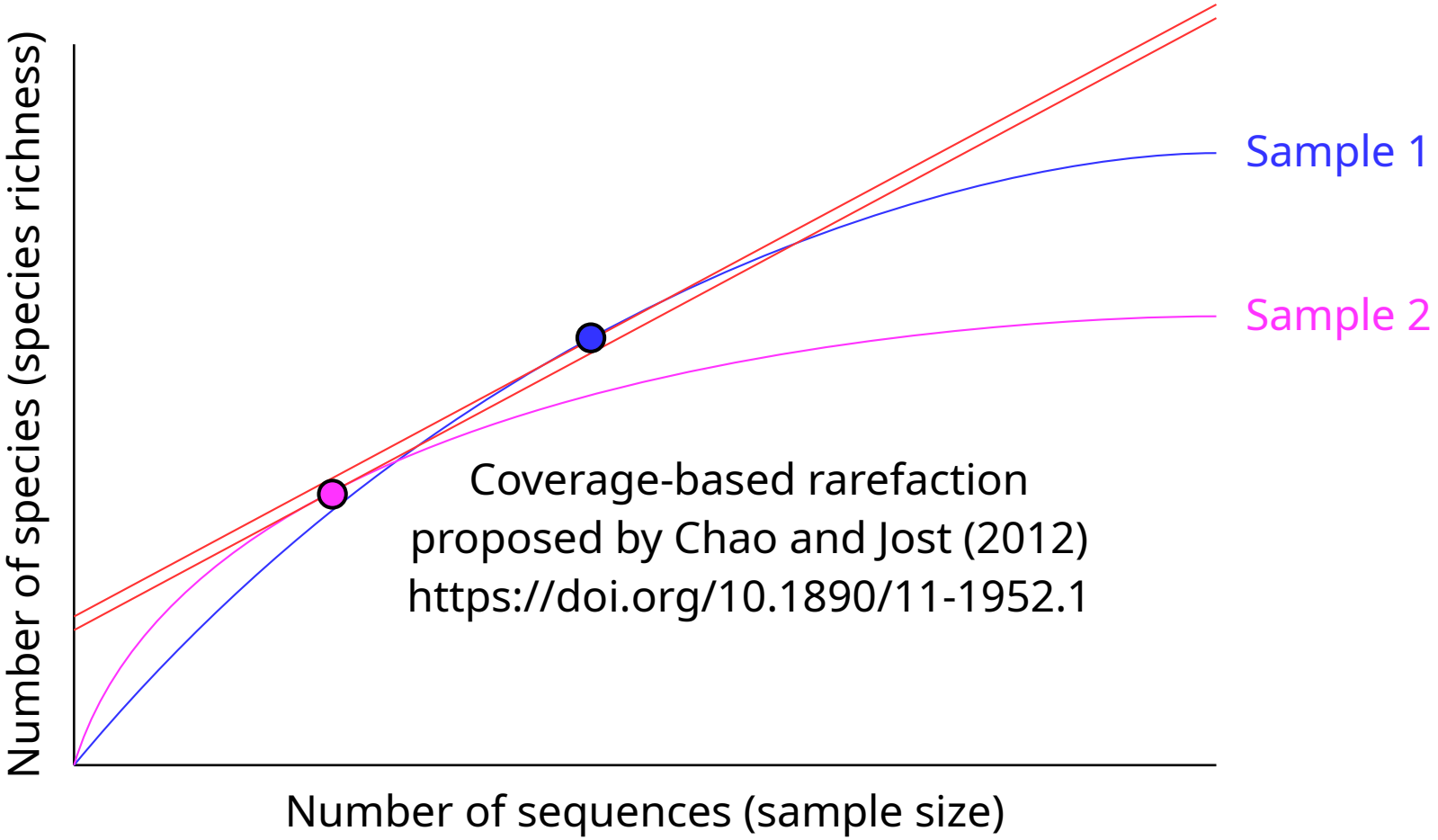
Interlude: Problem of unequal sequencing effort



Interlude: Problem of unequal sequencing effort



Interlude: Problem of unequal sequencing effort



Chapter 13: Calculating distance matrices

- Inputs
- RarefiedCommunity (data.frame)
- BinaryRarefiedCommunity (d.f.)

in R workspace

- Outputs
- BrayCurtis (dist)
- Jaccard (dist)
- BinaryJaccard (dist)
- BinaryRaupCrick (dist)

in R workspace

Chapter 13: Calculating distance matrices

Switch to Terminal

Interlude: Community distance (β diversity) metrics, PERMANOVA and NMDS

See

- Anderson et al. (2010) https://doi.org/10.1111/j.1461-0248.2010.01552.x
- Anderson (2001) https://doi.org/10.1111/j.1442-9993.2001.01070.pp.x
- Anderson (2017) https://doi.org/10.1002/9781118445112.stat07841
- 土居・岡村 (2010) https://doi.org/10.18960/seitai.61.1_3

Chapter 14: Executing PERMANOVA

- Inputs
- BrayCurtis (dist)
- Jaccard (dist)
- BinaryJaccard (dist)
- BinaryRaupCrick (dist)

in R workspace

- Metadata.tsv

in top directory

- Outputs
- BrayCurtisPERMANOVA.txt
- JaccardPERMANOVA.txt
- BinaryJaccardPERMANOVA.txt
- BinaryRaupCrickPERMANOVA.txt

in 11_RAnalysisResults

Chapter 14: Executing PERMANOVA

Switch to Terminal

Chapter 15: Executing NMDS

- Inputs
- BrayCurtis (dist)
- Jaccard (dist)
- BinaryJaccard (dist)
- BinaryRaupCrick (dist)

in R workspace

- Metadata.tsv

in top directory

- Outputs
- NMDS.pdf

in 11_RAnalysisResults

Switch to Terminal

Chapter 16: Executing Mantel correlogram analysis using geographical distance

- Inputs
- BrayCurtis (dist)
- Jaccard (dist)
- BinaryJaccard (dist)
- BinaryRaupCrick (dist)

  in R workspace
- Metadata.tsv

  in top directory

- Outputs
- GeoMCA.pdf

  in 11_RAnalysisResults

Chapter 16: Executing Mantel correlogram analysis using geographical distance

Switch to Terminal

Chapter 17: Executing Mantel correlogram analysis using date interval

- Inputs
- BrayCurtis (dist)
- Jaccard (dist)
- BinaryJaccard (dist)
- BinaryRaupCrick (dist)

  in R workspace
- Metadata.tsv

  in top directory

- Outputs
- DateMCA.pdf

  in 11_RAnalysisResults

Chapter 17: Executing Mantel correlogram analysis using date interval

Switch to Terminal

Conclusion: Metabarcoding analysis using Claident and R

- Claident is integrated package for translation from high-throughput amplicon sequence data into ecological communities
- Claident can remove contaminants including index-hopped sequences using unused index combinations and blank samples (negative controls)
- Most studies lack decontamination and this might affect the conclusion of such studies
- Detection power of metabarcoding should re-evaluate using decontamination and our knowledge of that might need to be revised
- Claident provides multiple taxonomic assignment methods and can merge those results
- R can import tab-separated text made by Claident
- vegan is strongly recommended for community ecological analyses