

W31-2

NGS データからの 超大規模群集行列データの作成

田辺晶史 (水産総合研究センター・中央水産研究所)

集会 Web ページ

メタバーコーディング・環境 DNA バーコーディング解析の技法

検索

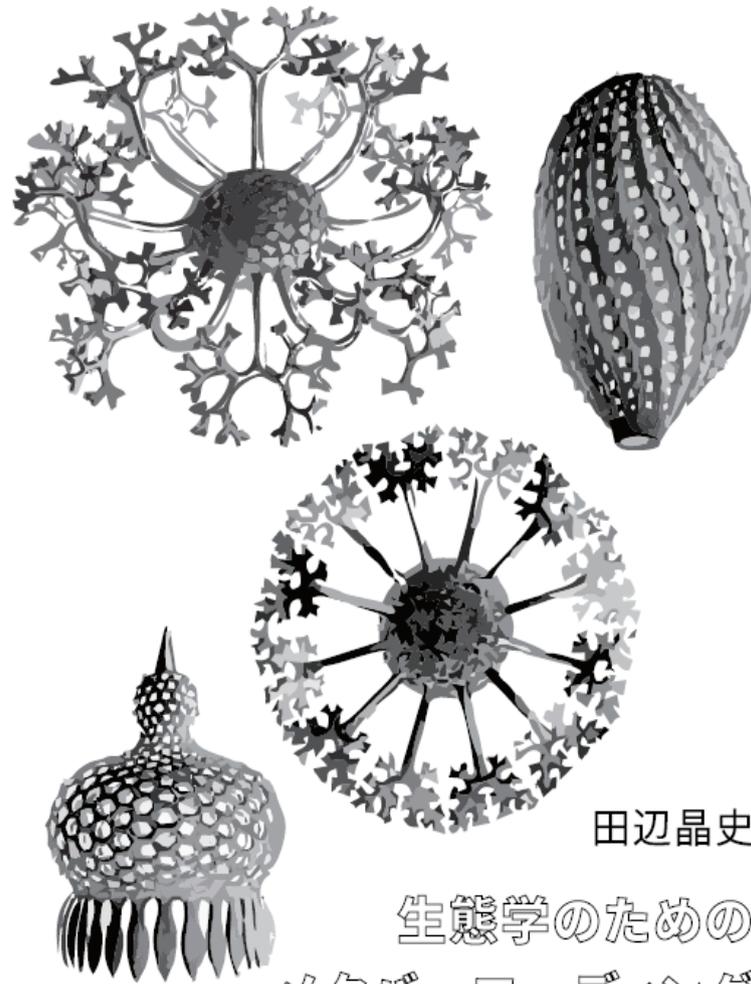
The screenshot shows a Google search page with the following elements:

- Browser Tab:** メタバーコーディン... x +
- Address Bar:** <https://www.google.co.jp/search?q=メタバーコーディング・環境DNAバーコ>
- Search Bar:** メタバーコーディング・環境DNAバーコーディング解析の技法
- Navigation:** すべて, 画像, ショッピング, ニュース, 動画, もっと見る, 検索ツール
- Results:**
 - 約 11,700 件 (0.27 秒)
 - 自由集会2016:メタバーコーディング・環境DNAバーコーディング ...**
www.fifthdimension.jp/wiki.cgi?...自由集会2016:メタバーコーディング...
メタバーコーディング・環境DNAバーコーディング解析の技法 に移動 - このページは日本生態学会2016年年次大会の自由集会『メタバーコーディング・環境DNAバーコーディング解析の技法』のページです。
 - FrontPage - Life is fifthdimension. - Akifumi S. Tanabe**
www.fifthdimension.jp/wiki.cgi
現在は独立行政法人水産総合研究センター・中央水産研究所・水産遺伝子解析センター・任期付研究員。専門は進化生物 ... 2009年4月: 筑波大学大学院生命環境科学研究科構造生物科学専攻微生物学分野微生物分子進化学研究室準研究員採用 ... 田辺晶史, 2013, "メタバーコーディング: DNAバーコーディングを応用した新しい未知生物探索技術", 日本進化学会第15回つくば大会シンポジウム『DNAバーコーディングで何が出来るか』。..... 自由集会2016:メタバーコーディング・環境DNAバーコーディング解析の技法.
 - ESJ63 自由集会 一覧 - 日本生態学会**
www.esj.ne.jp/meeting/abst/63/jiyu_index.html

参考テキスト

生態学のためのメタバーコーディングと DNA バーコーディング

検索



田辺晶史

生態学のための
メタバーコーディング
とDNAバーコーディング

データ解析の流れ

NGS
data

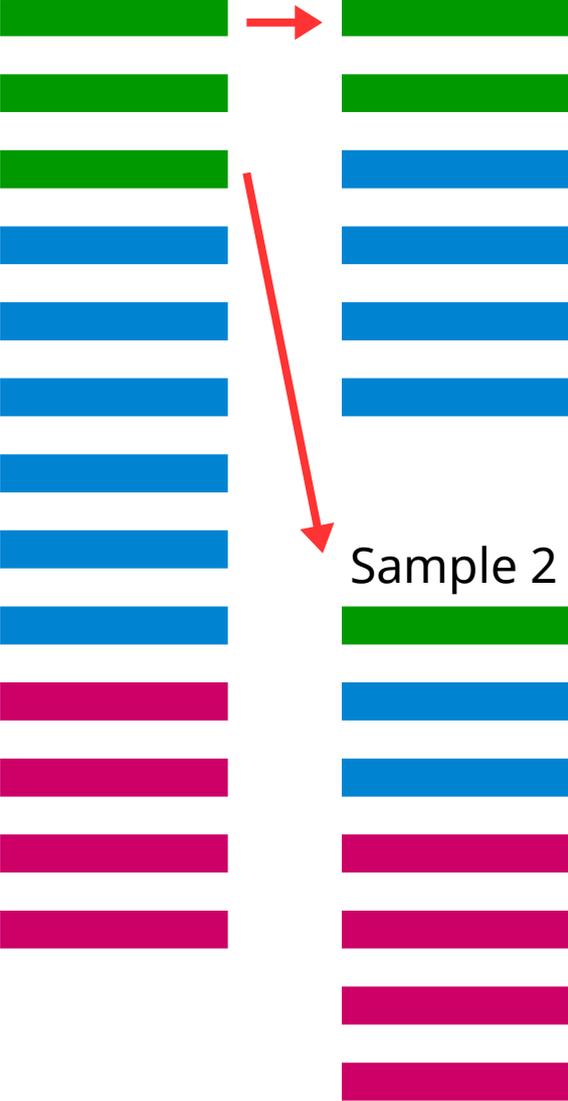


Demultiplexing

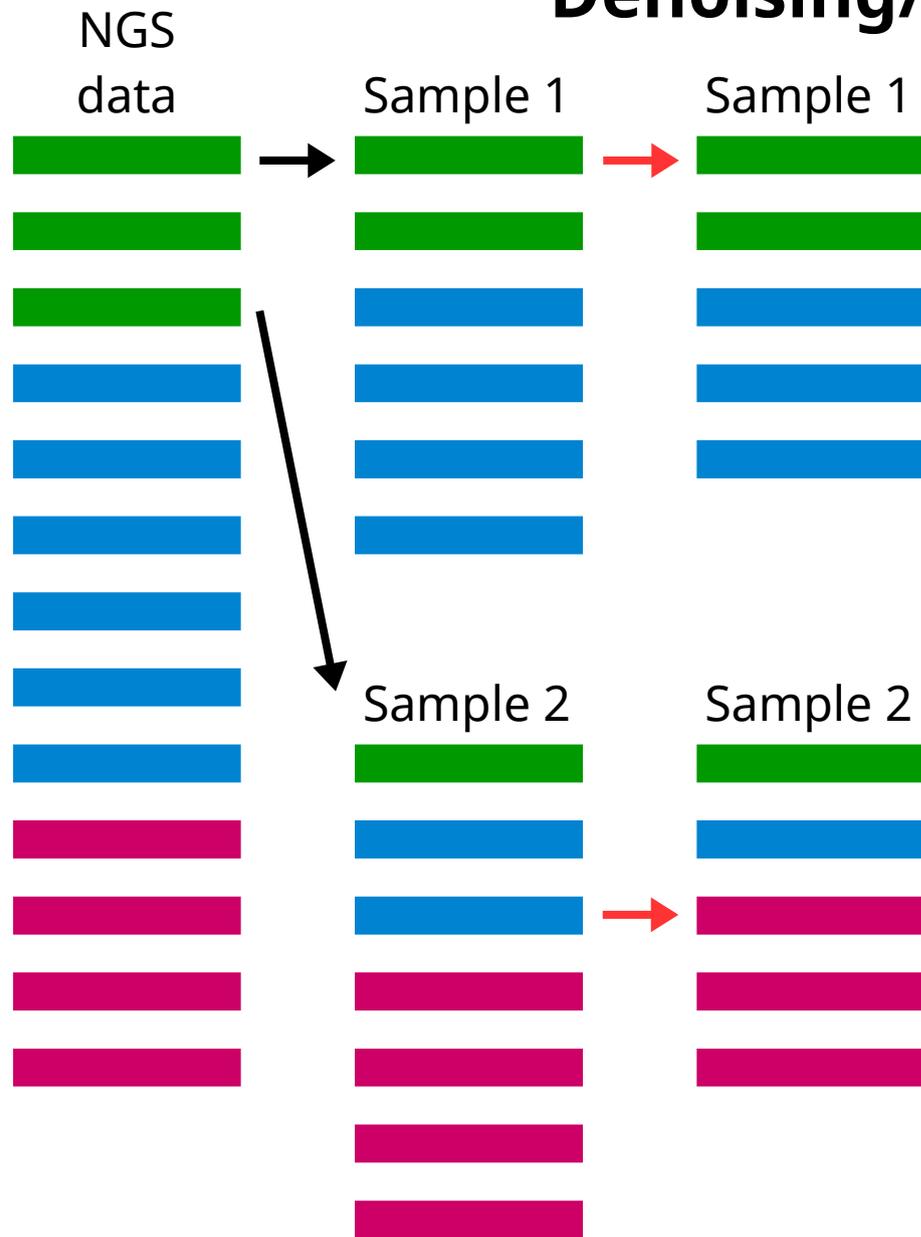
NGS
data

Sample 1

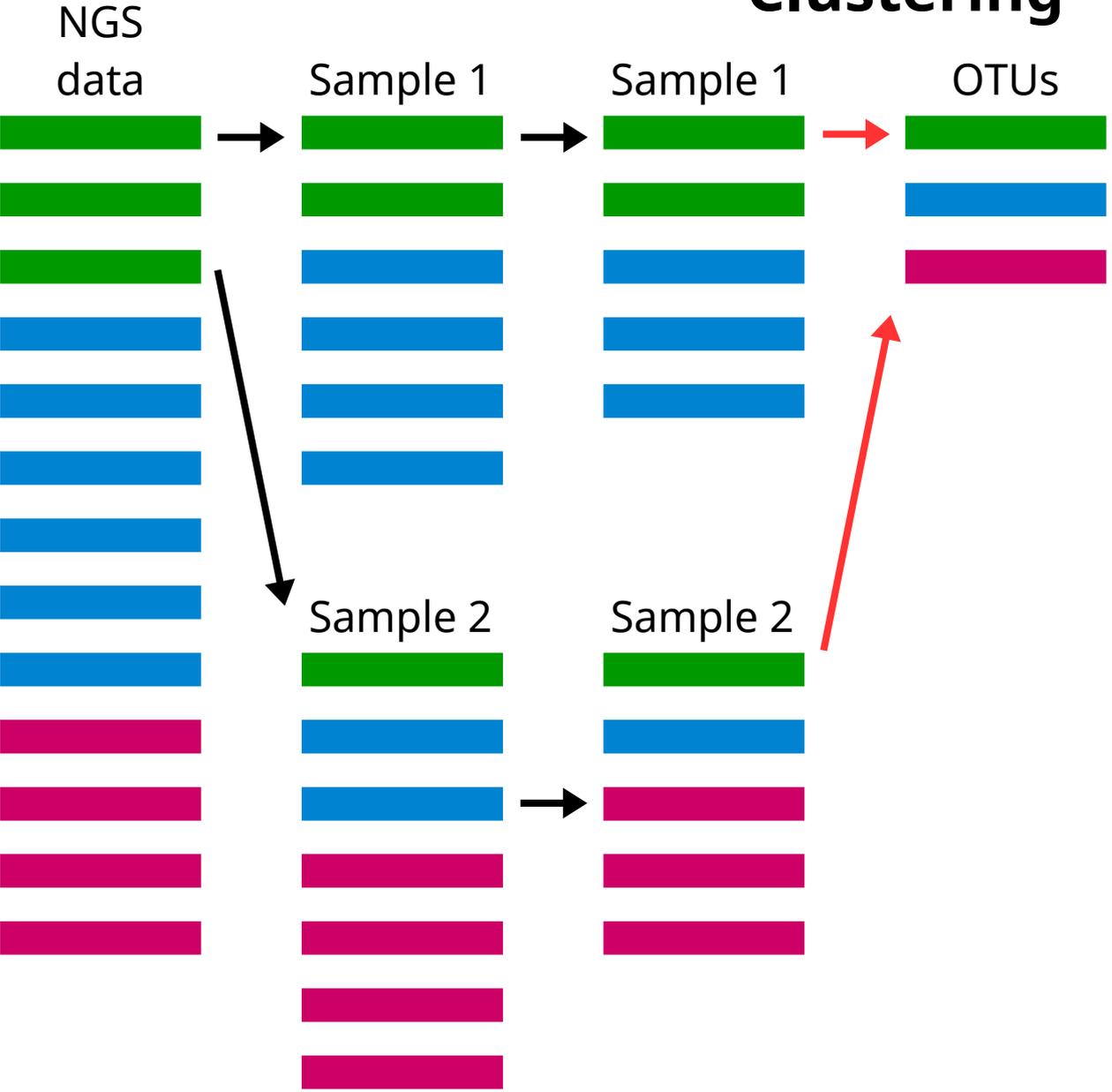
Sample 2



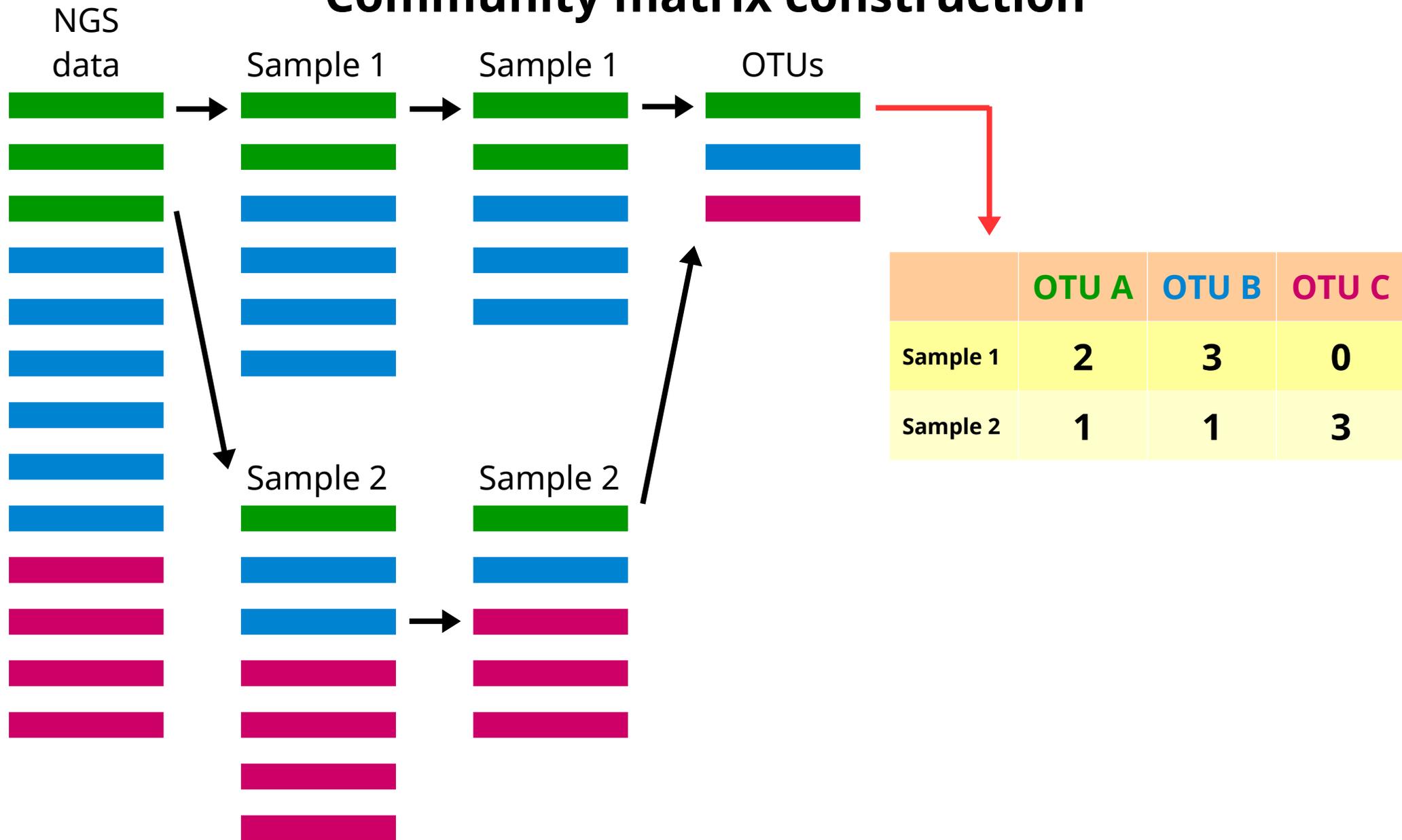
Denoising/Chimera removal



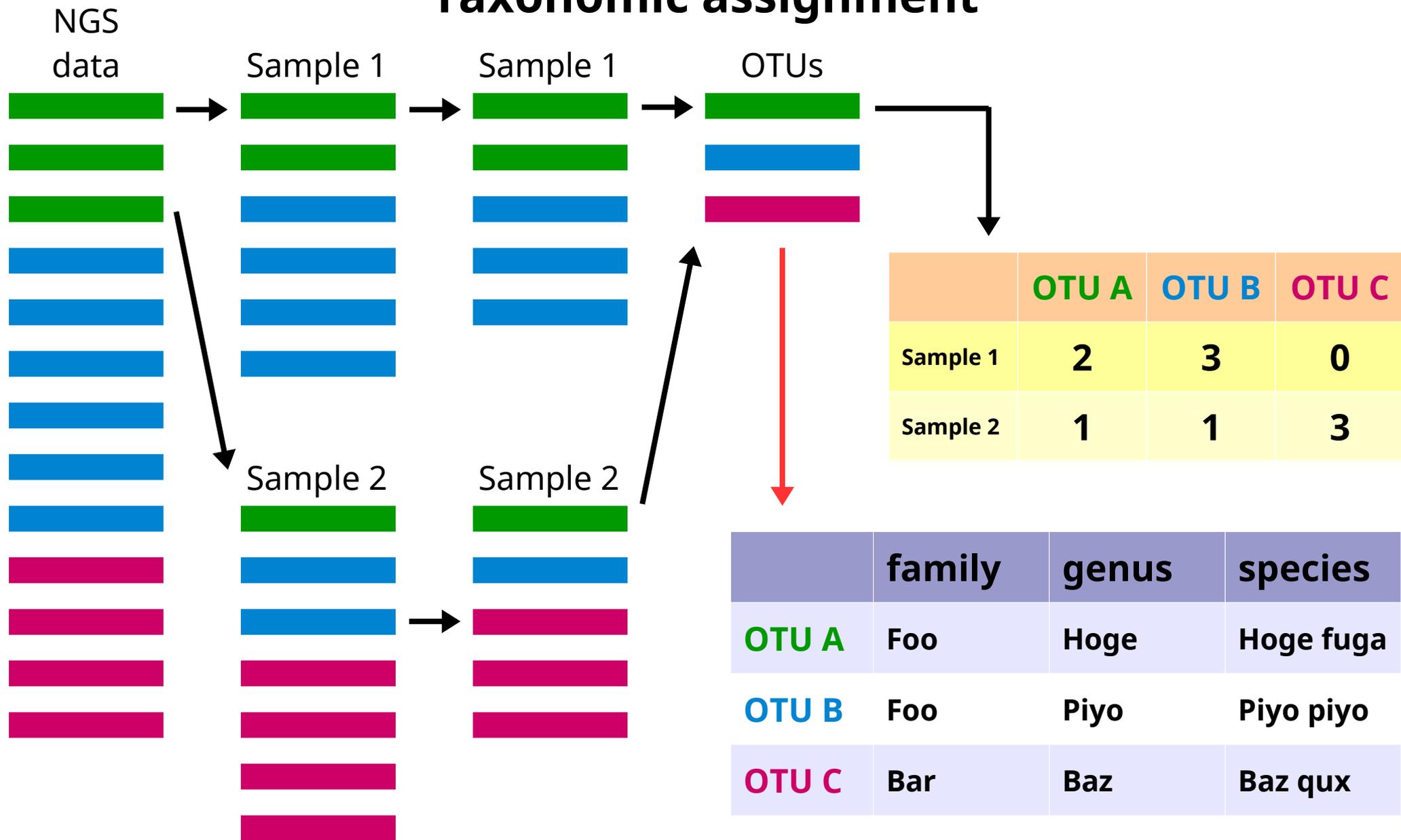
Clustering



Community matrix construction



Taxonomic assignment



Sequence extraction

Sequence extraction

- シーケンサの生データから汎用配列データファイルを作成する

Sequence extraction

- シーケンサの生データから汎用配列データファイルを作成する
- 454/Ion の場合

Sequence extraction

- シーケンサの生データから汎用配列データファイルを作成する
- 454/Ion の場合
 - SFF から sfffile や sff_extract -c で FASTQ を作成

Sequence extraction

- シーケンサの生データから汎用配列データファイルを作成する
- 454/Ion の場合
 - SFF から sfffile や sff_extract -c で FASTQ を作成
 - サンプル識別用タグ配列からデータが始まるようにする

Sequence extraction

- シーケンサの生データから汎用配列データファイルを作成する
- 454/Ion の場合
 - SFF から sfffile や sff_extract -c で FASTQ を作成
 - サンプル識別用タグ配列からデータが始まるようにする
- Illumina の場合

Sequence extraction

- シーケンサの生データから汎用配列データファイルを作成する
- 454/Ion の場合
 - SFF から sfffile や sff_extract -c で FASTQ を作成
 - サンプル識別用タグ配列からデータが始まるようにする
- Illumina の場合
 - BCL から BCL2FASTQ で FASTQ を作成

Sequence extraction

- シーケンサの生データから汎用配列データファイルを作成する
- 454/Ion の場合
 - SFF から sfffile や sff_extract -c で FASTQ を作成
 - サンプル識別用タグ配列からデータが始まるようにする
- Illumina の場合
 - BCL から BCL2FASTQ で FASTQ を作成
 - サンプル識別用タグ配列とプライマーから始まるデータ配列は別のファイルにする

Sequence extraction

- シーケンサの生データから汎用配列データファイルを作成する
- 454/Ion の場合
 - SFF から sfffile や sff_extract -c で FASTQ を作成
 - サンプル識別用タグ配列からデータが始まるようにする
- Illumina の場合
 - BCL から BCL2FASTQ で FASTQ を作成
 - サンプル識別用タグ配列とプライマーから始まるデータ配列は別のファイルにする
 - demultiplex 済のデータを渡される場合はプライマーも除去しておいてもらうこと

Importing FASTQ to Claident pipeline

Importing FASTQ to Claident pipeline

- FASTQ 形式のデータであれば、少し加工すれば取り込むことが可能

Importing FASTQ to Claident pipeline

- FASTQ 形式のデータであれば、少し加工すれば取り込むことが可能
 - DRA の公開データや demultiplex 済で渡された Illumina の配列データを取り込める

Importing FASTQ to Claident pipeline

- FASTQ 形式のデータであれば、少し加工すれば取り込むことが可能
 - DRA の公開データや demultiplex 済で渡された Illumina の配列データを取り込める
- ファイル名は下記の形式にする

Importing FASTQ to Claident pipeline

- FASTQ 形式のデータであれば、少し加工すれば取り込むことが可能
 - DRA の公開データや demultiplex 済で渡された Illumina の配列データを取り込める
- ファイル名は下記の形式にする
 - RunID_TagID_PrimerID.fastq

Importing FASTQ to Claident pipeline

- FASTQ 形式のデータであれば、少し加工すれば取り込むことが可能
 - DRA の公開データや demultiplex 済で渡された Illumina の配列データを取り込める
- ファイル名は下記の形式にする
 - RunID_TagID_PrimerID.fastq
 - TagID や PrimerID がない・わからない場合はダミーを使う

Importing FASTQ to Claident pipeline

- FASTQ 形式のデータであれば、少し加工すれば取り込むことが可能
 - DRA の公開データや demultiplex 済で渡された Illumina の配列データを取り込める
- ファイル名は下記の形式にする
 - RunID_TagID_PrimerID.fastq
 - TagID や PrimerID が無い・わからない場合はダミーを使う
- FASTQ の配列 ID 行を下記の形状に加工すればよい

Importing FASTQ to Claident pipeline

- FASTQ 形式のデータであれば、少し加工すれば取り込むことが可能
 - DRA の公開データや demultiplex 済で渡された Illumina の配列データを取り込める
- ファイル名は下記の形式にする
 - RunID_TagID_PrimerID.fastq
 - TagID や PrimerID が無い・わからない場合はダミーを使う
- FASTQ の配列 ID 行を下記の形状に加工すればよい
 - @UniqueID_RunID_TagID_PrimerID

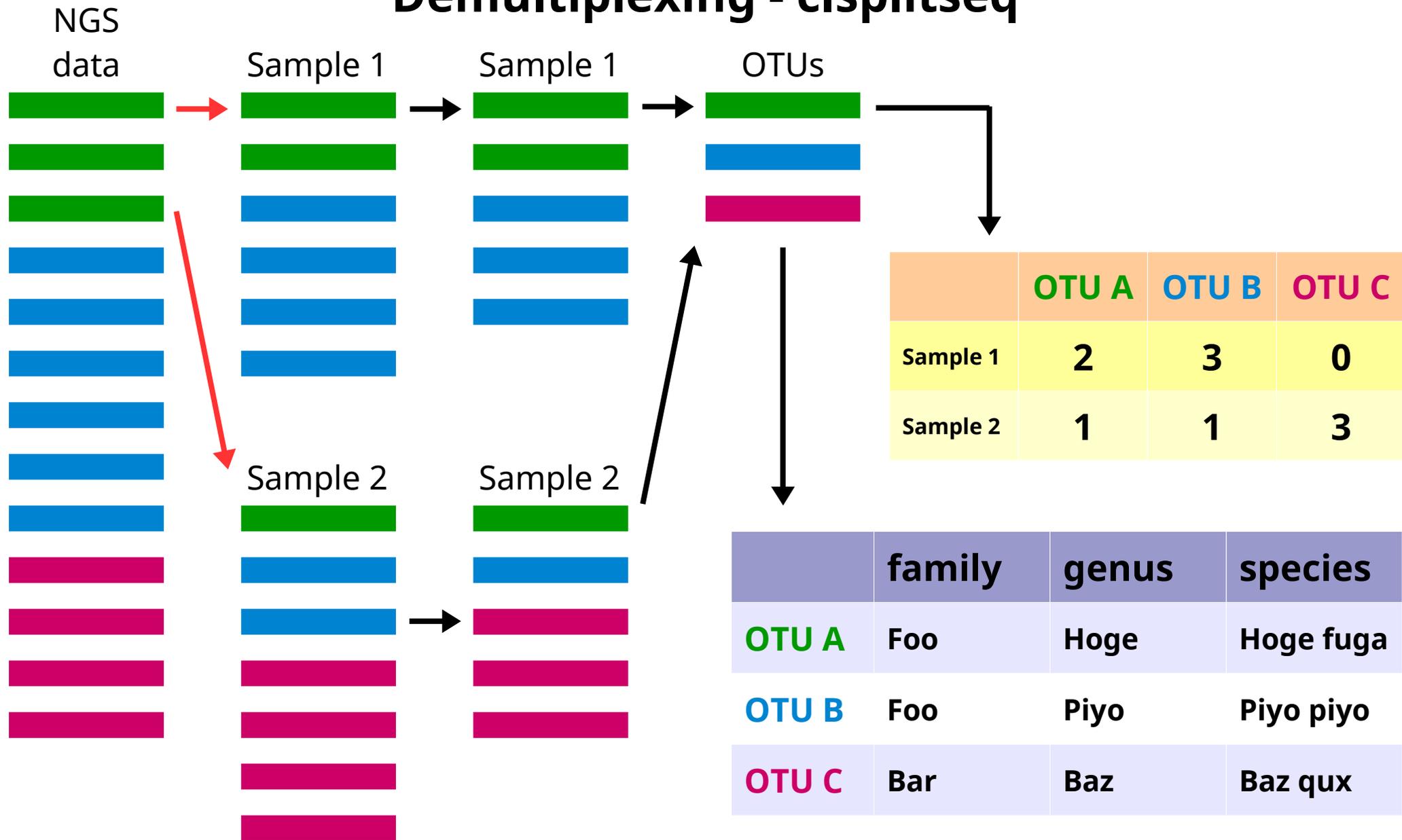
Importing FASTQ to Claident pipeline

- FASTQ 形式のデータであれば、少し加工すれば取り込むことが可能
 - DRA の公開データや demultiplex 済で渡された Illumina の配列データを取り込める
- ファイル名は下記の形式にする
 - RunID_TagID_PrimerID.fastq
 - TagID や PrimerID が無い・わからない場合はダミーを使う
- FASTQ の配列 ID 行を下記の形状に加工すればよい
 - @UniqueID_RunID_TagID_PrimerID
 - TagID や PrimerID が無い・わからない場合はダミーを使う

Importing FASTQ to Claident pipeline

- FASTQ 形式のデータであれば、少し加工すれば取り込むことが可能
 - DRA の公開データや demultiplex 済で渡された Illumina の配列データを取り込める
- ファイル名は下記の形式にする
 - RunID_TagID_PrimerID.fastq
 - TagID や PrimerID がない・わからない場合はダミーを使う
- FASTQ の配列 ID 行を下記の形状に加工すればよい
 - @UniqueID_RunID_TagID_PrimerID
 - TagID や PrimerID がない・わからない場合はダミーを使う
- これらの操作を補助する `climportfastq` を準備中

Demultiplexing - clsplitseq



Demultiplexing - csplitseq

- タグ配列やプライマー配列に基づいて由来サンプルを識別して別ファイルへ分離する

Demultiplexing - clsplitseq

- タグ配列やプライマー配列に基づいて由来サンプルを識別して別ファイルへ分離する
 - FASTA 形式のタグ配列リストとプライマー配列リストが必要

Demultiplexing - clsplitseq

- タグ配列やプライマー配列に基づいて由来サンプルを識別して別ファイルへ分離する
 - FASTA 形式のタグ配列リストとプライマー配列リストが必要
 - single-end、paired-end、single index、dual index によって 1 ~ 4 の FASTQ 形式配列ファイルを与える

Demultiplexing - clsplitseq

- タグ配列やプライマー配列に基づいて由来サンプルを識別して別ファイルへ分離する
 - FASTA 形式のタグ配列リストとプライマー配列リストが必要
 - single-end、paired-end、single index、dual index によって 1 ~ 4 の FASTQ 形式配列ファイルを与える
 - ここで得られる demultiplex 済 primer 除去済ファイルを DDBJ Sequence Read Archive (DRA) に登録する

Concatenating overlapped paired-end

Concatenating overlapped paired-end

- Paired-end シーケンスデータで両側配列にオーバーラップがある場合は連結する

Concatenating overlapped paired-end

- Paired-end シーケンスデータで両側配列にオーバーラップがある場合は連結する
- pear または vsearch --fastq_mergepairs で連結可能

Concatenating overlapped paired-end

- Paired-end シーケンスデータで両側配列にオーバーラップがある場合は連結する
- pear または vsearch --fastq_mergepairs で連結可能
- 連結対象データファイルが 2 組以上ある場合は forなどで連続処理

Concatenating nonoverlapped paired-end

Concatenating nonoverlapped paired-end

- 後述の quality-trimming/filtering 後に下記のように連結

Concatenating nonoverlapped paired-end

- 後述の quality-trimming/filtering 後に下記のように連結
 - <- reverse read - padding - forward read ->

Concatenating nonoverlapped paired-end

- 後述の quality-trimming/filtering 後に下記のように連結
 - <- reverse read - padding - forward read ->
- padding のため identity 過大になるので、clustering 時は offset を指定

Concatenating nonoverlapped paired-end

- 後述の quality-trimming/filtering 後に下記のように連結
 - <- reverse read - padding - forward read ->
- padding のため identity 過大になるので、clustering 時は offset を指定
- clustering 後に padding 配列に基づいて分離

Concatenating nonoverlapped paired-end

- 後述の quality-trimming/filtering 後に下記のように連結
 - <- reverse read - padding - forward read ->
- padding のため identity 過大になるので、clustering 時は offset を指定
- clustering 後に padding 配列に基づいて分離
- 分離したそれぞれで taxonomic assignment して結果を merge

Quality-trimming - cfilterseq

Quality-trimming - cfilterseq

- 配列の末端の QV の低い部分を削る

Quality-trimming - cfilterseq

- 配列の末端の QV の低い部分を削る
 - 閾値は 454 の場合 27 がよいとされている (Kunin et al. 2010)

Quality-trimming - cfilterseq

- 配列の末端の QV の低い部分を削る
 - 閾値は 454 の場合 27 がよいとされている (Kunin et al. 2010)
 - Ion の場合はデータを残すには 20 くらいまで下げざるを得ない

Quality-trimming - cfilterseq

- 配列の末端の QV の低い部分を削る
 - 閾値は 454 の場合 27 がよいとされている (Kunin et al. 2010)
 - Ion の場合はデータを残すには 20 くらいまで下げざるを得ない
 - Illumina の場合は 30 を使うことが多いが連結配列は不要

Quality-trimming - cfilterseq

- 配列の末端の QV の低い部分を削る
 - 閾値は 454 の場合 27 がよいとされている (Kunin et al. 2010)
 - Ion の場合はデータを残すには 20 くらいまで下げざるを得ない
 - Illumina の場合は 30 を使うことが多いが連結配列は不要
- PRINSEQ、Trimmomatic といった定番を使用してもよい

Quality-filtering - cfilterseq

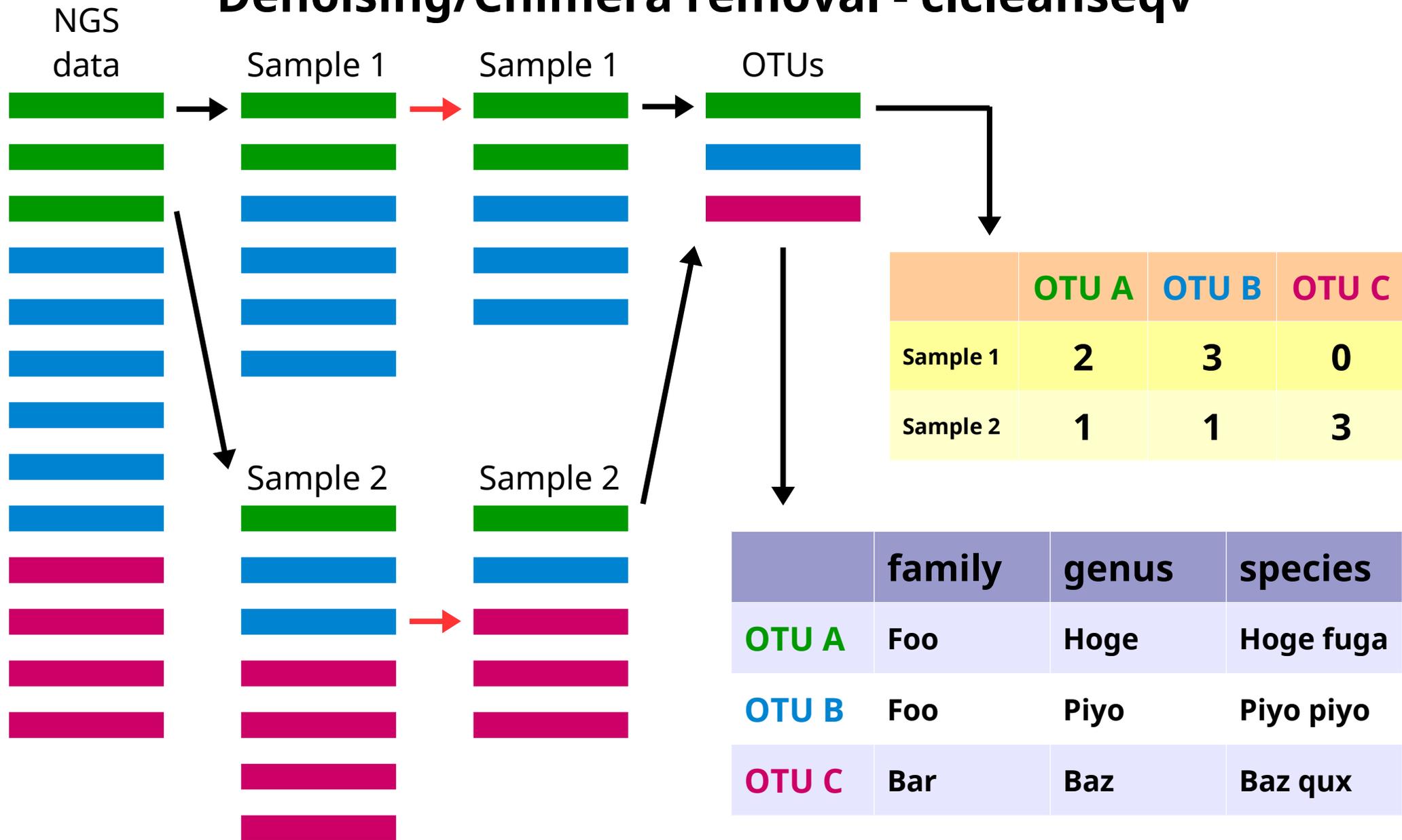
Quality-filtering - cfilterseq

- 長さや QV に基づいて配列を除去する

Quality-filtering - cfilterseq

- 長さや QV に基いて配列を除去する
- `vsearch --fastq_filter` を用いることで、QV から予想される配列中の期待読み間違い数・率に基づく処理も可能

Denoising/Chimera removal - clcleanseqv



Denoising/Chimera removal - clcleanseqv

- 読み間違いゼロの塩基配列がある程度存在すると仮定

Denoising/Chimera removal - clcleanseqv

- 読み間違いゼロの塩基配列がある程度存在すると仮定
- 読み間違いは無作為に入ると仮定

Denoising/Chimera removal - clcleanseqv

- 読み間違いゼロの塩基配列がある程度存在すると仮定
- 読み間違いは無作為に入ると仮定
- 完全一致でまとめたとき、真の配列クラスは配列が多いはず

Denoising/Chimera removal - clcleanseqv

- 読み間違いゼロの塩基配列がある程度存在すると仮定
- 読み間違いは無作為に入ると仮定
- 完全一致でまとめたとき、真の配列クラスターは配列が多いはず
- 配列 n 本未満のクラスターに属する配列を除去する

Denoising/Chimera removal - clcleanseqv

- 読み間違いゼロの塩基配列がある程度存在すると仮定
- 読み間違いは無作為に入ると仮定
- 完全一致でまとめたとき、真の配列クラスターは配列が多いはず
- 配列 n 本未満のクラスターに属する配列を除去する
- 仮定を満たすシーケンサなら、どの機種にも適用可能

Denoising/Chimera removal - clcleanseqv

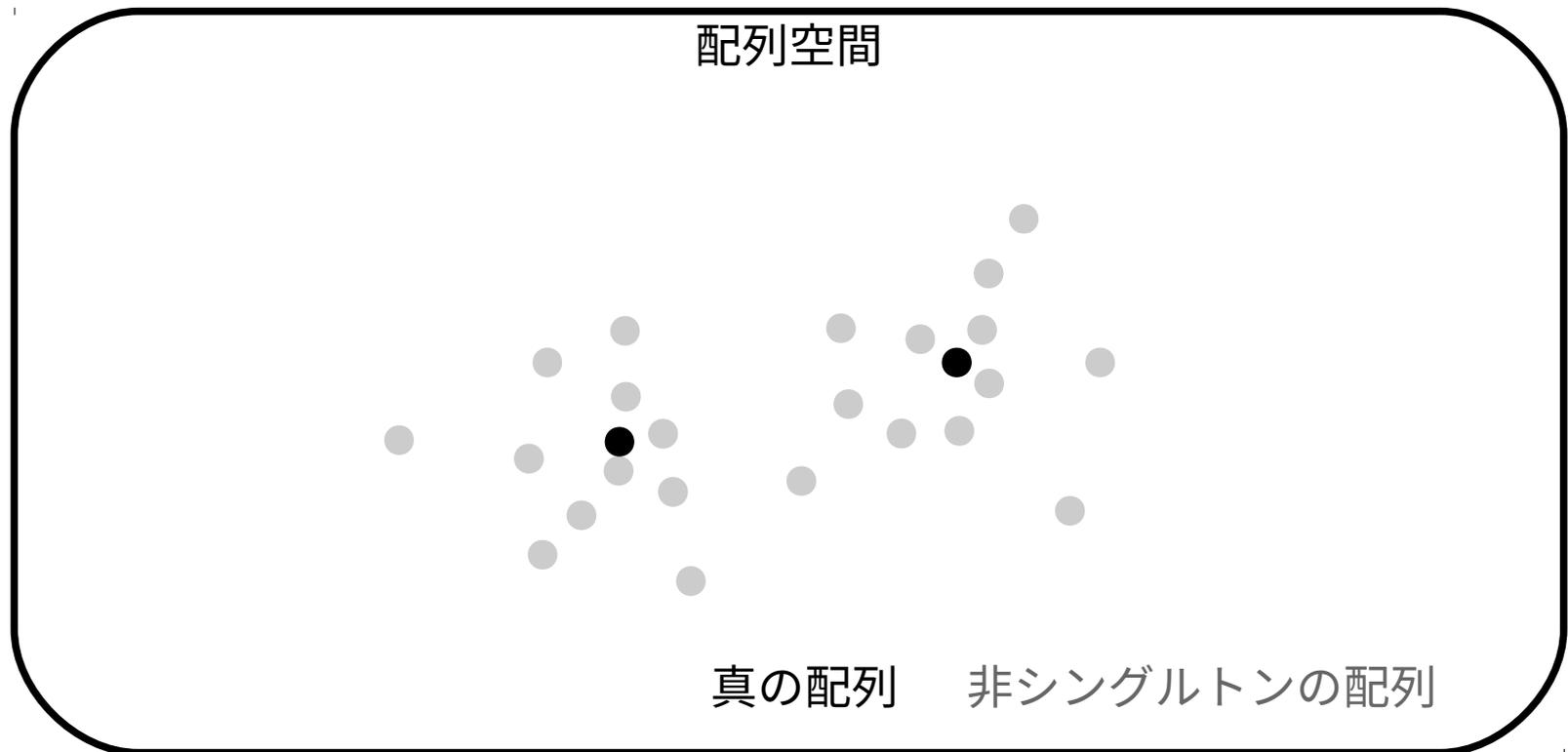
- 読み間違いゼロの塩基配列がある程度存在すると仮定
- 読み間違いは無作為に入ると仮定
- 完全一致でまとめたとき、真の配列クラスターは配列が多いはず
- 配列 n 本未満のクラスターに属する配列を除去する
- 仮定を満たすシーケンサなら、どの機種にも適用可能
- PCR でのコピーミスも検出可能

Denoising/Chimera removal - clcleanseqv

- 読み間違いゼロの塩基配列がある程度存在すると仮定
- 読み間違いは無作為に入ると仮定
- 完全一致でまとめたとき、真の配列クラスタは配列が多いはず
- 配列 n 本未満のクラスタに属する配列を除去する
- 仮定を満たすシーケンサなら、どの機種にも適用可能
- PCR でのコピーミスも検出可能
- ただし、データが減少する、適切な n の値が不明

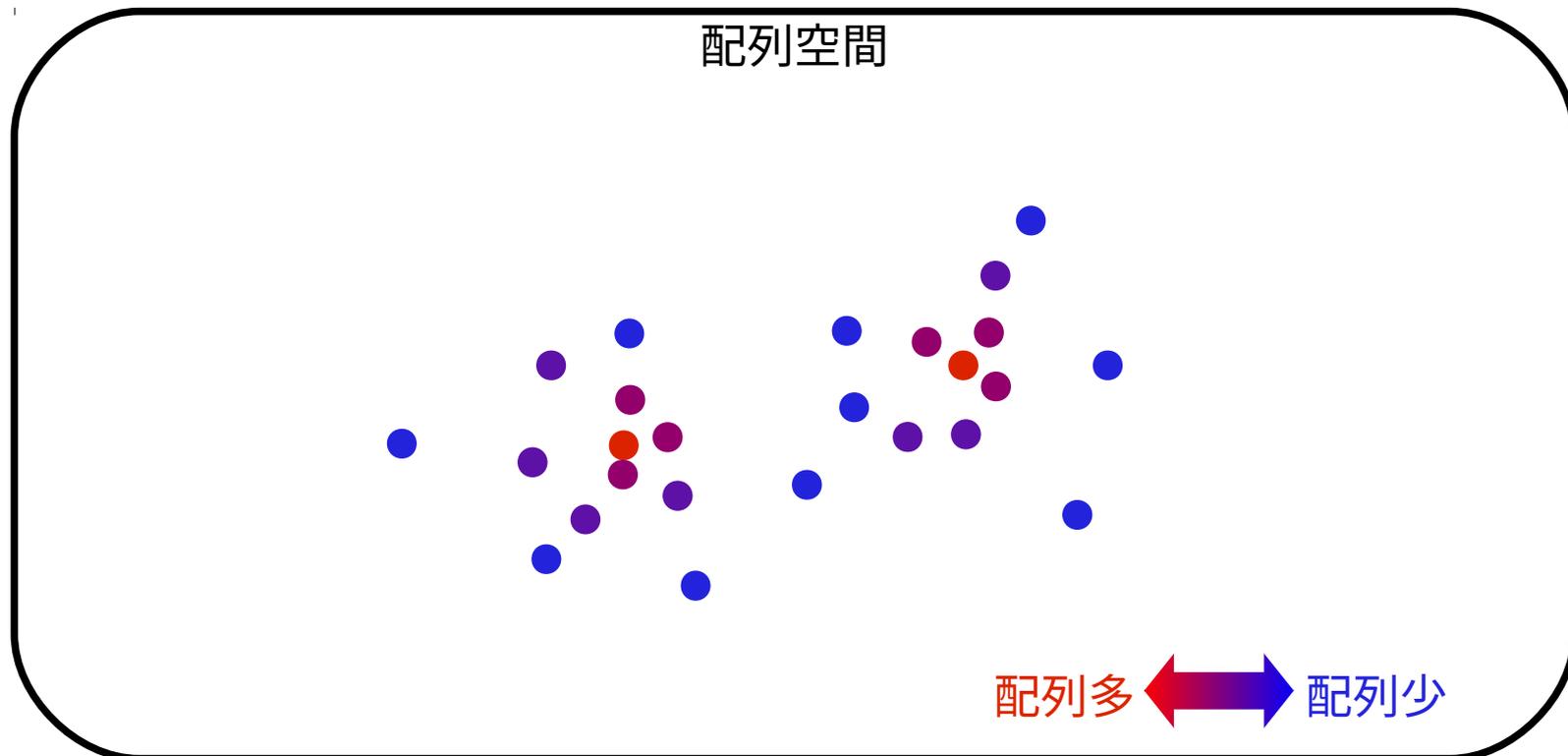
Li et al. (2012) による最適な n の決定方法

Li et al. (2012) による最適な n の決定方法



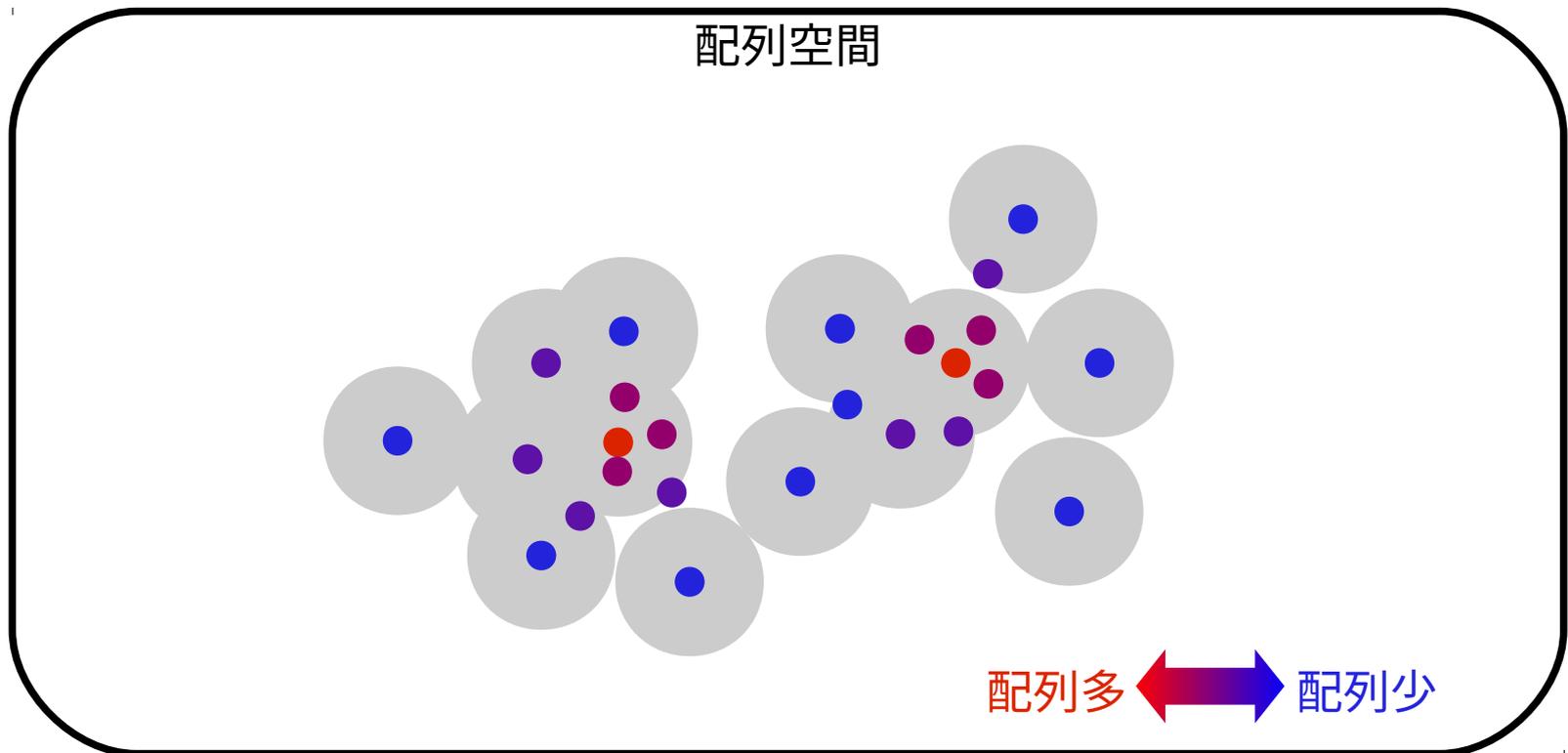
Li et al. (2012) による最適な n の決定方法

1. 完全一致配列をまとめて (赤~青の点)、シングルトン除去



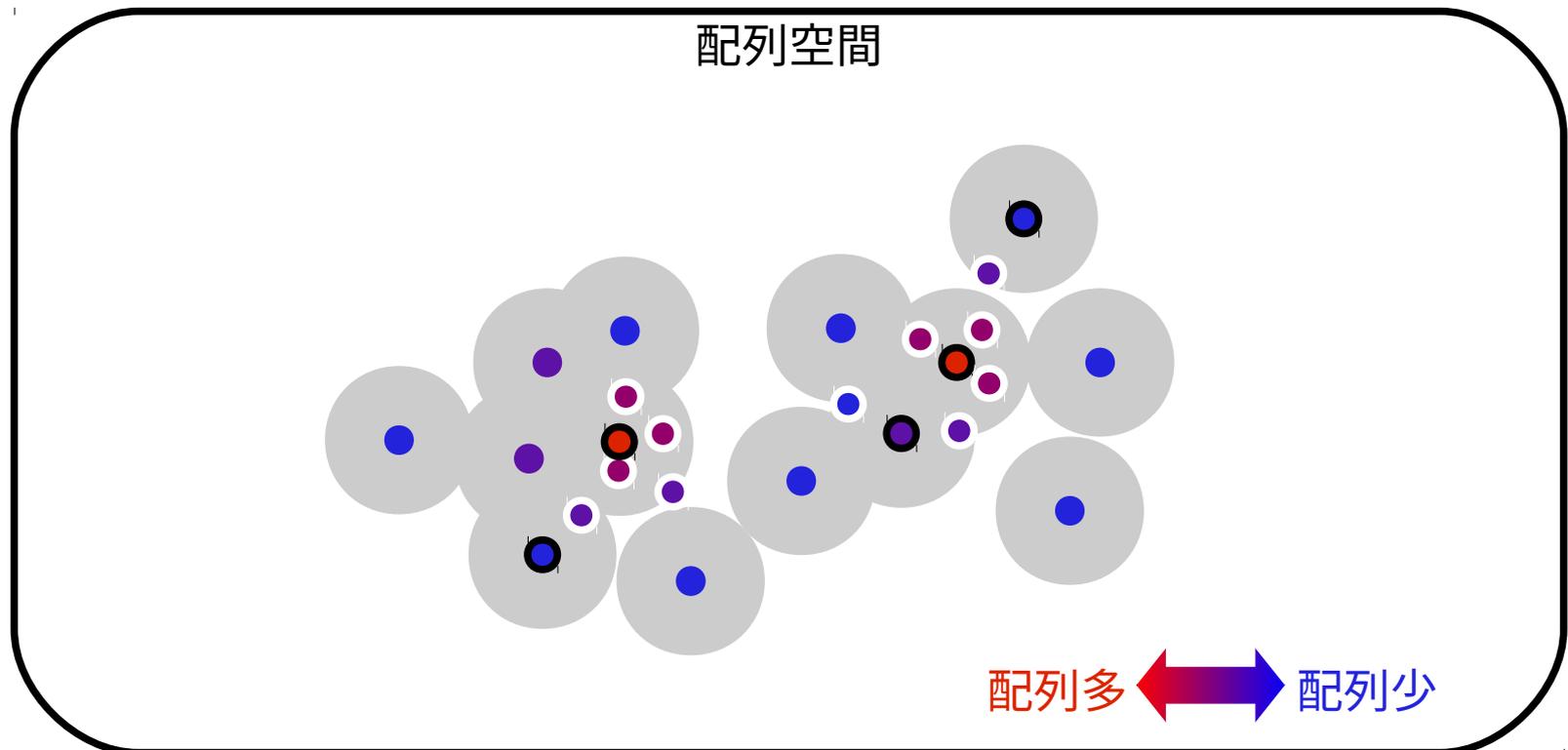
Li et al. (2012) による最適な n の決定方法

1. 完全一致配列をまとめて (赤~青の点)、シングルトン除去
2. さらに 2 塩基不一致許容で配列をまとめる (灰色の円)



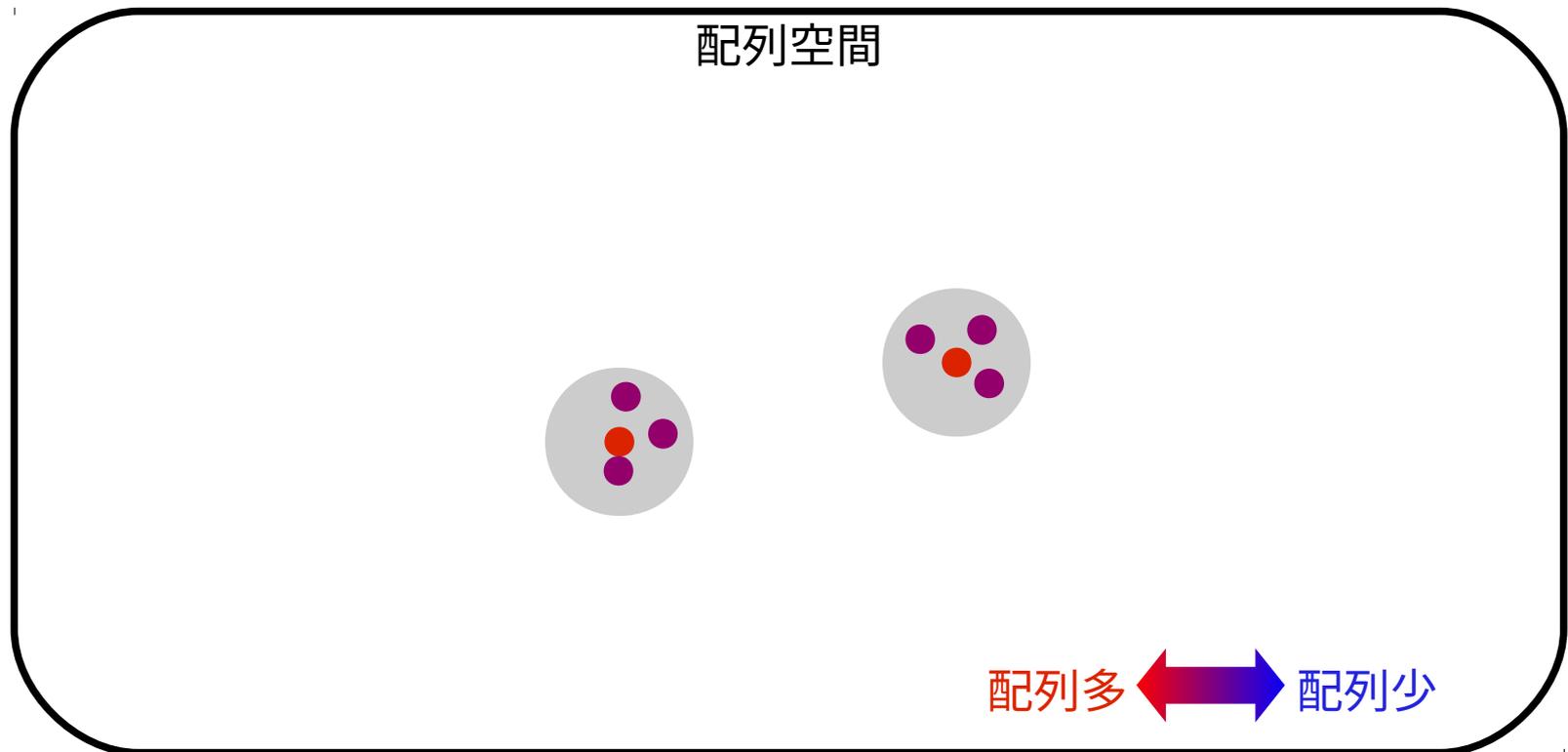
Li et al. (2012) による最適な n の決定方法

1. 完全一致配列をまとめて (赤~青の点)、シングルTON除去
2. さらに 2 塩基不一致許容で配列をまとめる (灰色の円)
3. 灰色の中で配列数 2 位以下のクラスター (白縁の点) の配列数分布の中央値を採用

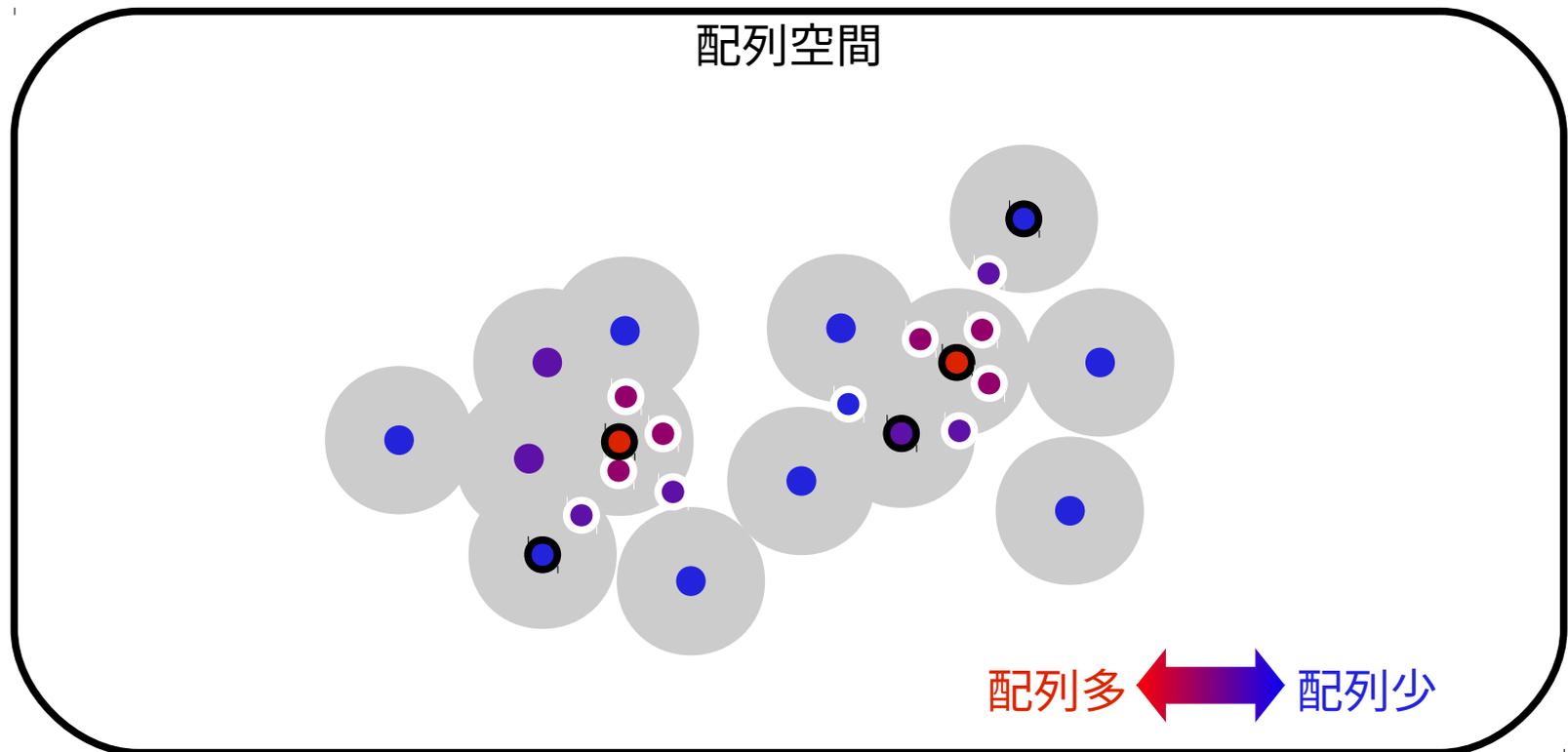


Li et al. (2012) による最適な n の決定方法

1. 完全一致配列をまとめて (赤～青の点)、シングルトン除去
2. さらに 2 塩基不一致許容で配列をまとめる (灰色の円)
3. 灰色の中で配列数 2 位以下のクラスター (白縁の点) の配列数分布の中央値を採用

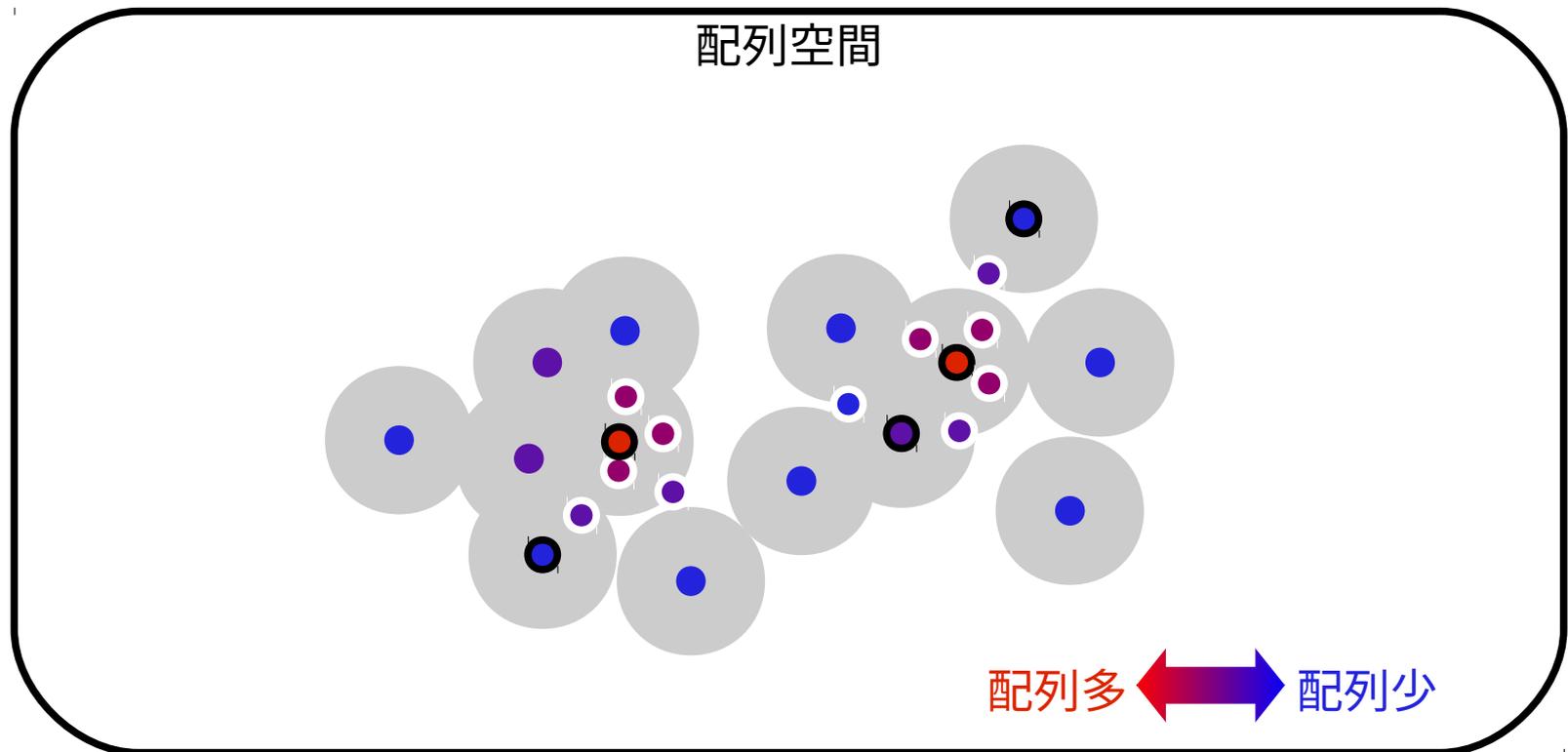


Li et al. (2012) による最適な n の決定方法



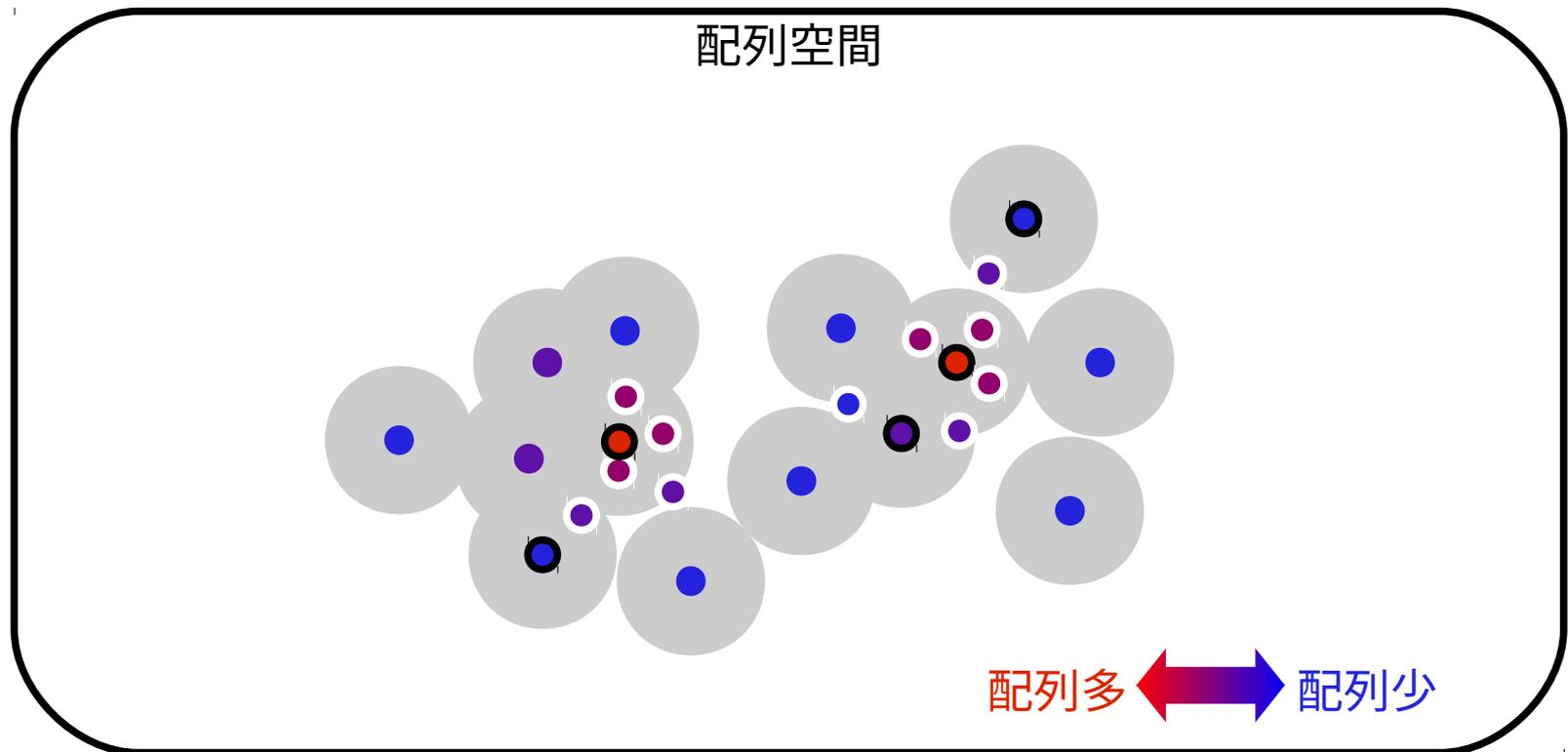
Li et al. (2012) による最適な n の決定方法

- 灰色内で配列数最大のクラスター（黒縁の点）は真の配列である可能性が高い



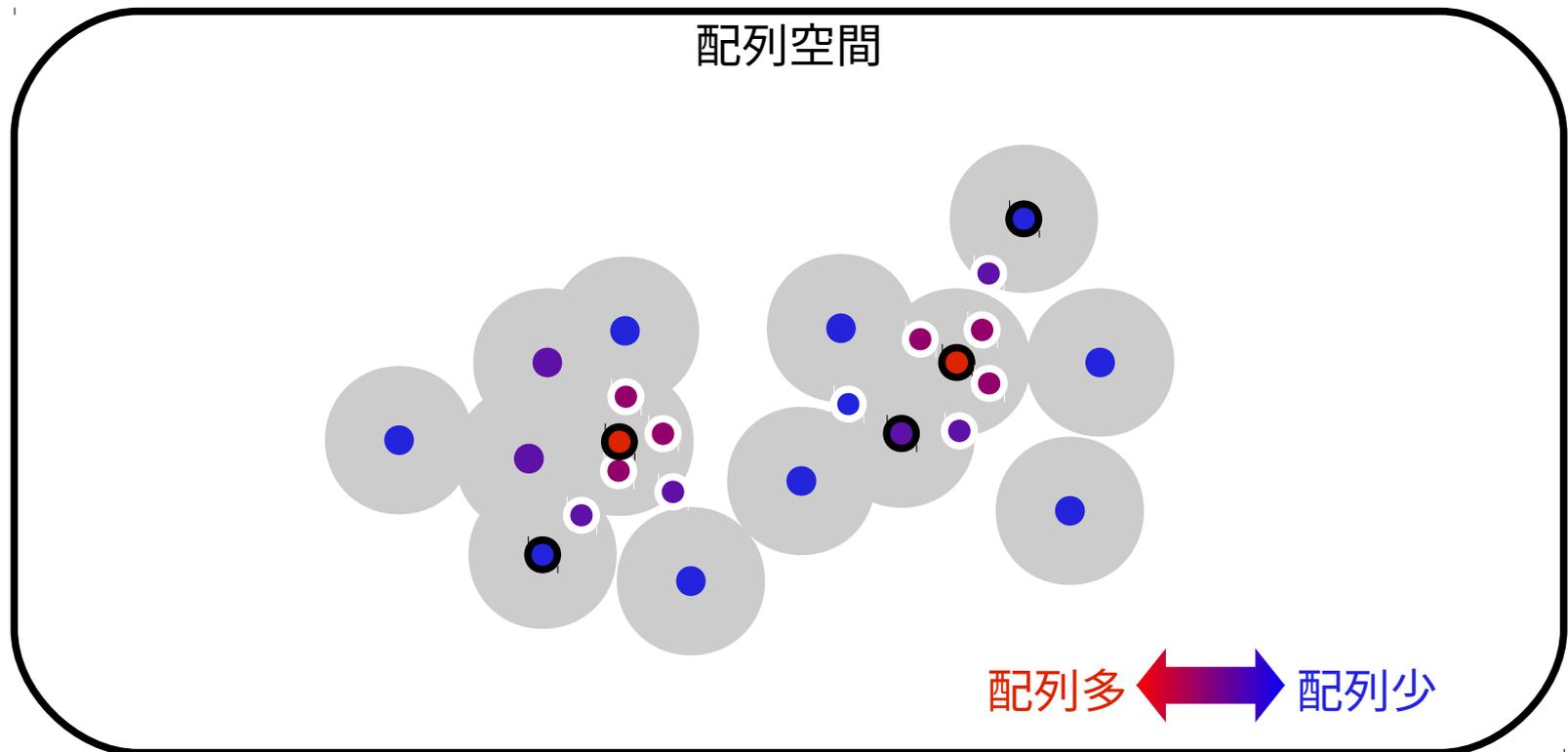
Li et al. (2012) による最適な n の決定方法

- 灰色内で配列数 2 位以下のクラスター (白縁の点) は「読み間違いはあるが、その数は 2 塩基以下」の配列である可能性が高い



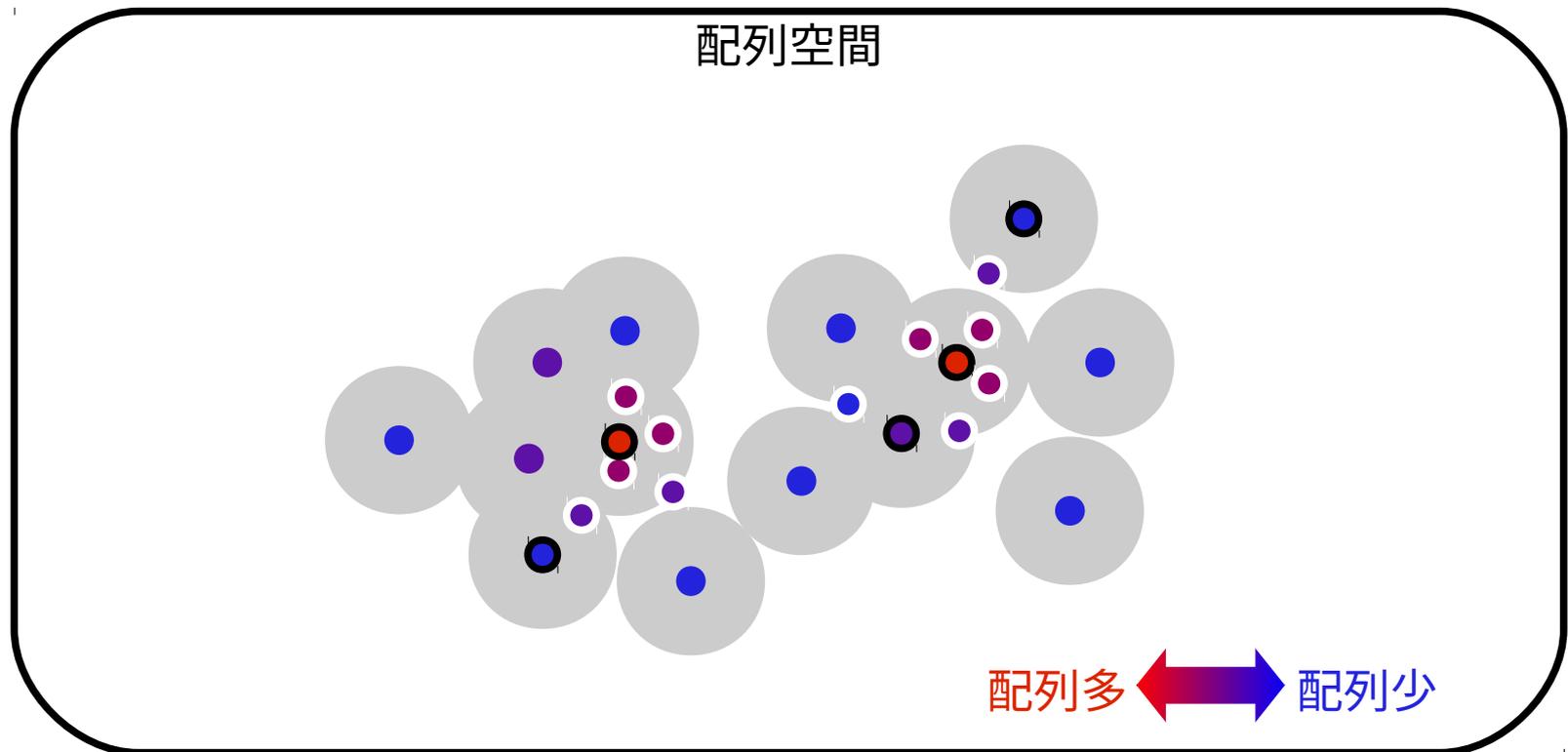
Li et al. (2012) による最適な n の決定方法

- 読み間違いの多いクラスターほど配列が少ないと仮定できる



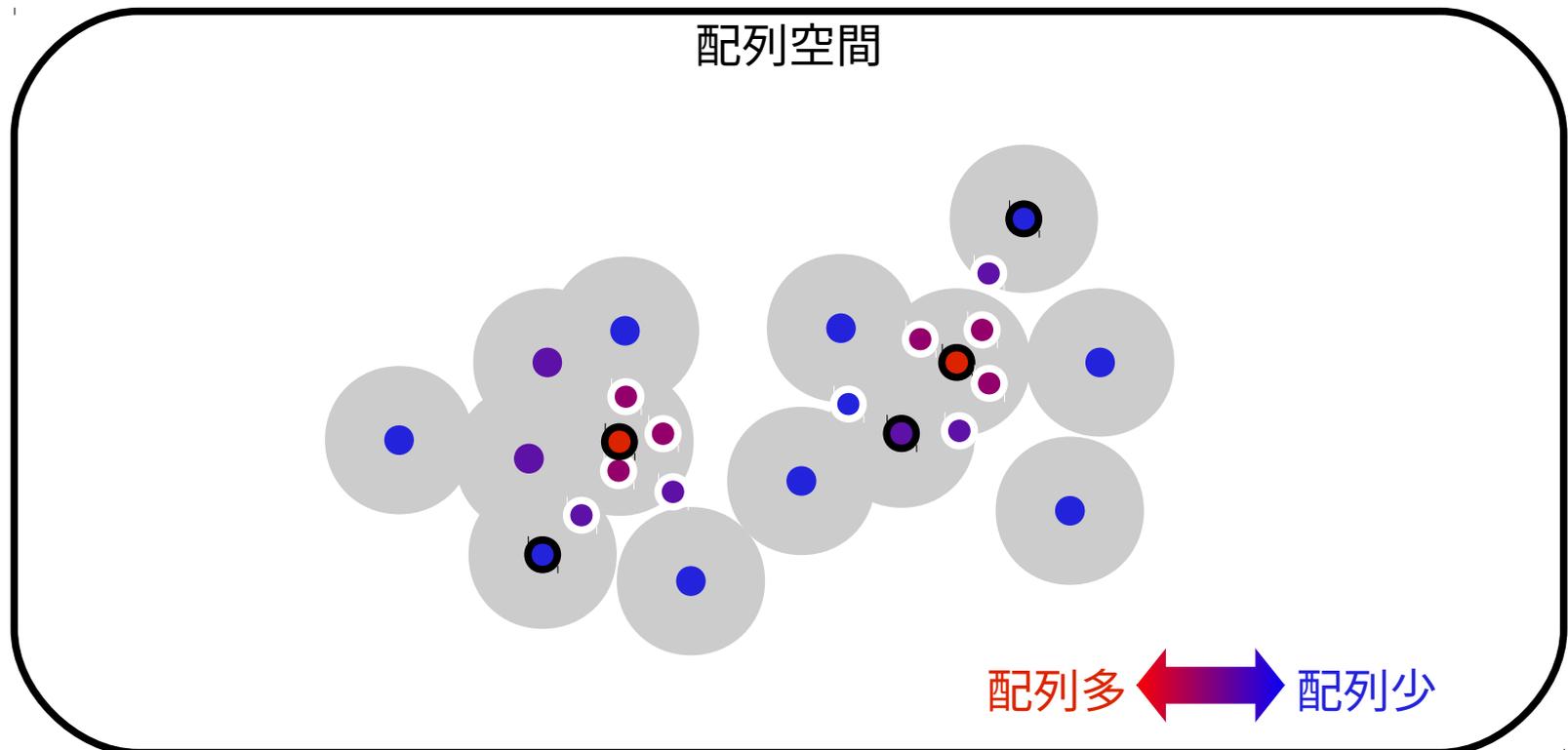
Li et al. (2012) による最適な n の決定方法

- 読み間違いの多いクラスタほど配列が少ないと仮定できる
- 「3塩基以上の読み間違いがある」クラスタ(縁なしの点)は「読み間違いはあるが2塩基以下」のクラスタ(白縁の点)よりも配列が少ないはず



Li et al. (2012) による最適な n の決定方法

- 読み間違いの多いクラスタほど配列が少ないと仮定できる
- 「3塩基以上の読み間違いがある」クラスタ (縁なしの点) の多くは「読み間違いはあるが2塩基以下」のクラスタ (白縁の点) の配列数分布の中央値より小さい可能性が高い



Li et al. (2012) と ccleanseqv の違い

Li et al. (2012) と clcleanseqv の違い

- CD-HIT の代わりに VSEARCH を使用

Li et al. (2012) と clcleanseqv の違い

- CD-HIT の代わりに VSEARCH を使用
- セカンダリクラスタリングは「1 塩基不一致許容」がデフォルト設定

Li et al. (2012) と clcleanseqv の違い

- CD-HIT の代わりに VSEARCH を使用
- セカンダリクラスタリングは「1 塩基不一致許容」がデフォルト設定
- プライマリ、セカンダリ両クラスタリングの不一致許容量は変更可能

Li et al. (2012) と ccleanseqv の違い

- CD-HIT の代わりに VSEARCH を使用
- セカンダリクラスタリングは「1 塩基不一致許容」がデフォルト設定
- プライマリ、セカンダリ両クラスタリングの不一致許容量は変更可能
- n の値として採用するクラスタサイズの分位点も変更可能

Li et al. (2012) と ccleanseqv の違い

- CD-HIT の代わりに VSEARCH を使用
- セカンダリクラスタリングは「1 塩基不一致許容」がデフォルト設定
- プライマリ、セカンダリ両クラスタリングの不一致許容量は変更可能
- n の値として採用するクラスタサイズの分位点も変更可能
- 読み間違いが多い場合は以下のように変更すると良い

Li et al. (2012) と ccleanseqv の違い

- CD-HIT の代わりに VSEARCH を使用
- セカンダリクラスタリングは「1 塩基不一致許容」がデフォルト設定
- プライマリ、セカンダリ両クラスタリングの不一致許容量は変更可能
- n の値として採用するクラスタサイズの分位点も変更可能
- 読み間違いが多い場合は以下のように変更すると良い
 - プライマリクラスタリングでは 1 塩基不一致許容

Li et al. (2012) と cldcleanseqv の違い

- CD-HIT の代わりに VSEARCH を使用
- セカンダリクラスタリングは「1 塩基不一致許容」がデフォルト設定
- プライマリ、セカンダリ両クラスタリングの不一致許容量は変更可能
- n の値として採用するクラスタサイズの分位点も変更可能
- 読み間違いが多い場合は以下のように変更すると良い
 - プライマリクラスタリングでは 1 塩基不一致許容
 - セカンダリクラスタリングでは 3 塩基不一致許容

Estimating expected error by VSEARCH

Estimating expected error by VSEARCH

- VSEARCH で QV に基いて期待読み間違い数を推定可能

Estimating expected error by VSEARCH

- VSEARCH で QV に基いて期待読み間違い数を推定可能
 - `vsearch --fastq_eestats FILENAME --output FILENAME`

Estimating expected error by VSEARCH

- VSEARCH で QV に基いて期待読み間違い数を推定可能
 - `vsearch --fastq_eestats FILENAME --output FILENAME`
- この推定結果に基いて denoising のプライマリクラスタリング、セカンダリクラスタリングの不一致許容量を決定する

Denoising/Chimera removal by PCR replication

Denoising/Chimera removal by PCR replication

採集

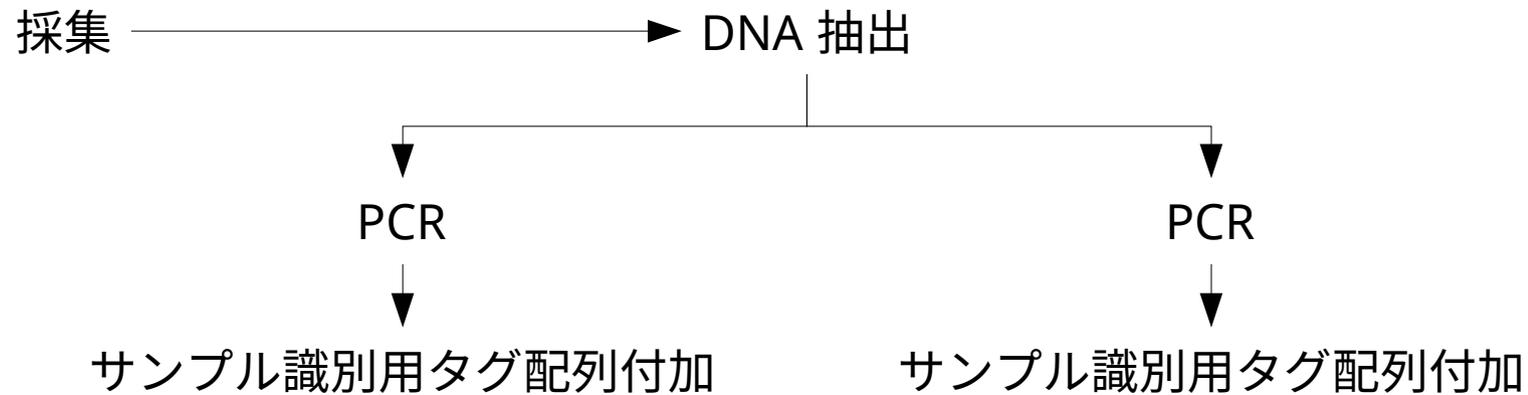
Denoising/Chimera removal by PCR replication

採集 —————▶ DNA 抽出

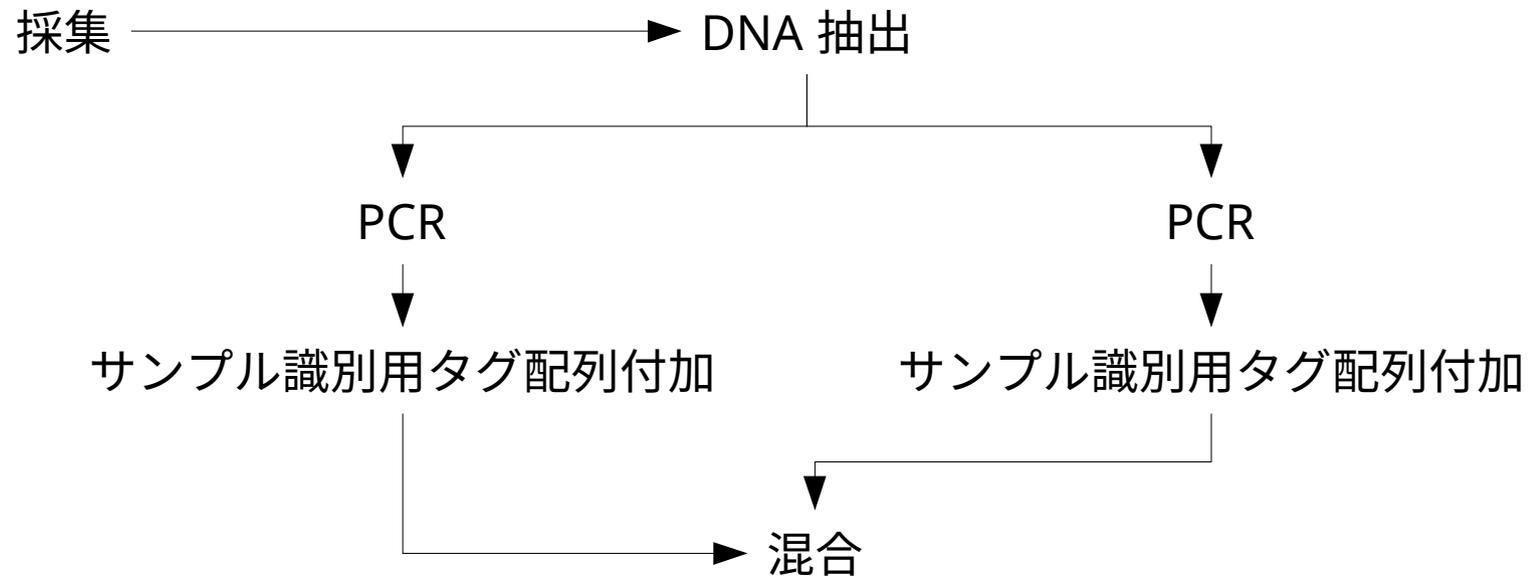
Denoising/Chimera removal by PCR replication



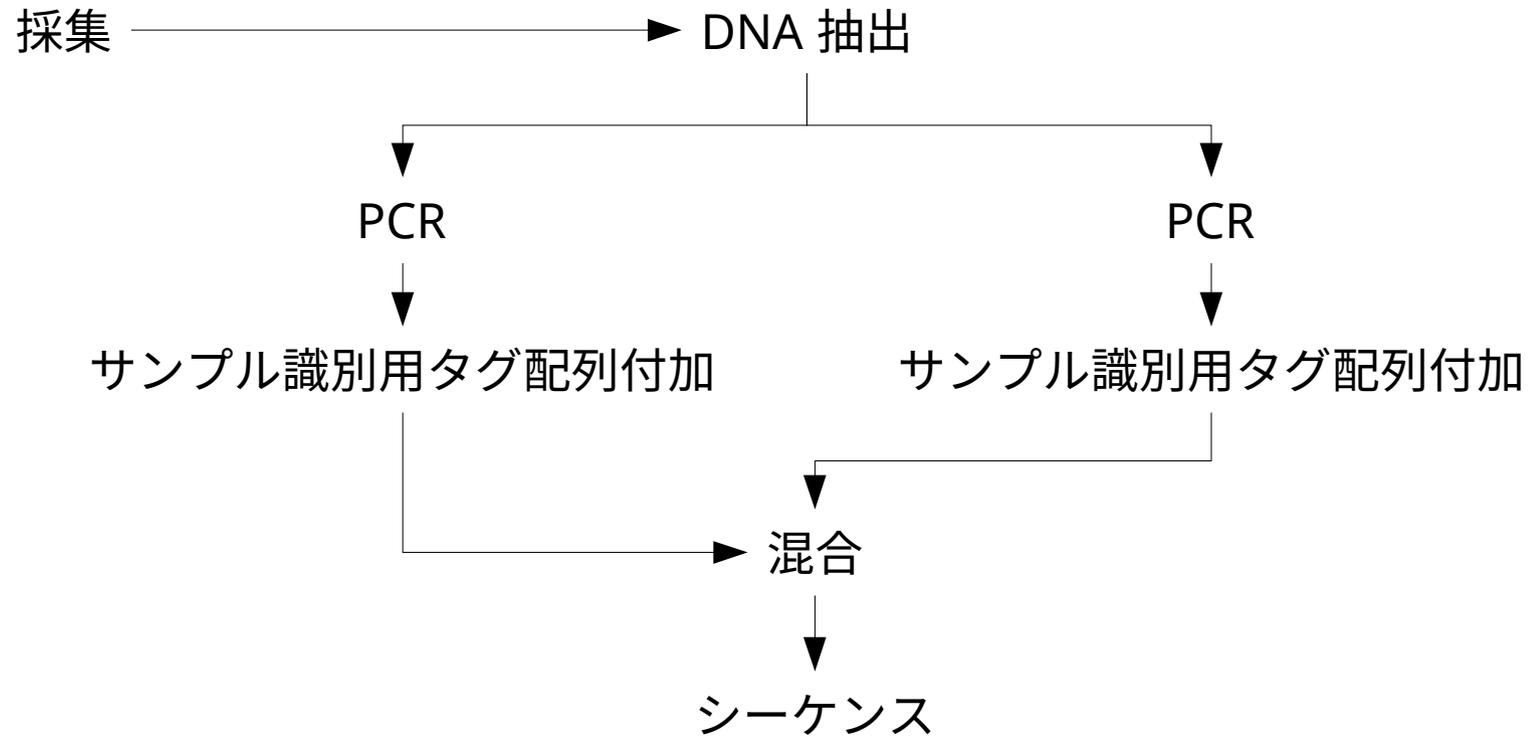
Denoising/Chimera removal by PCR replication



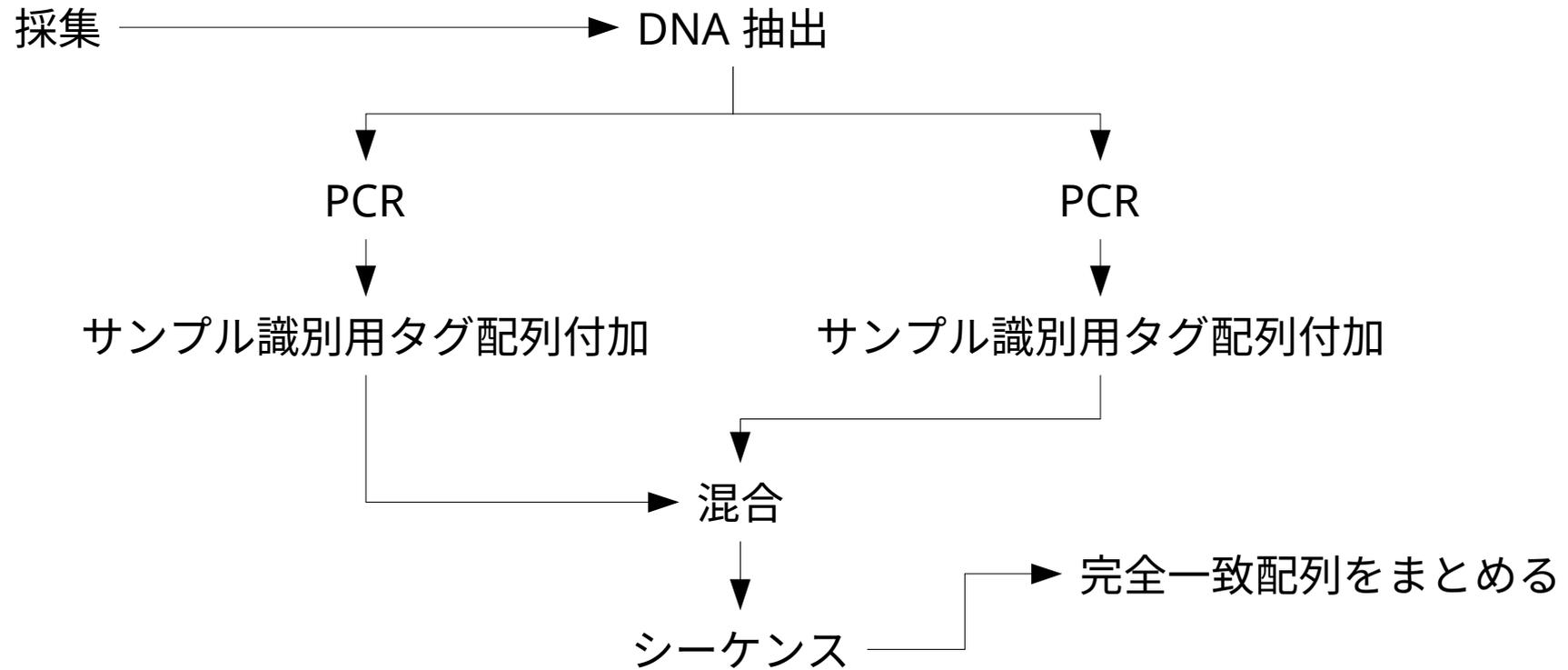
Denoising/Chimera removal by PCR replication



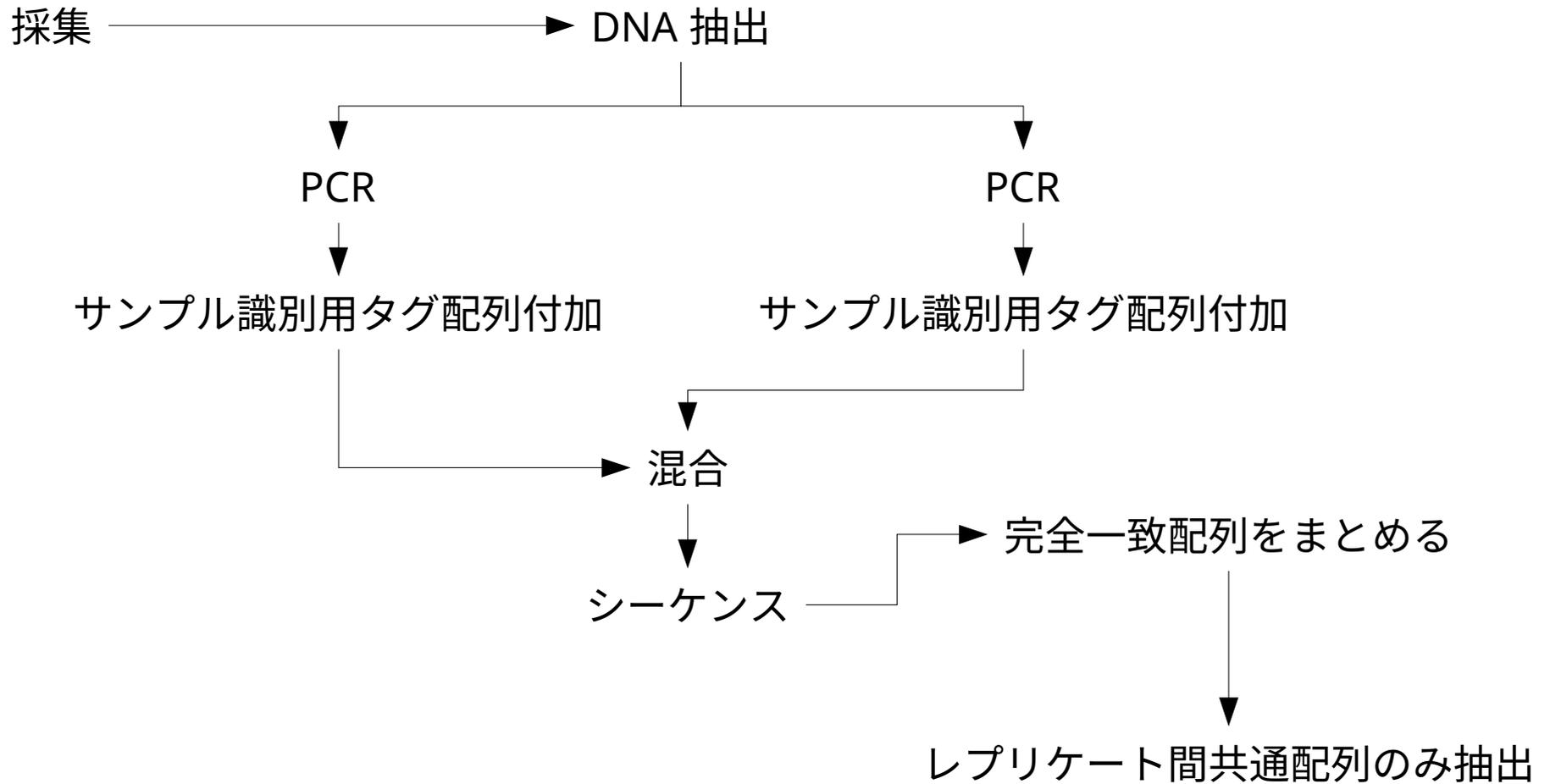
Denoising/Chimera removal by PCR replication



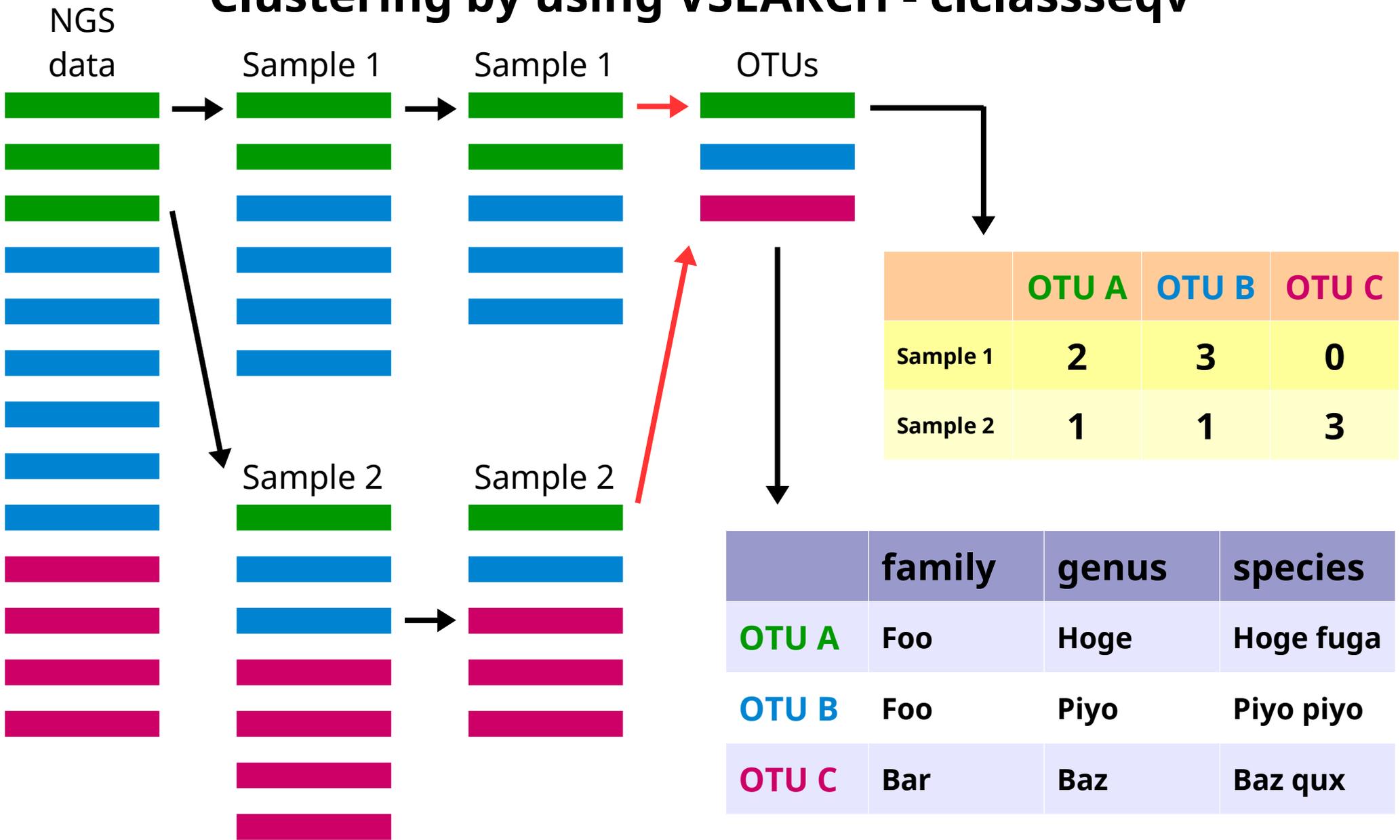
Denoising/Chimera removal by PCR replication



Denoising/Chimera removal by PCR replication



Clustering by using VSEARCH - clclasseqv

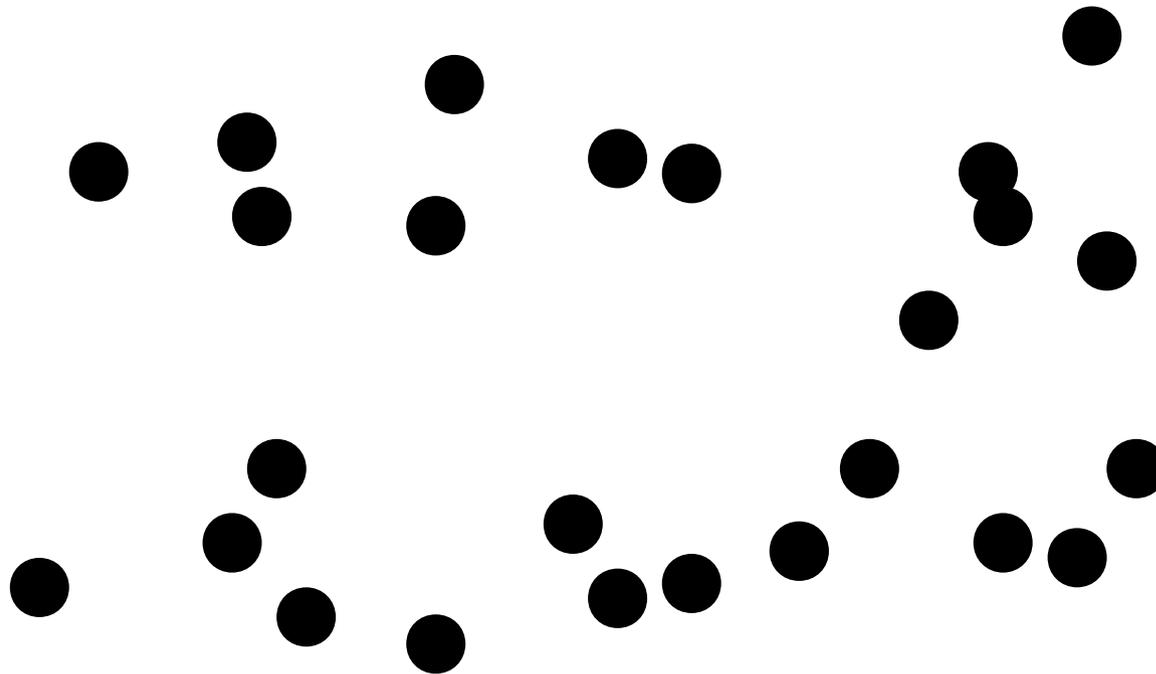


Clustering by using VSEARCH - clclasseqv

- 残っている配列を VSEARCH に渡してクラスタリング

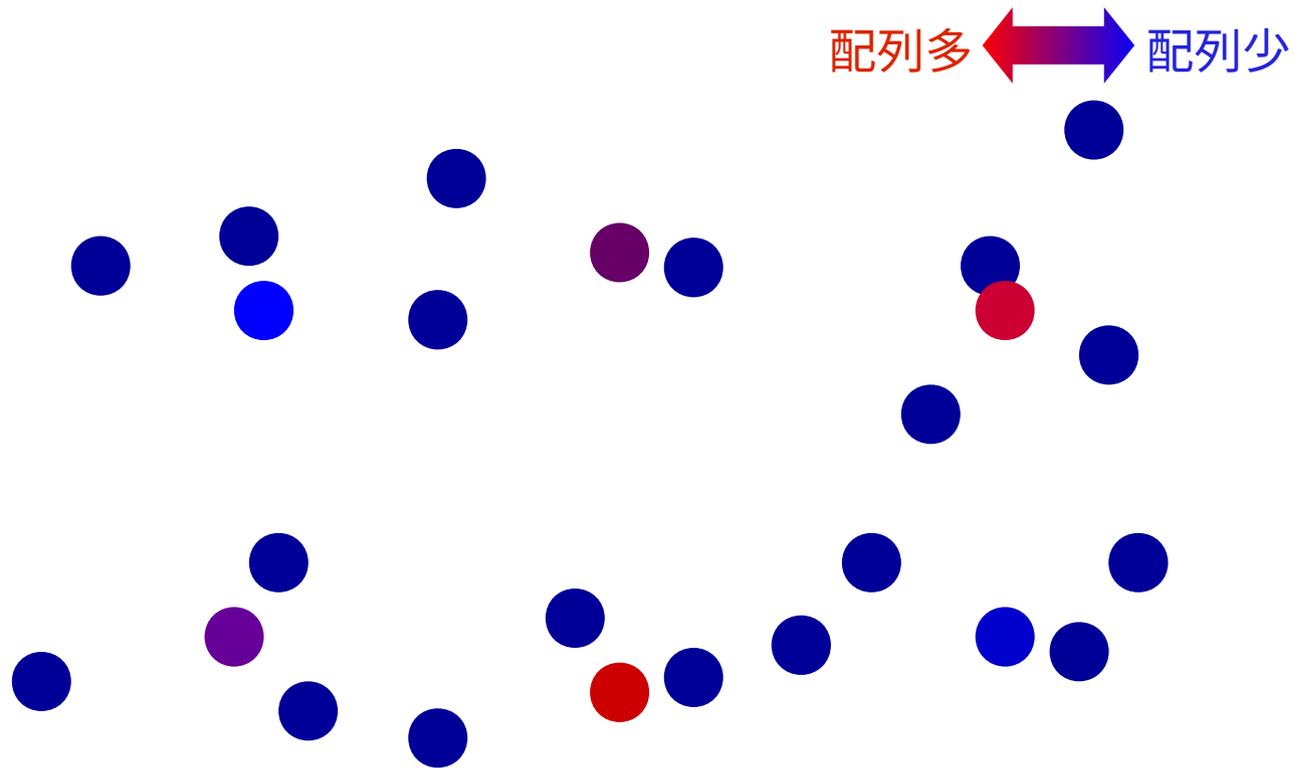
Clustering by using VSEARCH - clclasseqv

- 残っている配列を VSEARCH に渡してクラスタリング
 - 完全一致配列をクラスタリングしてコピー数をカウント



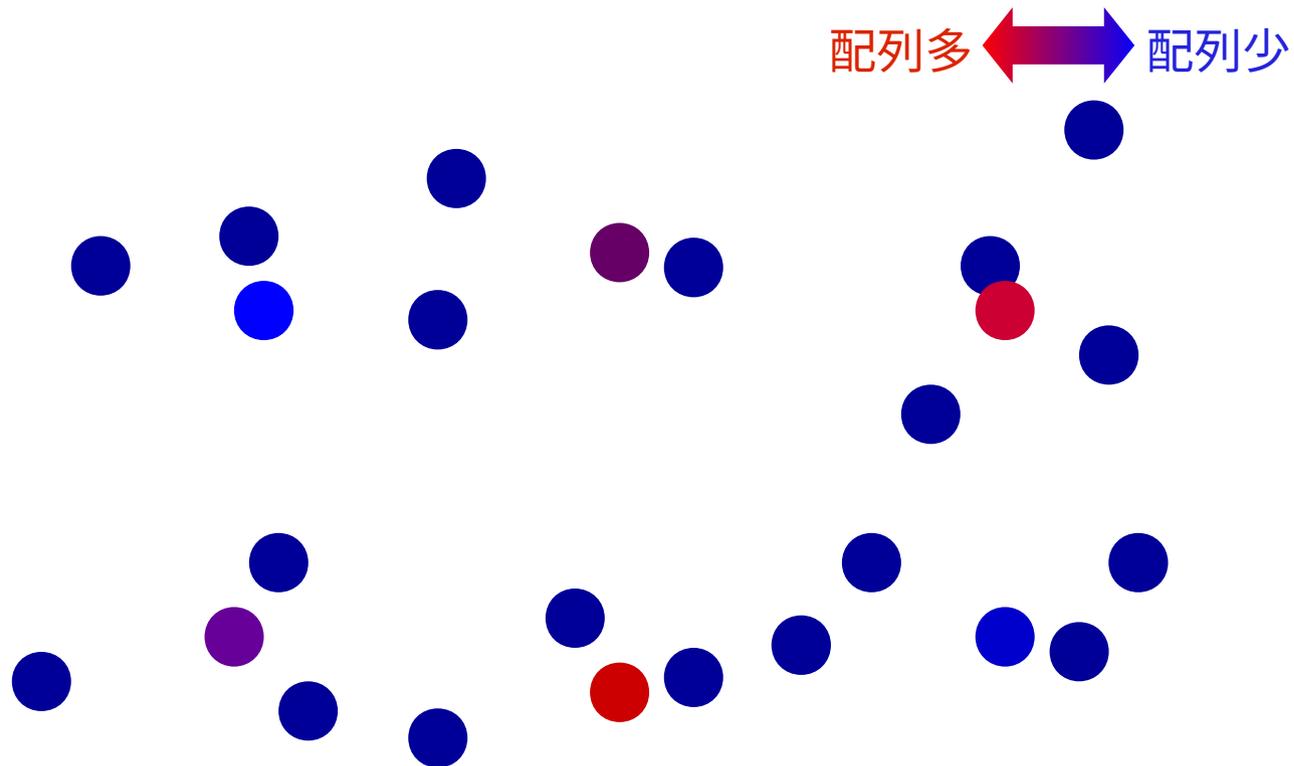
Clustering by using VSEARCH - clclasseqv

- 残っている配列を VSEARCH に渡してクラスタリング
 - 完全一致配列をクラスタリングしてコピー数をカウント



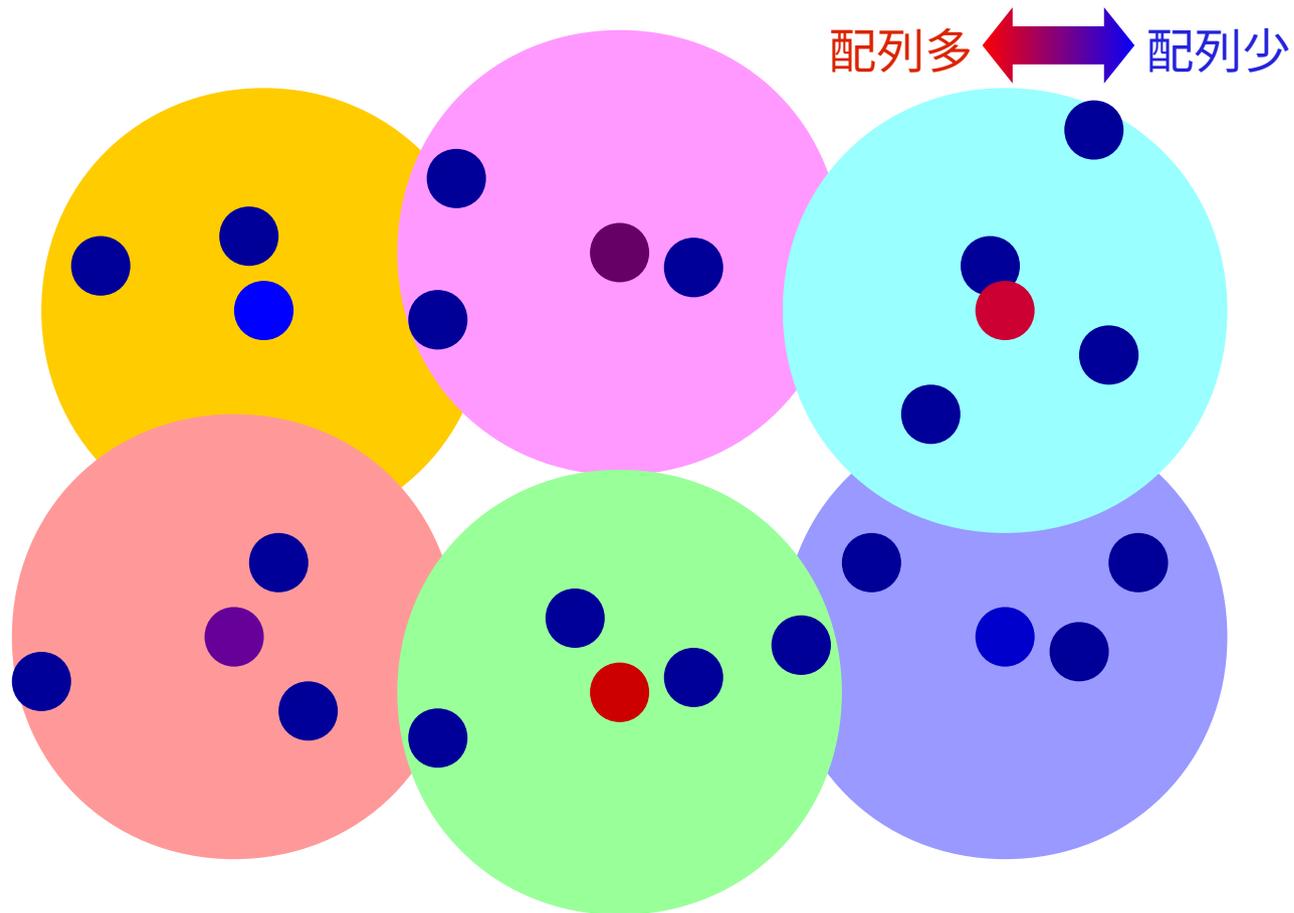
Clustering by using VSEARCH - clclasseqv

- 残っている配列を VSEARCH に渡してクラスタリング
 - 完全一致配列をクラスタリングしてコピー数をカウント
 - コピー数の多い配列から順によりコピー数の多い配列と比較
 - 類似度が閾値を上回ればまとめる



Clustering by using VSEARCH - clclasseqv

- 残っている配列を VSEARCH に渡してクラスタリング
 - 完全一致配列をクラスタリングしてコピー数をカウント
 - コピー数の多い配列から順によりコピー数の多い配列と比較
 - 類似度が閾値を上回ればまとめる



Recovering excluded sequences - clrecoveryseqv

Recovering excluded sequences - clrecoverseqv

- quality-filtering や denoising で除去された配列でも、クラスタ内に入るなら復活させてやる

Recovering excluded sequences - clrecoverseqv

- quality-filtering や denoising で除去された配列でも、クラスタ内に入るなら復活させてやる
- 本当にやった方がいいかどうかは不明

De novo chimera removal - clrunuchime



De novo chimera removal - clrunuchime

キメラは PCR 中に生成されるので、
PCR 前からある親配列よりコピー数が小さいはず



De novo chimera removal - clrunuchime

キメラ配列の親配列のリード数

V

キメラ配列のうちの 1 種類のリード数

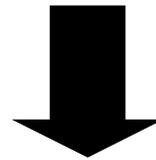


De novo chimera removal - clrunuchime

キメラ配列の親配列のリード数

V

キメラ配列のうちの 1 種類のリード数



上の条件を満たす配列をキメラとみなして除去

(Edgar et al. 2011)

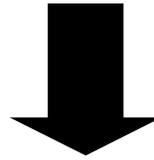
Chimera removal using reference - clrunuchime

Chimera removal using reference - clrunuchime

データをクエリにして参照配列データベースを検索

Chimera removal using reference - clrunuchime

データをクエリにして参照配列データベースを検索



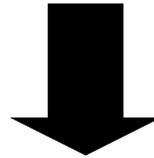
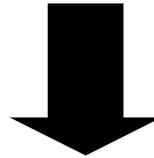
参照配列 1 

参照配列 2 

データ中のキメラ配列 1 

Chimera removal using reference - clrunuchime

データをクエリにして参照配列データベースを検索



上の条件を満たす配列をキメラとみなして除去

(Edgar et al. 2011)

Reference sequence databases for UCHIME

Reference sequence databases for UCHIME

- 細菌 16S 用参照配列データベースは RDP から入手可能

Reference sequence databases for UCHIME

- 細菌 16S 用参照配列データベースは RDP から入手可能
- 真菌 ITS 用参照配列データベースは UNITE から入手可能

Reference sequence databases for UCHIME

- 細菌 16S 用参照配列データベースは RDP から入手可能
- 真菌 ITS 用参照配列データベースは UNITE から入手可能
- 動物ミトゲノムや植物葉緑体ゲノム用参照配列データベースを準備中

Target locus extraction

Target locus extraction

- ITSx による ITS1/ITS2 の抽出

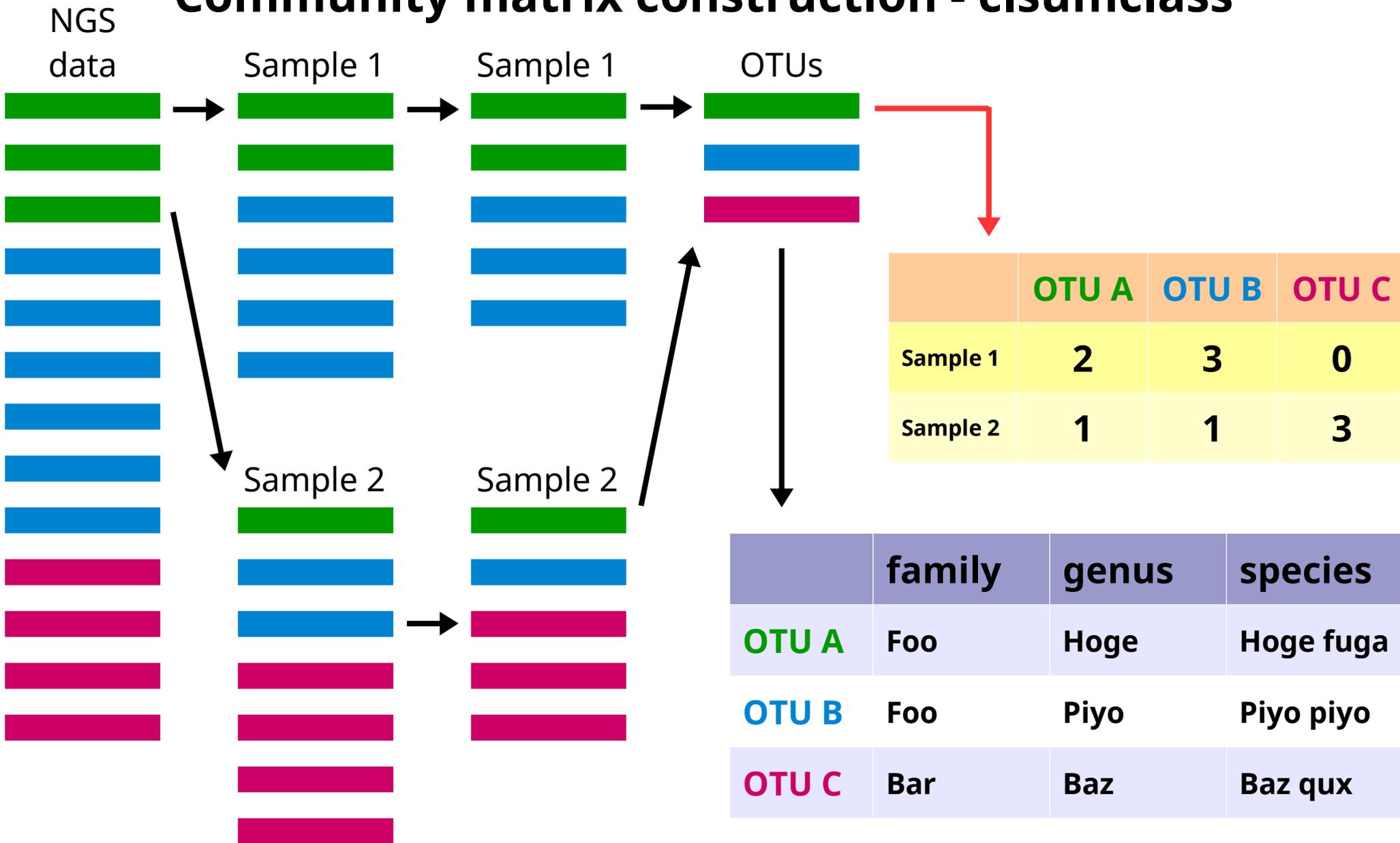
Target locus extraction

- ITSx による ITS1/ITS2 の抽出
- Metaxa による LSU/SSU rRNA の抽出

Target locus extraction

- ITSx による ITS1/ITS2 の抽出
- Metaxa による LSU/SSU rRNA の抽出
- cldivseq による切断

Community matrix construction - clsumclass



Community matrix construction - clsumclass

- clustered.otu.gz から以下のテーブルを生成する

Community matrix construction - clsumclass

- clustered.otu.gz から以下のテーブルを生成する

samplename	OTU1	OTU2	OTU3	OTU4	OTU5	OTUN
sample1	10	0	101	344	433	674
sample2	883	101002	6	0	3	0
sample3	403	206	98	633	334	122
sample4	5769	22737	9954	5001	1224	184
sampleN	558	0	0	22	105	199

Community matrix construction - clsumclass

- clustered.otu.gz から以下のテーブルを生成する
- クラスタリング後に配列をフィルタリングした場合は cfiltersum で加工

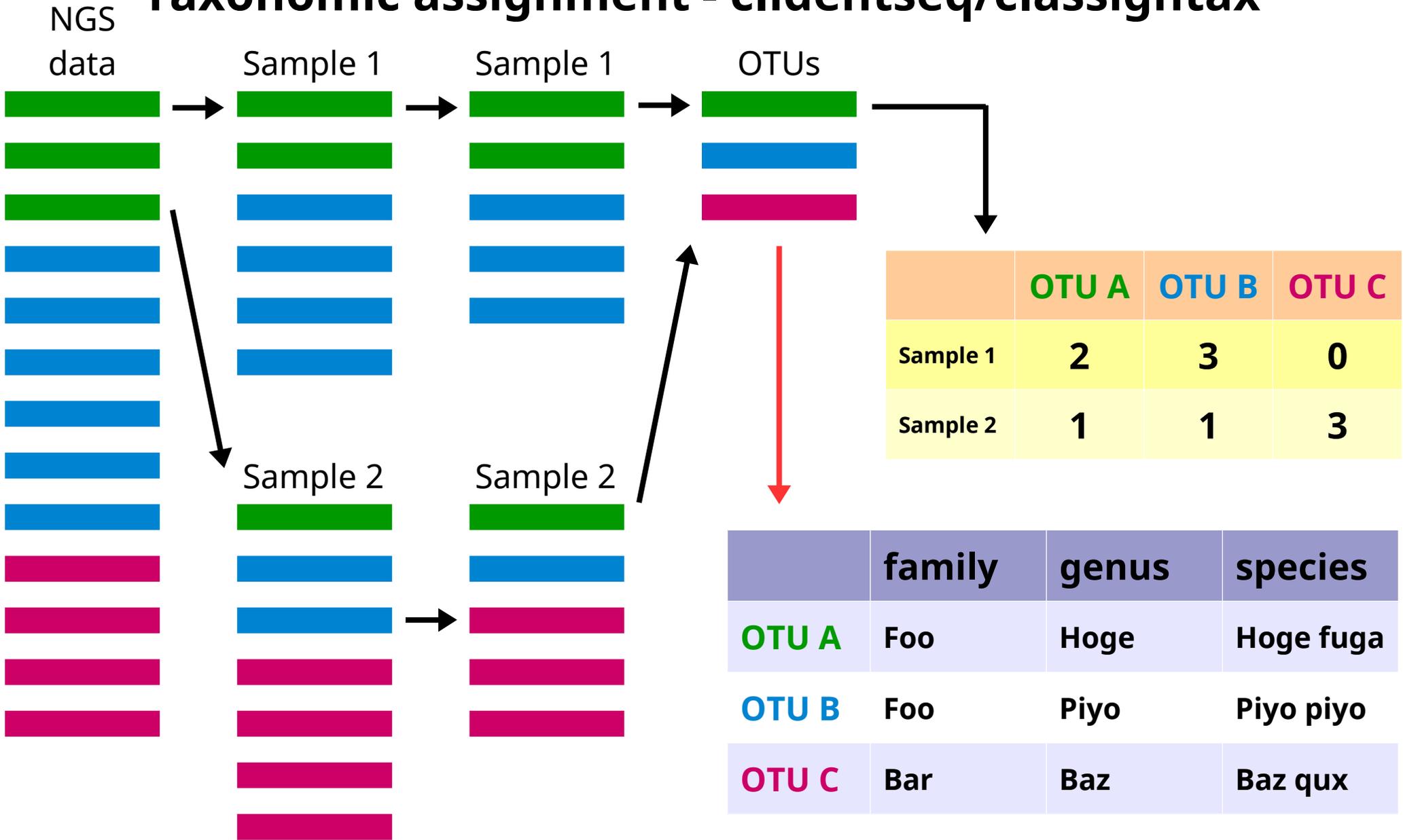
samplename	OTU1	OTU2	OTU3	OTU4	OTU5	OTUN
sample1	10	0	101	344	433	674
sample2	883	101002	6	0	3	0
sample3	403	206	98	633	334	122
sample4	5769	22737	9954	5001	1224	184
sampleN	558	0	0	22	105	199

Community matrix construction - clsumclass

- clustered.otu.gz から以下のテーブルを生成する
- クラスタリング後に配列をフィルタリングした場合は cfiltersum で加工
 - 配列データに合わせてテーブルをフィルタリングする

samplename	OTU1	OTU2	OTU3	OTU4	OTU5	OTUN
sample1	10	0	101	344	433	674
sample2	883	101002	6	0	3	0
sample3	403	206	98	633	334	122
sample4	5769	22737	9954	5001	1224	184
sampleN	558	0	0	22	105	199

Taxonomic assignment - clidentseq/classignntax



Taxonomic assignment - clidentseq/classigntax

- 参照配列データベースから得た BLAST 検索結果に基づいて OTU を分子同定

Taxonomic assignment - clidentseq/classigntax

- 参照配列データベースから得た BLAST 検索結果に基づいて OTU を分子同定
 1. 参照配列データベースを対象に BLAST 検索して近隣配列群を取得

Taxonomic assignment - clidentseq/classigntax

- 参照配列データベースから得た BLAST 検索結果に基づいて OTU を分子同定
 1. 参照配列データベースを対象に BLAST 検索して近隣配列群を取得
 2. 近隣配列群の分類群情報の厳密合意分類群を採用

Taxonomic assignment - clidentseq/classigntax

- 参照配列データベースから得た BLAST 検索結果に基づいて OTU を分子同定
 1. 参照配列データベースを対象に BLAST 検索して近隣配列群を取得
 2. 近隣配列群の分類群情報の厳密合意分類群を採用
- 厳密合意を任意の閾値の多数決合意に変更可能

Taxonomic assignment - clidentseq/classigntax

- 参照配列データベースから得た BLAST 検索結果に基づいて OTU を分子同定
 1. 参照配列データベースを対象に BLAST 検索して近隣配列群を取得
 2. 近隣配列群の分類群情報の厳密合意分類群を採用
- 厳密合意を任意の閾値の多数決合意に変更可能
 - 誤同定されている配列がヒットすると下位まで同定できないのを回避

Criterion proposed in Tanabe and Toju (2013)

Criterion proposed in Tanabe and Toju (2013)

問い合わせ配列と最近隣配列間の変異量

Criterion proposed in Tanabe and Toju (2013)

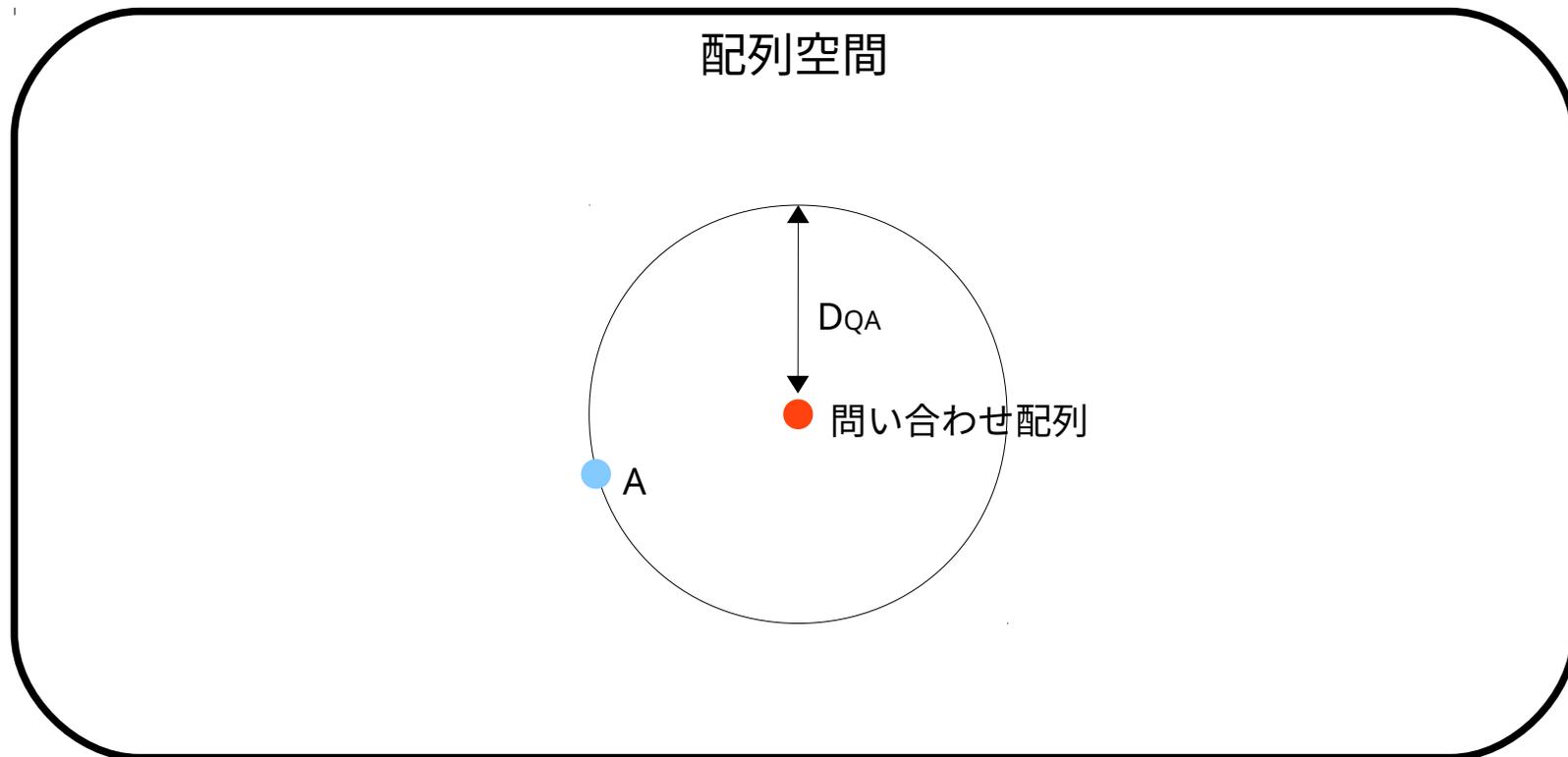
問い合わせ配列と最近隣配列間の変異量

∧

同定結果分類群内の最大変異量

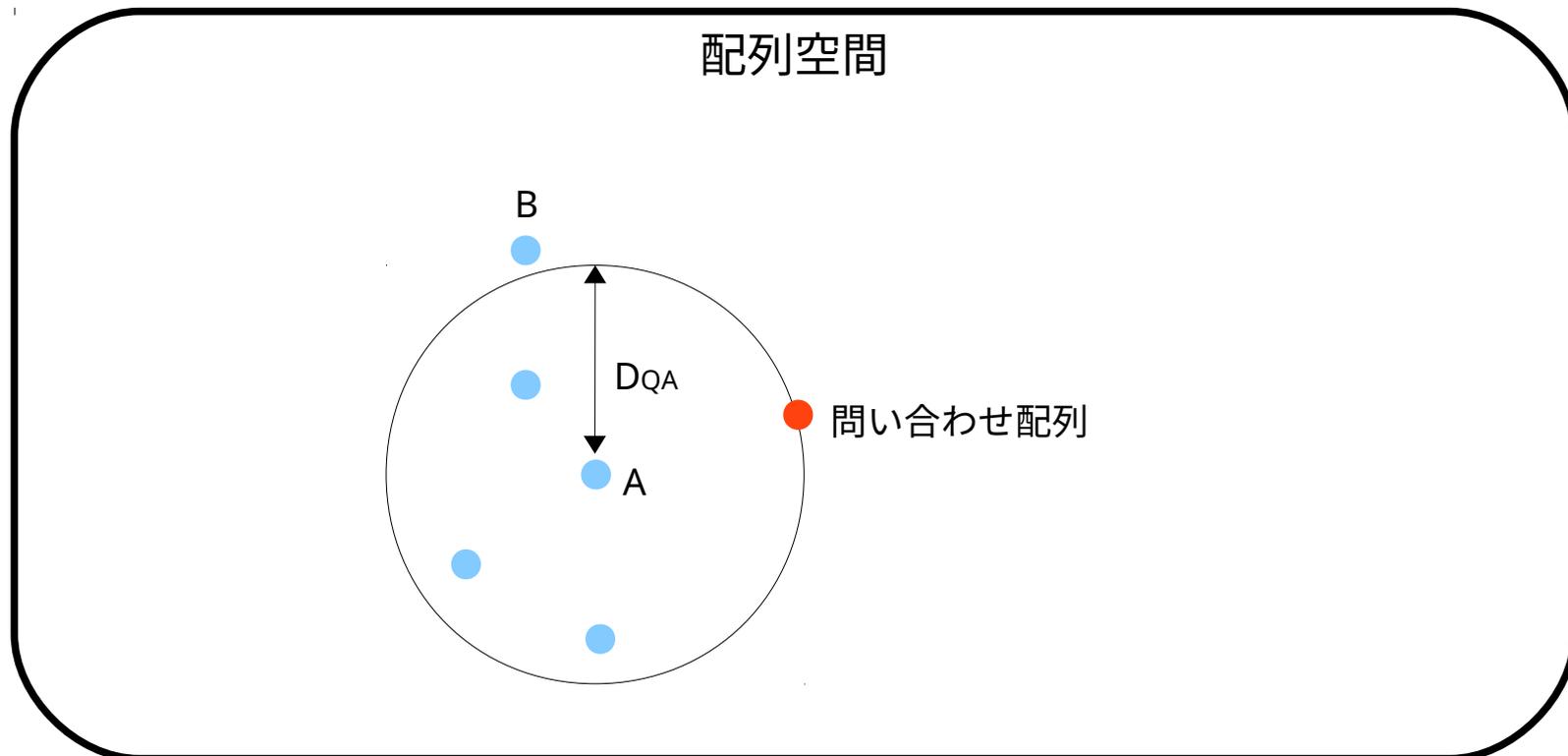
Query-centric auto- k -NN (QCauto) method

1. 最近隣配列 (A) を探し出して変異量 (D_{QA}) を算出



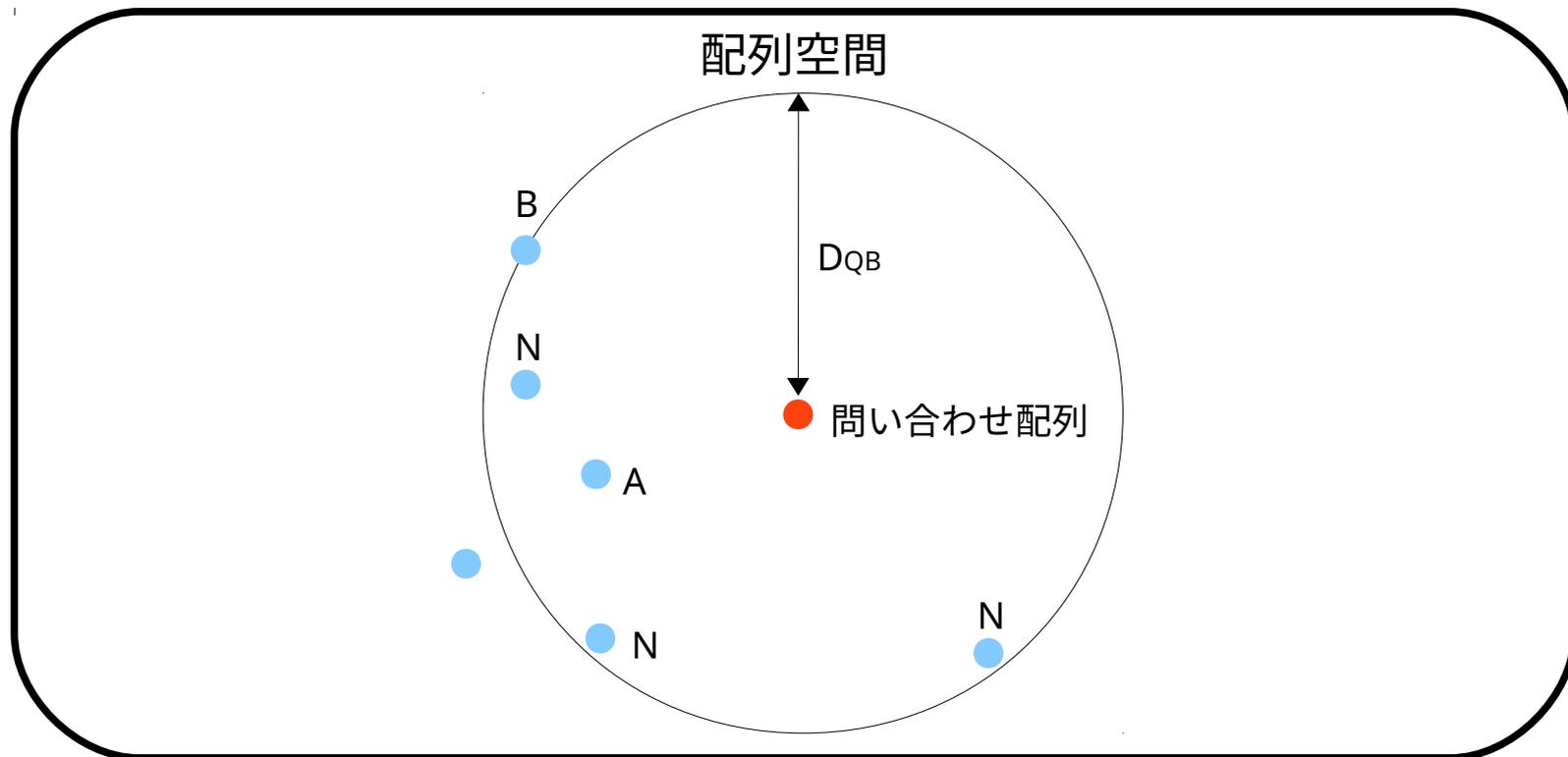
Query-centric auto- k -NN (QCauto) method

1. 最近隣配列 (A) を探し出して変異量 (D_{QA}) を算出
2. $D_{AB} > D_{QA}$ を満たす配列のうち最も A に近い配列 (B) を得る



Query-centric auto- k -NN (QCauto) method

1. 最近隣配列 (A) を探し出して変異量 (D_{QA}) を算出
2. $D_{AB} > D_{QA}$ を満たす配列のうち最も A に近い配列 (B) を得る
3. $D_{QN} \leq D_{QB}$ を満たすすべての配列 (N) を得る
4. A, B, N の全配列で共通する分類群を採用



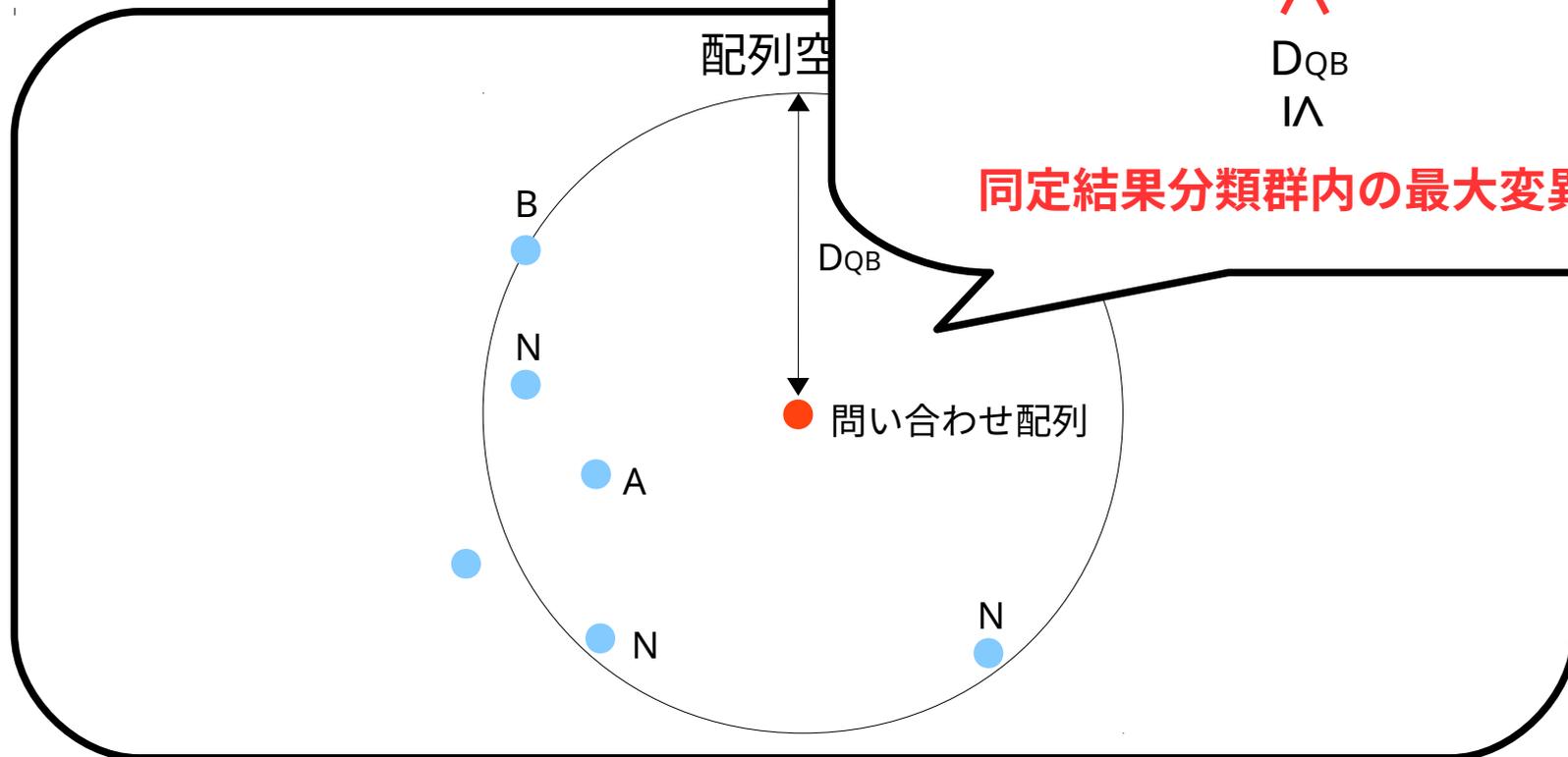
Query-centric auto- k -NN (QCauto) method

1. 最近隣配列 (A) を探し出して変異量 (D_{QA}) を算出
2. $D_{AB} > D_{QA}$ を満たす配列のうち最も A に近い配列 (B) を得る
3. $D_{QN} \leq D_{QB}$ を満たすすべての配列 (N) を抽出
4. A, B, N の全配列で共通する分類群を抽出

問い合わせ配列と最近隣配列間の変異量

\parallel
 D_{QA}
 \wedge
 D_{QB}
 \cap

同定結果分類群内の最大変異量



イメージ図

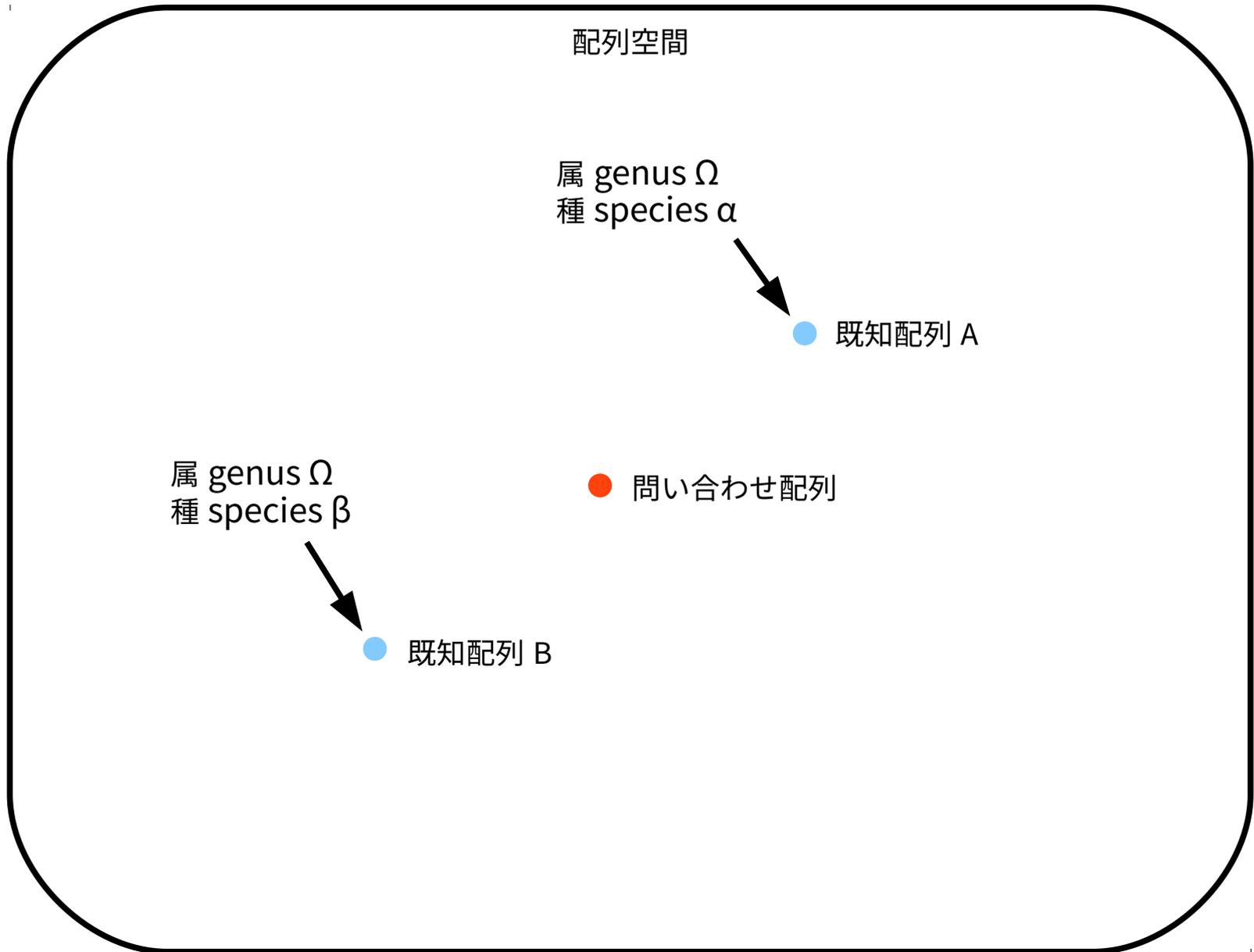
配列空間

● 既知配列 A

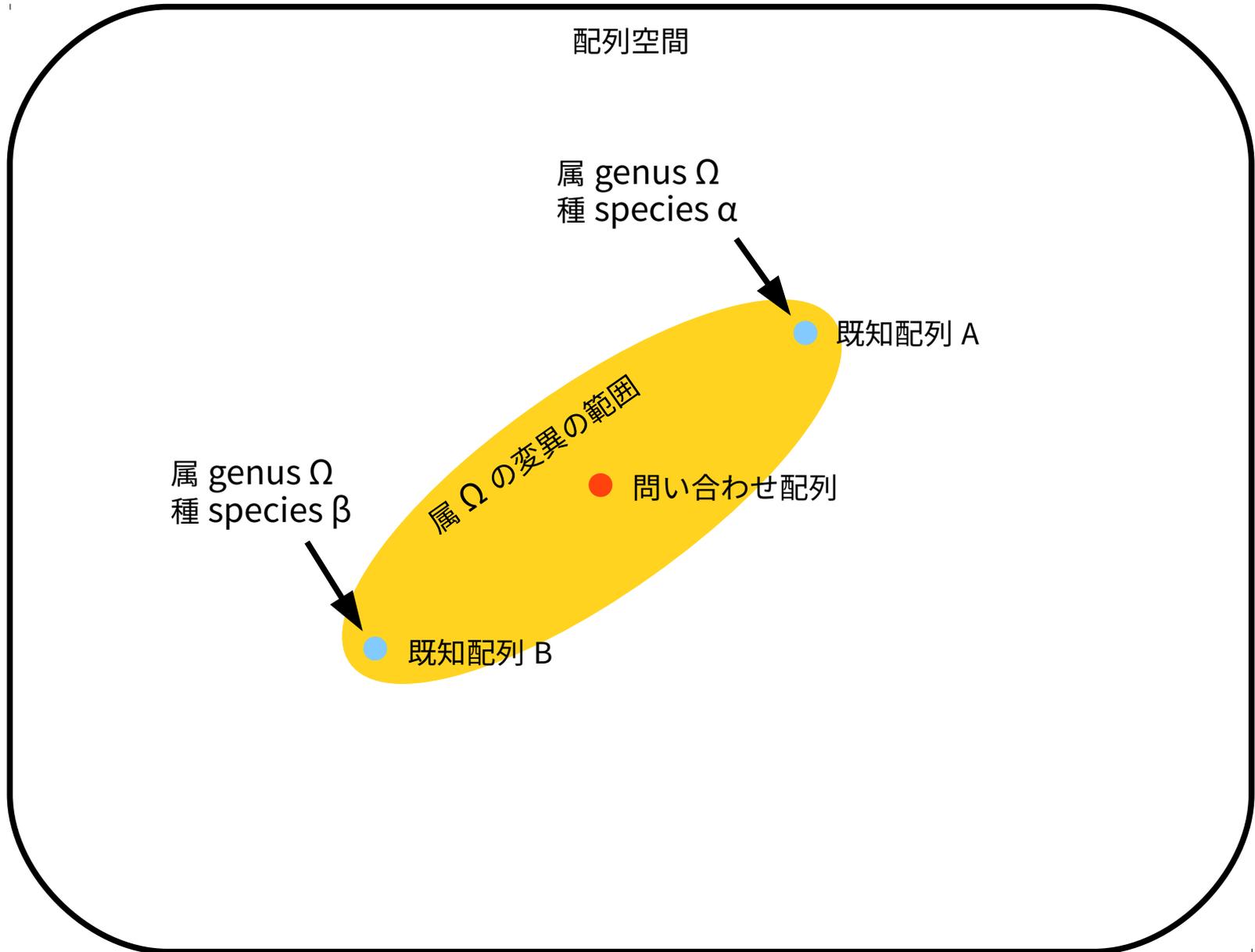
● 問い合わせ配列

● 既知配列 B

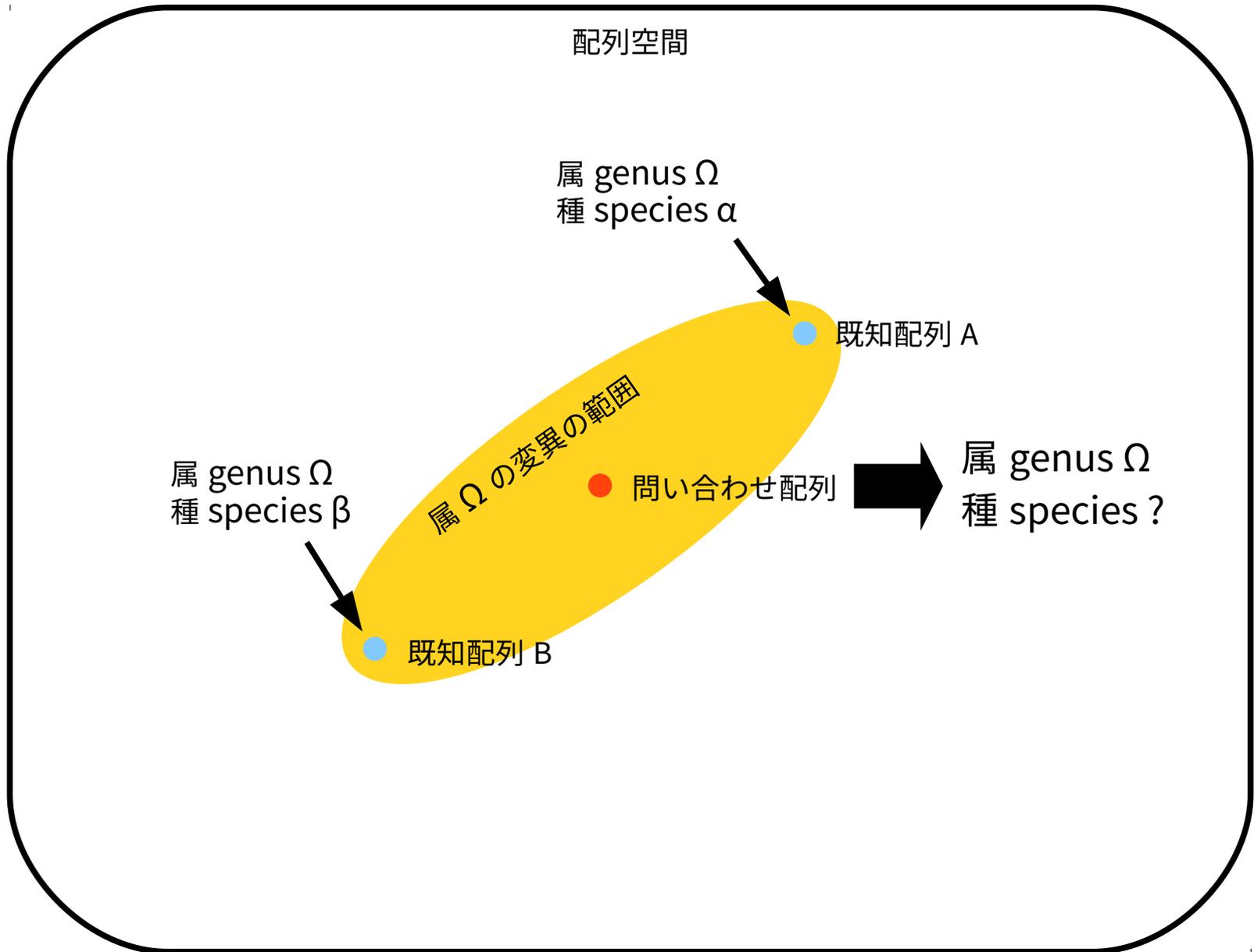
イメージ図



イメージ図



イメージ図



Merging multiple assignment results - clmergeassign

Merging multiple assignment results - clmergeassign

- 複数の既知配列データベースでの同定結果を統合

Merging multiple assignment results - clmergeassign

- 複数の既知配列データベースでの同定結果を統合
 - 種以下まで情報がある既知配列のみのデータベースでの同定結果
 - 科以下まで情報がある既知配列のみのデータベースでの同定結果

Merging multiple assignment results - clmergeassign

- 複数の既知配列データベースでの同定結果を統合
 - 種以下まで情報がある既知配列のみのデータベースでの同定結果
 - 科以下まで情報がある既知配列のみのデータベースでの同定結果
- 厳しい制約下の同定結果と制約を緩めた同定結果を統合

Merging multiple assignment results - clmergeassign

- 複数の既知配列データベースでの同定結果を統合
 - 種以下まで情報がある既知配列のみのデータベースでの同定結果
 - 科以下まで情報がある既知配列のみのデータベースでの同定結果
- 厳しい制約下の同定結果と制約を緩めた同定結果を統合
 - 類似配列の厳密合意分類群を採用した同定結果
 - 類似配列の 90% 多数決合意分類群を採用した同定結果

Merging multiple assignment results - clmergeassign

- 複数の既知配列データベースでの同定結果を統合
 - 種以下まで情報がある既知配列のみのデータベースでの同定結果
 - 科以下まで情報がある既知配列のみのデータベースでの同定結果
- 厳しい制約下の同定結果と制約を緩めた同定結果を統合
 - 類似配列の厳密合意分類群を採用した同定結果
 - 類似配列の 90% 多数決合意分類群を採用した同定結果
- 複数の遺伝子座での同定結果を統合

Merging multiple assignment results - clmergeassign

- 複数の既知配列データベースでの同定結果を統合
 - 種以下まで情報がある既知配列のみのデータベースでの同定結果
 - 科以下まで情報がある既知配列のみのデータベースでの同定結果
- 厳しい制約下の同定結果と制約を緩めた同定結果を統合
 - 類似配列の厳密合意分類群を採用した同定結果
 - 類似配列の 90% 多数決合意分類群を採用した同定結果
- 複数の遺伝子座での同定結果を統合
 - 遺伝子座 A の配列での同定結果
 - 遺伝子座 B の配列での同定結果

Merging multiple assignment results - clmergeassign

- 複数の既知配列データベースでの同定結果を統合
 - 種以下まで情報がある既知配列のみのデータベースでの同定結果
 - 科以下まで情報がある既知配列のみのデータベースでの同定結果
- 厳しい制約下の同定結果と制約を緩めた同定結果を統合
 - 類似配列の厳密合意分類群を採用した同定結果
 - 類似配列の 90% 多数決合意分類群を採用した同定結果
- 複数の遺伝子座での同定結果を統合
 - 遺伝子座 A の配列での同定結果
 - 遺伝子座 B の配列での同定結果
- 優先順位を設定可能

Merging multiple assignment results - clmergeassign

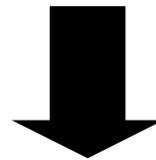
- 複数の既知配列データベースでの同定結果を統合
 - 種以下まで情報がある既知配列のみのデータベースでの同定結果
 - 科以下まで情報がある既知配列のみのデータベースでの同定結果
- 厳しい制約下の同定結果と制約を緩めた同定結果を統合
 - 類似配列の厳密合意分類群を採用した同定結果
 - 類似配列の 90% 多数決合意分類群を採用した同定結果
- 複数の遺伝子座での同定結果を統合
 - 遺伝子座 A の配列での同定結果
 - 遺伝子座 B の配列での同定結果
- 優先順位を設定可能
 - 優先順位は自分で決める必要がある

Merging taxonomy with priority

優先度	綱	目	科	属	種
高	哺乳綱	霊長目	ヒト科		
低	哺乳綱	霊長目	ヒト科	ヒト属	ヒト

Merging taxonomy with priority

優先度	綱	目	科	属	種
高	哺乳綱	霊長目	ヒト科		
低	哺乳綱	霊長目	ヒト科	ヒト属	ヒト



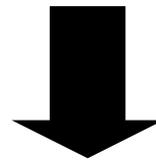
	綱	目	科	属	種
統合結果	哺乳綱	霊長目	ヒト科	ヒト属	ヒト

Merging taxonomy with priority

優先度	綱	目	科	属	種
高	哺乳綱	霊長目	ヒト科	ゴリラ属	ゴリラ
低	哺乳綱	霊長目	ヒト科	ヒト属	ヒト

Merging taxonomy with priority

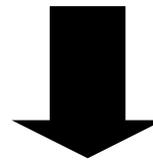
優先度	綱	目	科	属	種
高	哺乳綱	霊長目	ヒト科	ゴリラ属	ゴリラ
低	哺乳綱	霊長目	ヒト科	ヒト属	ヒト



	綱	目	科	属	種
統合結果	哺乳綱	霊長目	ヒト科	ゴリラ属	ゴリラ

Merging taxonomy with priority

優先度	綱	目	科	属	種
高	哺乳綱	霊長目	ヒト科	ゴリラ属	ゴリラ
低	哺乳綱	霊長目	ヒト科	ヒト属	ヒト



	綱	目	科	属	種
統合結果	哺乳綱	霊長目	ヒト科	ゴリラ属	ゴリラ

優先度の低い同定結果は

- 優先度の高い同定結果より低レベルまで同定できている
- 優先度の高い同定結果と矛盾がない

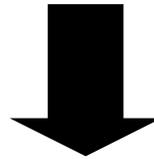
場合に限り結果を採用される

Merging taxonomy with priority

優先度	綱	目	科	属	種
同	哺乳綱	霊長目	ヒト科	ゴリラ属	ゴリラ
同	哺乳綱	霊長目	ヒト科	ヒト属	ヒト

Merging taxonomy with priority

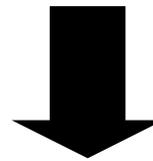
優先度	綱	目	科	属	種
同	哺乳綱	霊長目	ヒト科	ゴリラ属	ゴリラ
同	哺乳綱	霊長目	ヒト科	ヒト属	ヒト



	綱	目	科	属	種
統合結果	哺乳綱	霊長目	ヒト科		

Merging taxonomy with priority

優先度	綱	目	科	属	種
同	哺乳綱	霊長目	ヒト科	ゴリラ属	ゴリラ
同	哺乳綱	霊長目	ヒト科	ヒト属	ヒト



	綱	目	科	属	種
統合結果	哺乳綱	霊長目	ヒト科		

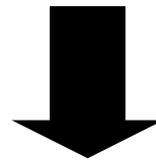
優先度が同じ同定結果に矛盾がある場合、
厳密合意を採用

Merging taxonomy with priority

優先度	綱	目	科	属	種
同	哺乳綱	霊長目	ヒト科	ゴリラ属	ゴリラ
同	哺乳綱	霊長目	ヒト科	ヒト属	

Merging taxonomy with priority

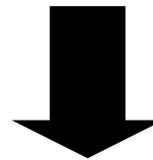
優先度	綱	目	科	属	種
同	哺乳綱	霊長目	ヒト科	ゴリラ属	ゴリラ
同	哺乳綱	霊長目	ヒト科	ヒト属	



	綱	目	科	属	種
統合結果	哺乳綱	霊長目	ヒト科	ゴリラ属	ゴリラ

Merging taxonomy with priority

優先度	綱	目	科	属	種
同	哺乳綱	霊長目	ヒト科	ゴリラ属	ゴリラ
同	哺乳綱	霊長目	ヒト科	ヒト属	



	綱	目	科	属	種
統合結果	哺乳綱	霊長目	ヒト科	ゴリラ属	ゴリラ

優先度が同じ同定結果に矛盾がある場合、 --preferlower オプションでより低レベルまで同定できている結果を採用

Sampling as multiple filtering problem

Sampling as multiple filtering problem

- 群集生態学では飽和サンプリングできるかどうか、できないなら飽和度の均質化が重要になる

Sampling as multiple filtering problem

- 群集生態学では飽和サンプリングできるかどうか、できないなら飽和度の均質化が重要になる
- 野外観察・採集ではサンプリングは 1 段階

Sampling as multiple filtering problem

- 群集生態学では飽和サンプリングできるかどうか、できないなら飽和度の均質化が重要になる
- 野外観察・採集ではサンプリングは1段階
- メタバーコーディングでは、多段階になる

Sampling as multiple filtering problem

- 群集生態学では飽和サンプリングできるかどうか、できないなら飽和度の均質化が重要になる
- 野外観察・採集ではサンプリングは1段階
- メタバーコーディングでは、多段階になる
 1. 現物のサンプリング
 2. DNA 抽出
 3. PCR
 4. シーケンス (塩基配列のサンプリング)

Sampling as multiple filtering problem

- 群集生態学では飽和サンプリングできるかどうか、できないなら飽和度の均質化が重要になる
- 野外観察・採集ではサンプリングは1段階
- メタバーコーディングでは、多段階になる
 1. 現物のサンプリング
 2. DNA 抽出
 3. PCR
 4. シーケンス (塩基配列のサンプリング)
- シーケンスを飽和させても、サンプリング量不足・DNA抽出法が不適切・PCRバイアス等で実際には飽和サンプリングできていないかもしれない

Sampling as multiple filtering problem

- 群集生態学では飽和サンプリングできるかどうか、できないなら飽和度の均質化が重要になる
- 野外観察・採集ではサンプリングは1段階
- メタバーコーディングでは、多段階になる
 1. 現物のサンプリング
 2. DNA 抽出
 3. PCR
 4. シーケンス (塩基配列のサンプリング)
- シーケンスを飽和させても、サンプリング量不足・DNA抽出法が不適切・PCRバイアス等で実際には飽和サンプリングできていないかもしれない
 - レプリケートの確保とレプリケートを足したときに OTU が増加するかのチェックが有効

相対・絶対 abundance の定量

相対・絶対 abundance の定量

無理

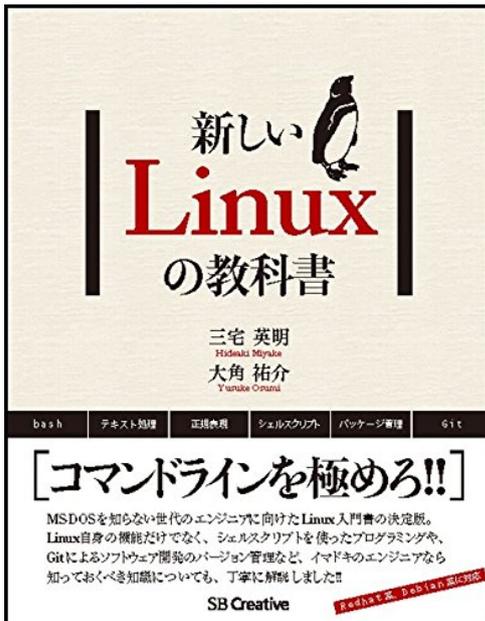
必要となる基礎知識

必要となる基礎知識

- UNIX コマンドライン操作

必要となる基礎知識

- UNIX コマンドライン操作



必要となる基礎知識

- UNIX コマンドライン操作



必要となる基礎知識

- UNIX コマンドライン操作
- R によるデータ解析



必要となる基礎知識

- UNIX コマンドライン操作
- R によるデータ解析



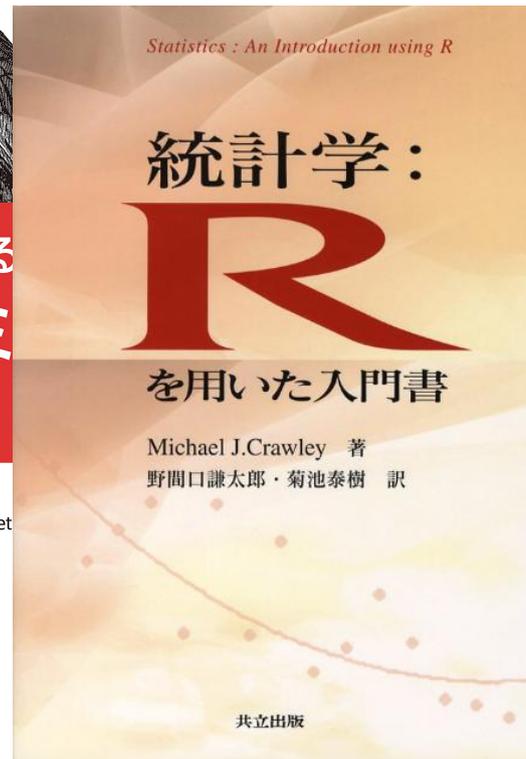
Garrett Grolemond 著
大橋 真也 監訳
長尾 高弘 訳

必要となる基礎知識

- UNIX コマンドライン操作
- R によるデータ解析



Garret

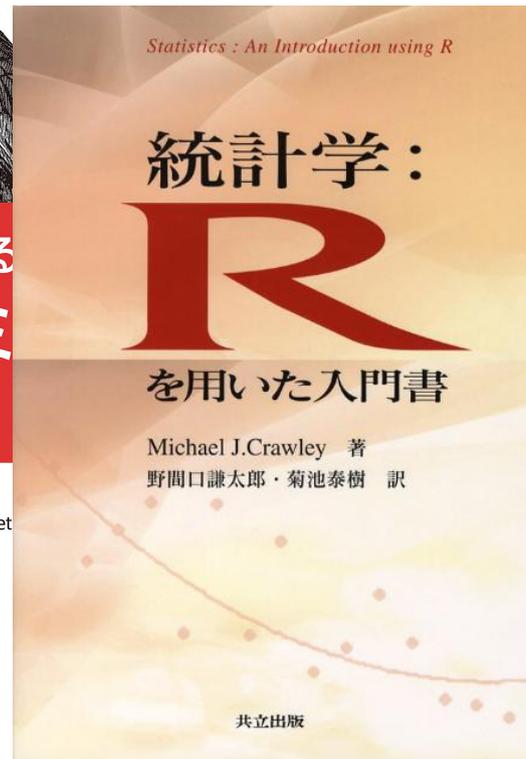


必要となる基礎知識

- UNIX コマンドライン操作
- R によるデータ解析
- 各種データフォーマット

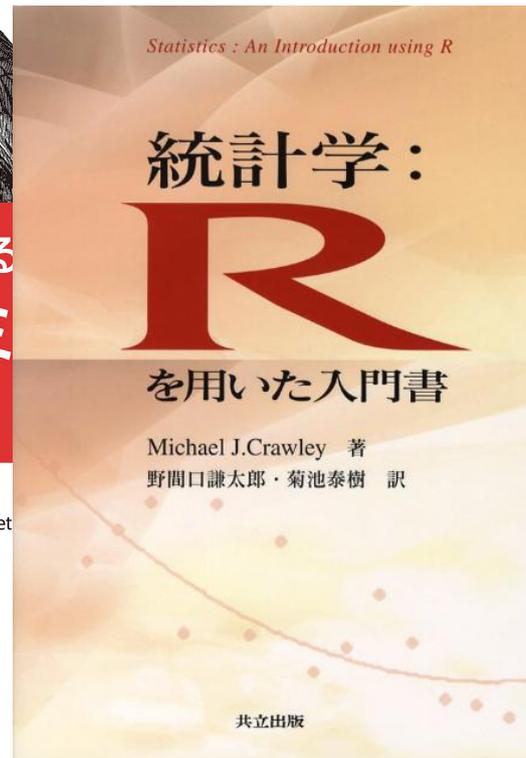


Garret



必要となる基礎知識

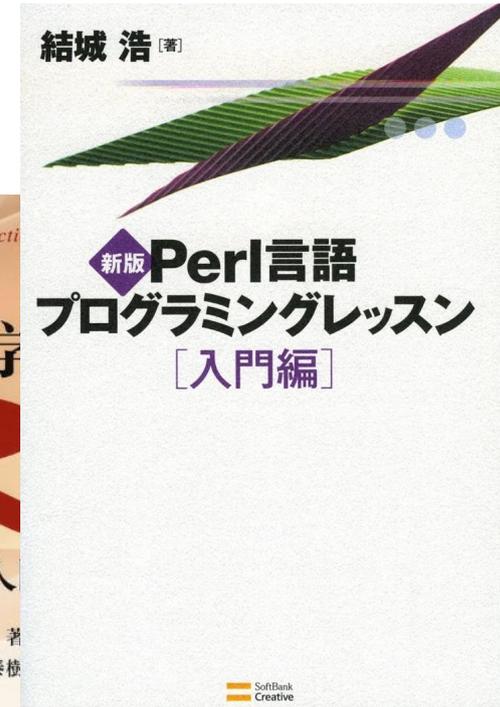
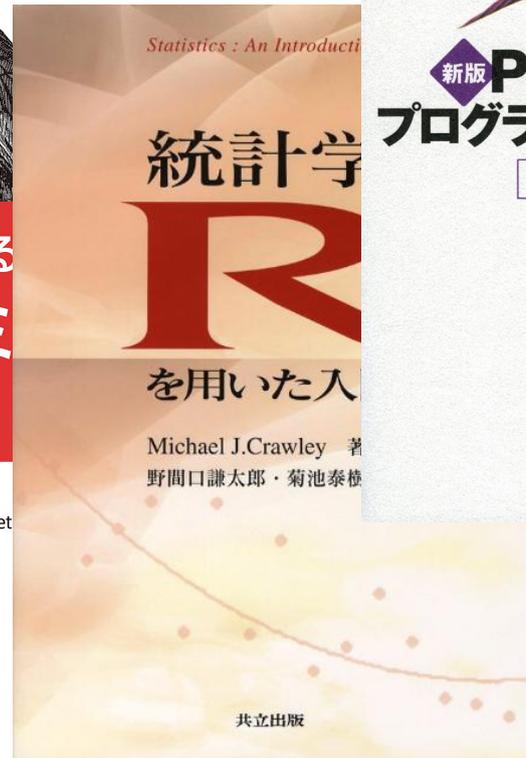
- UNIX コマンドライン操作
- R によるデータ解析
- 各種データフォーマット
- 場合によっては Perl ・ Python ・ Ruby 等による簡易プログラミング



Garret

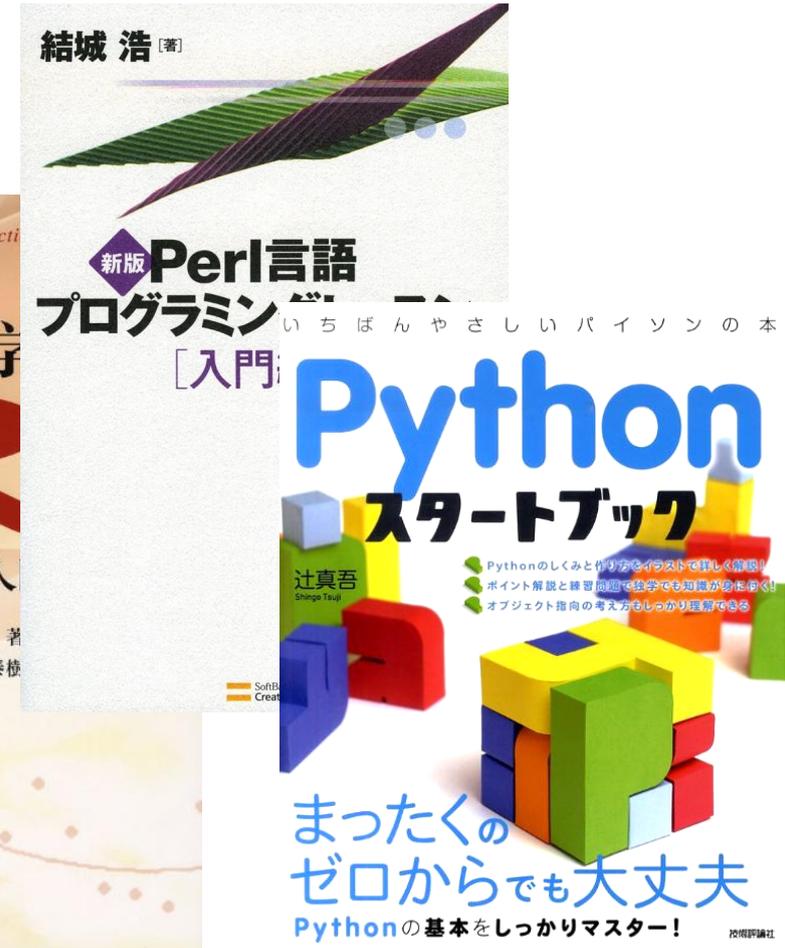
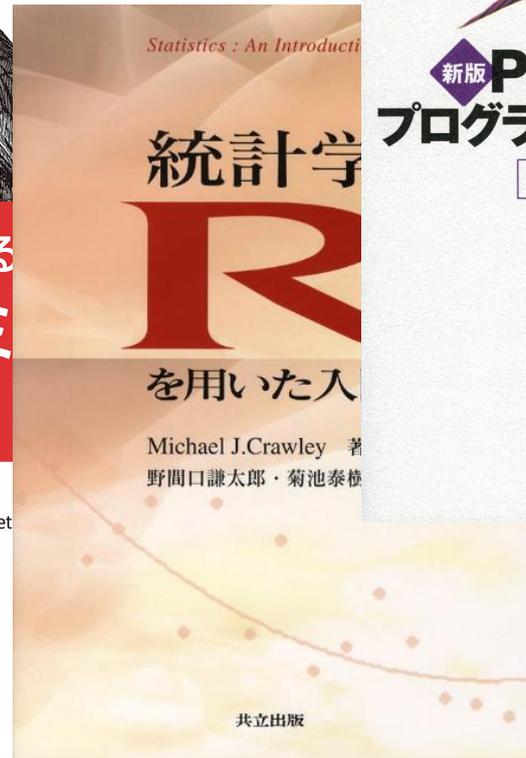
必要となる基礎知識

- UNIX コマンドライン操作
- R によるデータ解析
- 各種データフォーマット
- 場合によっては Perl ・ Python ・ Ruby 等による簡易プログラミング



必要となる基礎知識

- UNIX コマンドライン操作
- R によるデータ解析
- 各種データフォーマット
- 場合によっては Perl ・ Python ・ Ruby 等による簡易プログラミング



必要なハードウェアスペック

必要なハードウェアスペック

- CPU 4 コア、メモリ 32GB が下限

必要なハードウェアスペック

- CPU 4 コア、メモリ 32GB が下限
 - 20 万くらいで買える

必要なハードウェアスペック

- CPU 4 コア、メモリ 32GB が下限
 - 20 万くらいで買える
 - プラス 2 万程度で OS インストールディスクを SSD 化

必要なハードウェアスペック

- CPU 4 コア、メモリ 32GB が下限
 - 20 万くらいで買える
 - プラス 2 万程度で OS インストールディスクを SSD 化
 - プラス 6 万程度でデータディスクを RAID1 化

必要なハードウェアスペック

- CPU 4 コア、メモリ 32GB が下限
 - 20 万くらいで買える
 - プラス 2 万程度で OS インストールディスクを SSD 化
 - プラス 6 万程度でデータディスクを RAID1 化
 - プラス 6 万程度でメモリ 64GB 化

必要な

発行日：2016年03月17日
 有効期限：上記発行日より1週間
 ※「ご注意事項」を必ずお読みください
 商談番号：[M20160317-54371749]
 お名前：ほげほげ様

商談メモ

uniV <http://univpc.com>



UNI-i7ZS

見積もり数量：1台
 見積もり単価：229,200円
 見積もり価格：229,200円
 配送料：0円
 消費税：18,336円
合計金額：247,536円

- CPU 4 コア、メモ
- 20 万くらいで
- プラス 2 万程
- プラス 6 万程
- プラス 6 万程

■詳細は下記の通りです。

区分	仕様
CPU	Core i7 6700 4core/8thread 3.4GH z (Turbo Boost時最大4.0GHz) 8MB HD530 HT対応 Skylake
CPUクーラー	CoolerMaster Hyper212 高耐久静音ファン
マザーボード	intel Z170 Express チップセット
メモリー	32GB (8GBx4) DDR4-2133 Dual-Channel (高品質永久保証品)
HDD1 BootDevice	SSD 240GB MTBF120万時間高耐久 intel MLC
HDD2	SATAIII 3TB
HDD3	SATAIII 3TB
HDD4	なし
SSDキャッシュ	なし
RAID	なし
PCI-Expressストレージ	なし
光学ドライブ	Blu-Rayドライブ BD-R/RE/ROM DVD±R/RW/DL 対応
追加デバイス	なし
カードリーダー	なし
ビデオコントローラー	GeForceGT610 1GB D-Sub/ DVI/ HDMI
ディスプレイ	なし
サウンド	onboard HighDefinition Audio
ネットワーク	onboard Gigabit Ether
ケース	UNI-i7ZS 標準サーマルアドバンスドシャーシミドルタワー (W200xH430xD460mm)
電源	600W 80PLUS BRONZE認証電源
キーボード	なし
マウス	なし
スピーカー	なし
OS	なし
OS 2	なし
MS Office	なし
ウイルス対策ソフト	なし
インテルコンパイラ	なし
保証体制	センドバック保証 1年

必要なハードウェアスペック

- CPU 4 コア、メモリ 32GB が下限
 - 20 万くらいで買える
 - プラス 2 万程度で OS インストールディスクを SSD 化
 - プラス 6 万程度でデータディスクを RAID6 化
 - プラス 6 万程度でメモリ 64GB 化
- CPU 16 コア、メモリ 128GB 以上あるとだいぶ楽になる

必要なハードウェアスペック

- CPU 4 コア、メモリ 32GB が下限
 - 20 万くらいで買える
 - プラス 2 万程度で OS インストールディスクを SSD 化
 - プラス 6 万程度でデータディスクを RAID6 化
 - プラス 6 万程度でメモリ 64GB 化
- CPU 16 コア、メモリ 128GB 以上あるとだいぶ楽になる
 - 100 万コース

必要な!

発行日： 2016年03月17日
 有効期限： 上記発行日より1週間
 ※「ご注意事項」を必ずお読みください
 商談番号： [M20160317-13561746]
 お名前： ほげほげ 様

商談メモ

uniV <http://univpc.com>



見積もり数量： 1台
 見積もり単価： 1,110,800円
 見積もり価格： 1,110,800円
 配送料： 0円
 消費税： 88,864円
合計金額： 1,199,664 円

UNI-XW-E5HX

- CPU 4 コア、メモリ
 - 20 万くらいで買
 - プラス 2 万程度
 - プラス 6 万程度
 - プラス 6 万程度
- CPU 16 コア、メモ
 - 100 万コース

■詳細は下記の通りです。

区分	仕様
CPU	16Core Xeon E5-2640 v3 2.6GHz (8core x 2CPU)
マザーボード	intel C612Chipset (UNI-XW-E5HX)
メモリー	128GB (16GBx8) ECC Registered DDR4-2133 Quad-Channel
HDD1 BootDevice	SSD 240GB MTBF120万時間高耐久 intel MLC
HDD2	SATAIII 3TB MTBF120万時間 エンタープライズ用高信頼性HDD
HDD3	SATAIII 3TB MTBF120万時間 エンタープライズ用高信頼性HDD
HDD4	SATAIII 3TB MTBF120万時間 エンタープライズ用高信頼性HDD
HDD5	SATAIII 3TB MTBF120万時間 エンタープライズ用高信頼性HDD
HDD6	SATAIII 3TB MTBF120万時間 エンタープライズ用高信頼性HDD
SSDキャッシュ	なし
RAID	なし
PCI-Expressストレージ	なし
光学ドライブ	Blu-Rayドライブ BD-R/RE/ROM DVD±R/RW/DL 対応
追加デバイス	なし
カードリーダー	なし
ビデオコントローラー	NVIDIA Quadro K620 2.0GB DVI / DisplayPort
ディスプレイ	なし
サウンド	onboard HighDefinition Audio
ネットワーク	onboard Dual GigabitEther
InfiniBand	なし
ケース	UNI-XW-E5シリーズ 静音フルタワーケース (W178xH452xD648mm)
電源	1200W 静音 80PLUS PLATINUM認証電源 (UNIV Xeon搭載フルタワー用)
キーボード	なし
マウス	なし
スピーカー	なし
OS	なし
OS 2	なし
MS Office	なし
ウイルス対策ソフト	なし
インテルコンパイラ	なし
保証体制	標準センドバック 3年

必要なハードウェアスペック

- CPU 4 コア、メモリ 32GB が下限
 - 20 万くらいで買える
 - プラス 2 万程度で OS インストールディスクを SSD 化
 - プラス 6 万程度でデータディスクを RAID6 化
 - プラス 6 万程度でメモリ 64GB 化
- CPU 16 コア、メモリ 128GB 以上あるとだいぶ楽になる
 - 100 万コース
- CPU 32 コア、メモリ 1TB 以上あれば最高

必要なハードウェアスペック

- CPU 4 コア、メモリ 32GB が下限
 - 20 万くらいで買える
 - プラス 2 万程度で OS インストールディスクを SSD 化
 - プラス 6 万程度でデータディスクを RAID6 化
 - プラス 6 万程度でメモリ 64GB 化
- CPU 16 コア、メモリ 128GB 以上あるとだいぶ楽になる
 - 100 万コース
- CPU 32 コア、メモリ 1TB 以上あれば最高
 - 500 万コース

必要な!

発行日： 2016年03月17日
 有効期限： 上記発行日より1週間
 ※「ご注意事項」を必ずお読みください
 商談番号： [M20160317-44101740]
 お名前： ほげほげ 様

商談メモ

uniV <http://univpc.com>



UNI-XW-E5HX

見積もり数量： 1台
 見積もり単価： 4,778,600円
 見積もり価格： 4,778,600円
 配送料： 0円
 消費税： 382,288円
合計金額： 5,160,888 円

- CPU 4 コア、メモリ
 - 20 万くらいで買
 - プラス 2 万程度
 - プラス 6 万程度
 - プラス 6 万程度
- CPU 16 コア、メモ
 - 100 万コース
- CPU 32 コア、メモ
 - 500 万コース

■詳細は下記の通りです。

区分	仕様
CPU	36Core Xeon E5-2699 v3 2.3GHz (18core x 2CPU)
マザーボード	intel C612Chipset (UNI-XW-E5HX)
メモリー	1024GB (64GBx16) ECC Registered DDR4-2133 Quad-Channel
HDD1 BootDevice	SSD 800GB MTBF200万時間高耐久 intel MLC
HDD2	SATAIII 6TB MTBF120万時間 エンタープライズ用高信頼性HDD
HDD3	SATAIII 6TB MTBF120万時間 エンタープライズ用高信頼性HDD
HDD4	SATAIII 6TB MTBF120万時間 エンタープライズ用高信頼性HDD
HDD5	SATAIII 6TB MTBF120万時間 エンタープライズ用高信頼性HDD
HDD6	SATAIII 6TB MTBF120万時間 エンタープライズ用高信頼性HDD
SSDキャッシュ	なし
RAID	なし
PCI-Expressストレージ	なし
光学ドライブ	Blu-Rayドライブ BD-R/RE/ROM DVD±R/RW/DL 対応
追加デバイス	なし
カードリーダー	なし
ビデオコントローラー	NVIDIA Quadro K620 2.0GB DVI / DisplayPort
ディスプレイ	なし
サウンド	onboard HighDefinition Audio
ネットワーク	onboard Dual GigabitEther
InfiniBand	なし
ケース	UNI-XW-E5シリーズ 静音フルタワーケース (W178xH452xD648mm)
電源	1200W 静音 80PLUS PLATINUM認証電源 (UNIV Xeon搭載フルタワー用)
キーボード	なし
マウス	なし
スピーカー	なし
OS	なし
OS 2	なし
MS Office	なし
ウイルス対策ソフト	なし
インテルコンパイラ	なし
保証体制	標準センドバック 3年