Alignment-free analyses of next-generation sequencing data

Andrew P Morgan

Fernando Pardo-Manuel de Villena lab
Department of Genetics

BCB Research-in-Progress Seminar
9 October 2014

Acknowledgments

Algorithms
J Matthew Holt
Leonard McMillan

Applications
John Didion
Fernando Pardo-Manuel de Villena

Data
Sanger Center/Wellcome Trust
Jim Crowley

Funding
P50GM076468
F30MH103925

Goals of next-generation sequencing (NGS)

Discovery of molecular variation

- DNAseq
- RNAseq
- metagenomics

Quantification

- RNAseq
- CHIPseq
- other exotic *-seq
- metagenomics

Goals of next-generation sequencing (NGS)

Discovery of molecular variation

- DNAseq
- RNAseq
- metagenomics

Quantification

- RNAseq
- CHIPseq
- other exotic *-seq
- metagenomics

Goals of next-generation sequencing (NGS)

Discovery of molecular variation

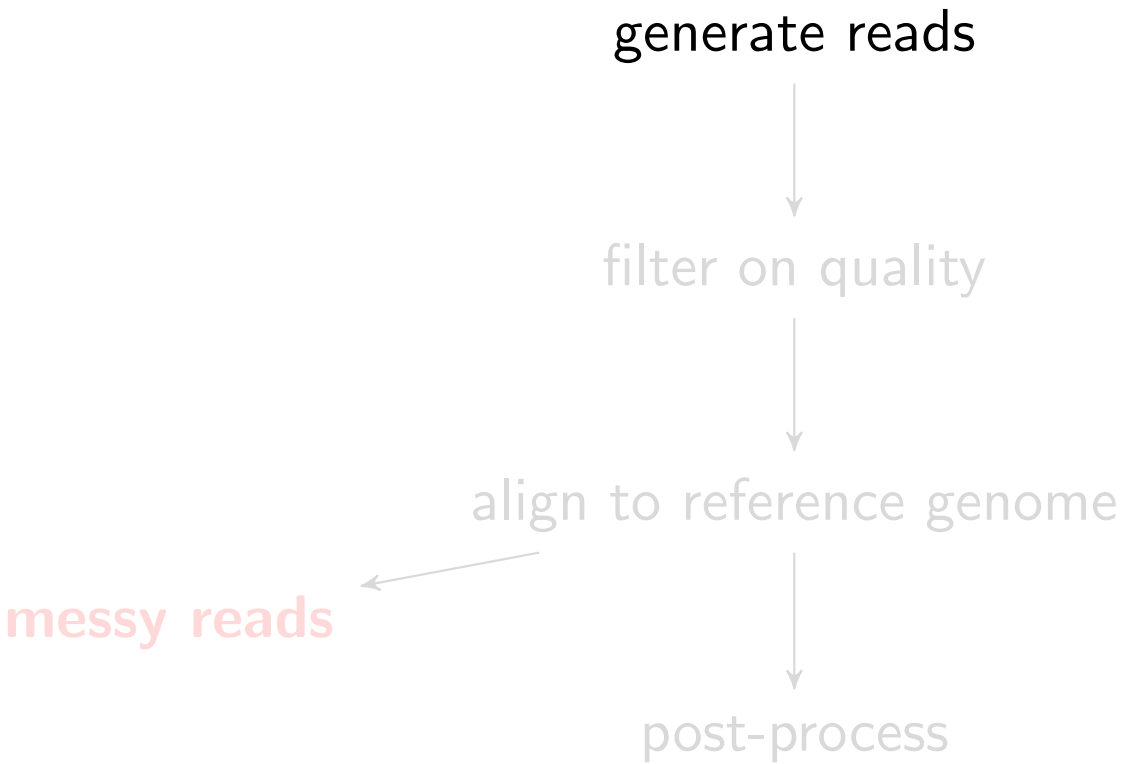
- DNAseq
- RNAseq
- metagenomics

Quantification

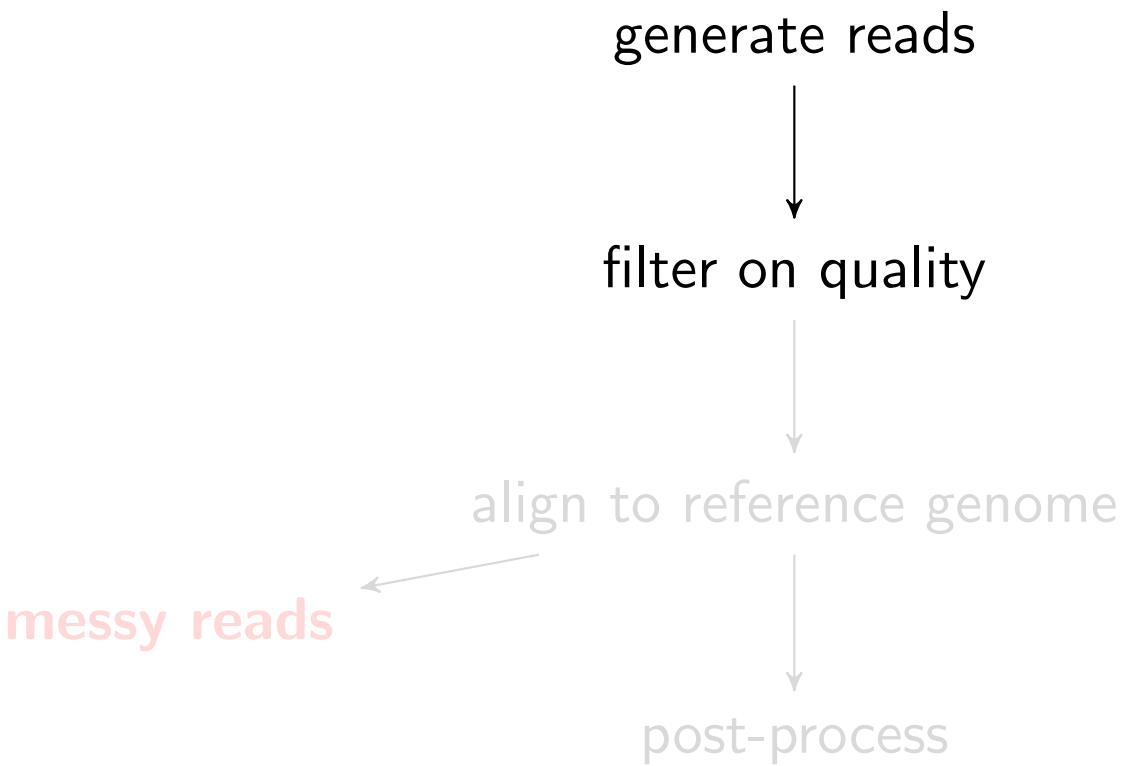
- RNAseq
- CHIPseq
- other exotic *-seq
- metagenomics

Generic NGS workflow

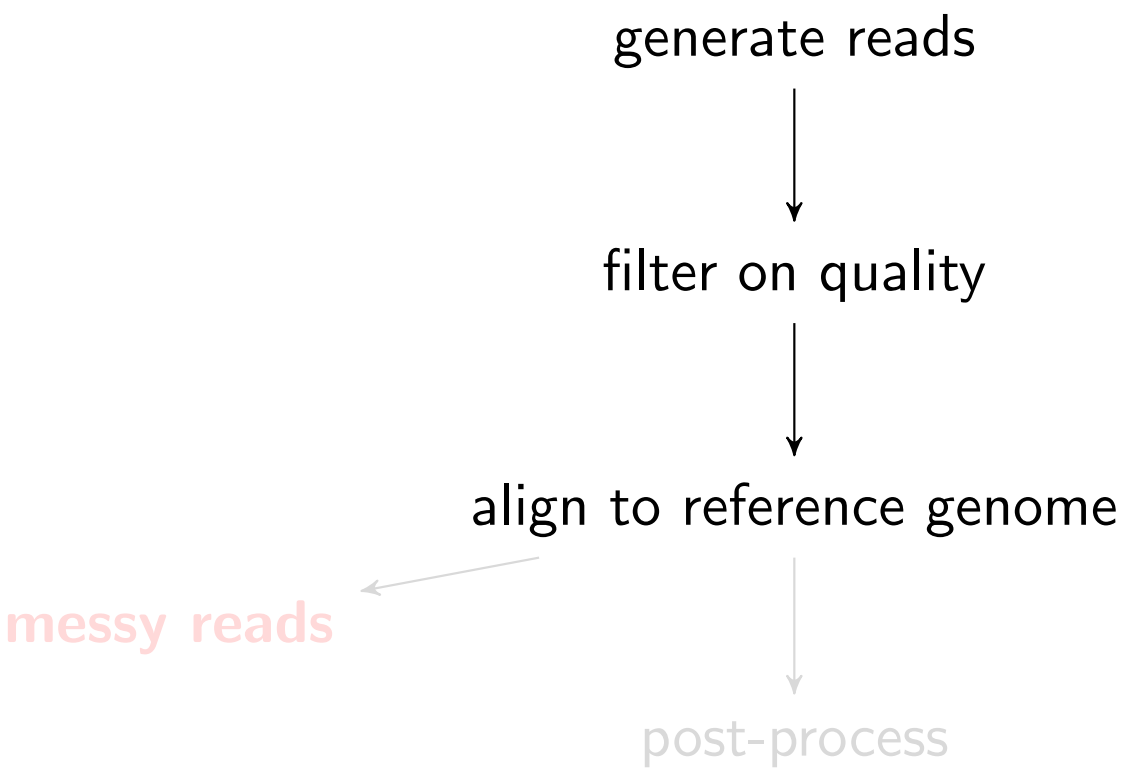
Generic NGS workflow



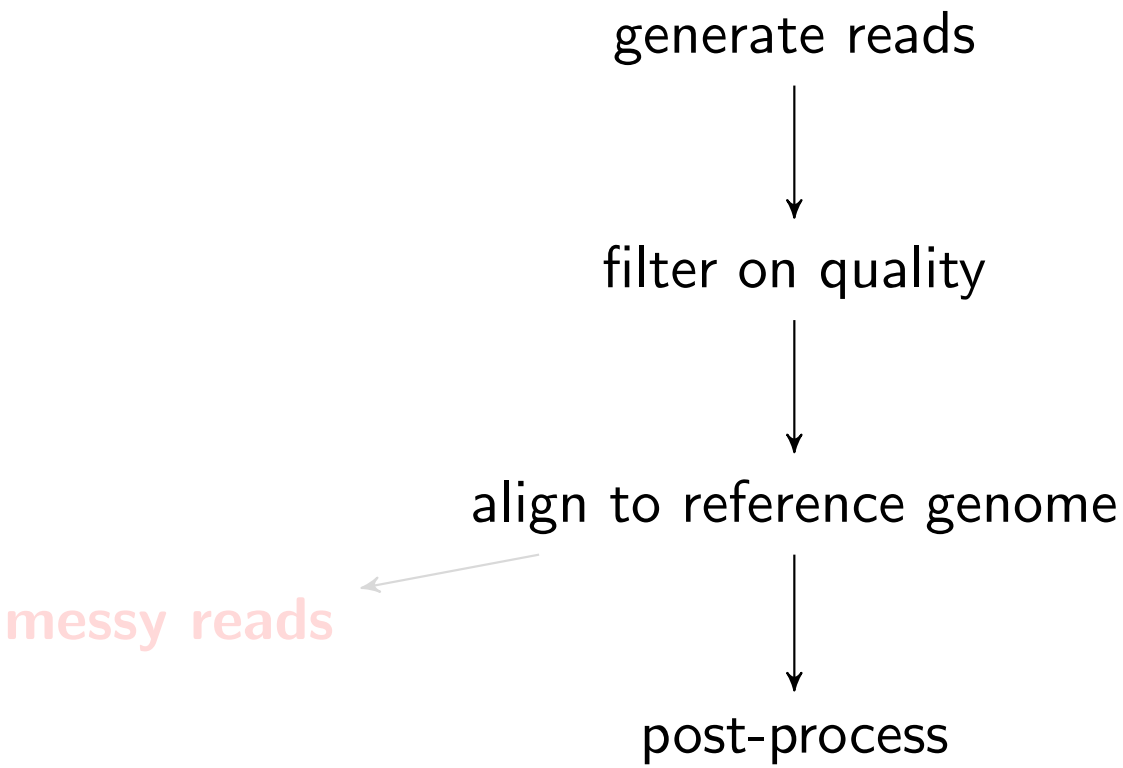
Generic NGS workflow



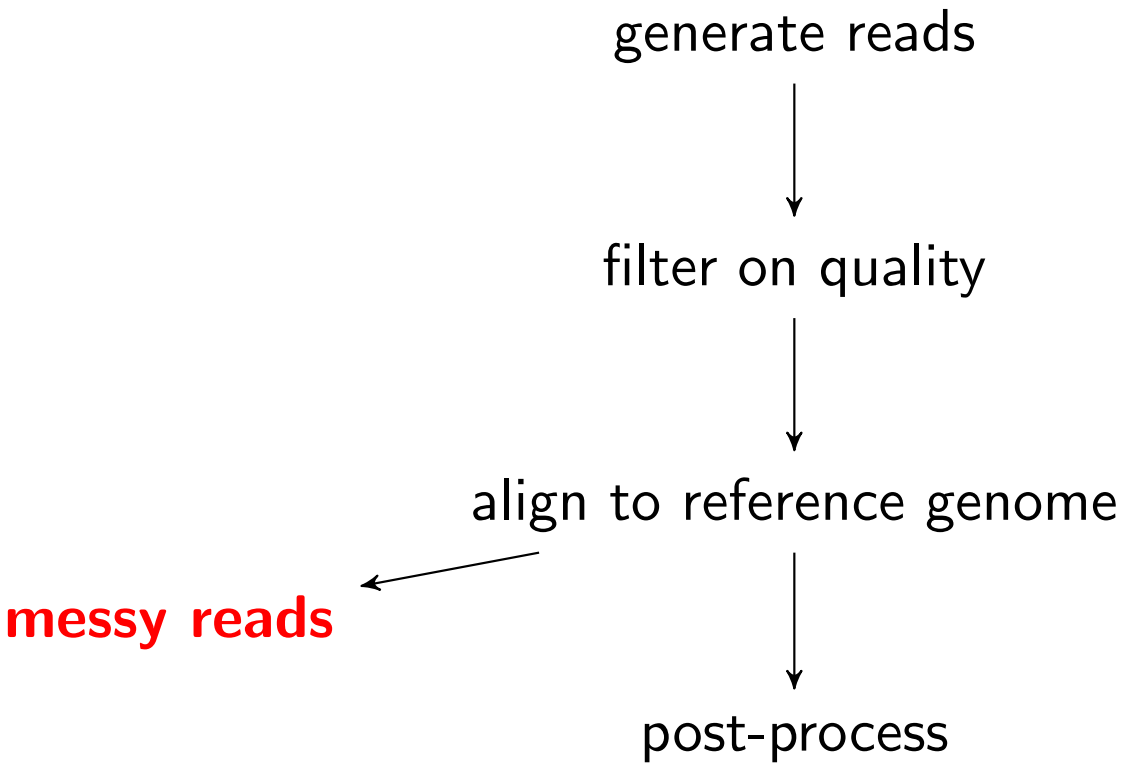
Generic NGS workflow



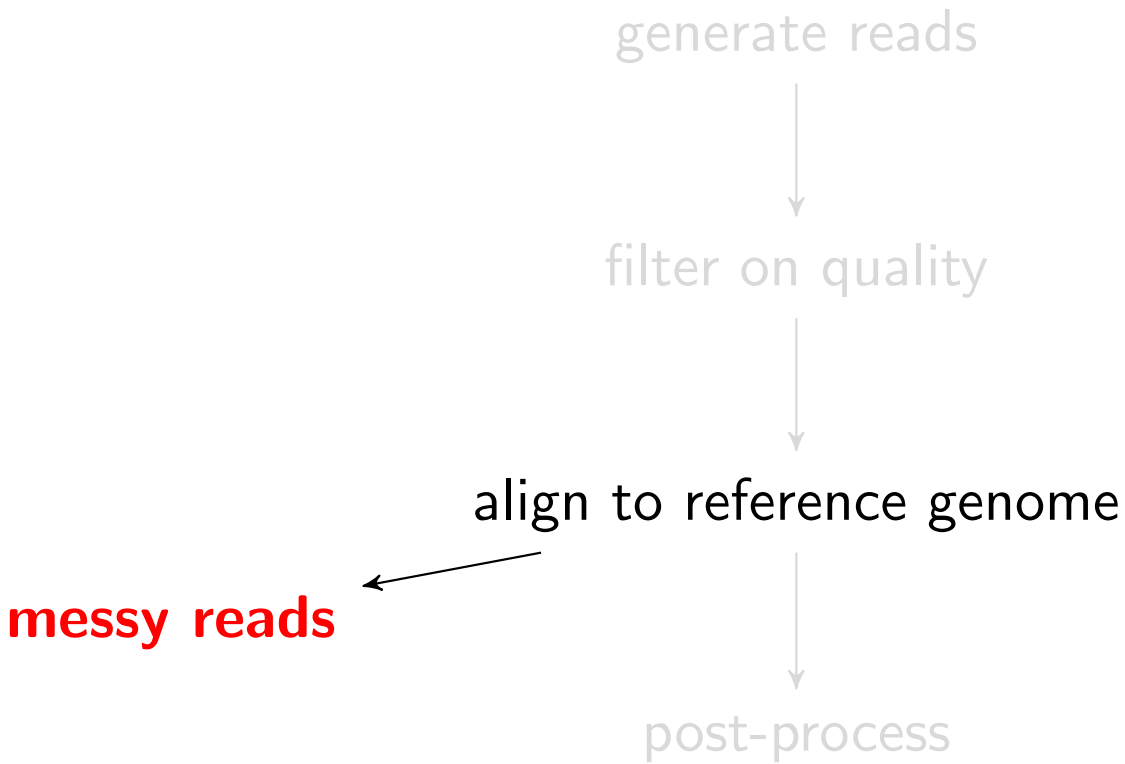
Generic NGS workflow



Generic NGS workflow



Generic NGS workflow



What is a reference genome?

THE LINEAR ARRANGEMENT OF SIX SEX-LINKED FACTORS IN DROSOPHILA, AS SHOWN BY THEIR MODE OF ASSOCIATION

A. H. STURTEVANT

From the Zoölogical Laboratory, Columbia University

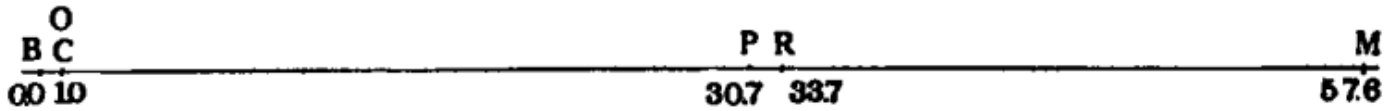


Diagram 1

Insight: genomes of multiple individuals are collinear.

What is a reference genome?

THE LINEAR ARRANGEMENT OF SIX SEX-LINKED FACTORS IN DROSOPHILA, AS SHOWN BY THEIR MODE OF ASSOCIATION

A. H. STURTEVANT

From the Zoölogical Laboratory, Columbia University

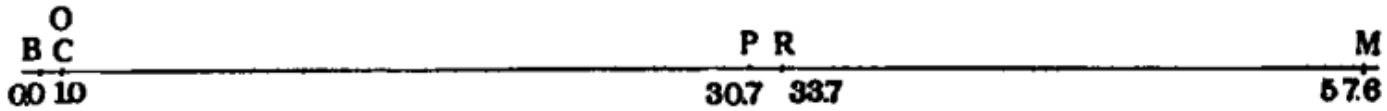


Diagram 1

Insight: genomes of multiple individuals are collinear.

What is a reference genome?

THE LINEAR ARRANGEMENT OF SIX SEX-LINKED FACTORS IN DROSOPHILA, AS SHOWN BY THEIR MODE OF ASSOCIATION

A. H. STURTEVANT

From the Zoölogical Laboratory, Columbia University

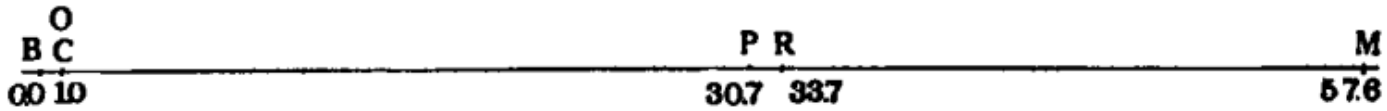
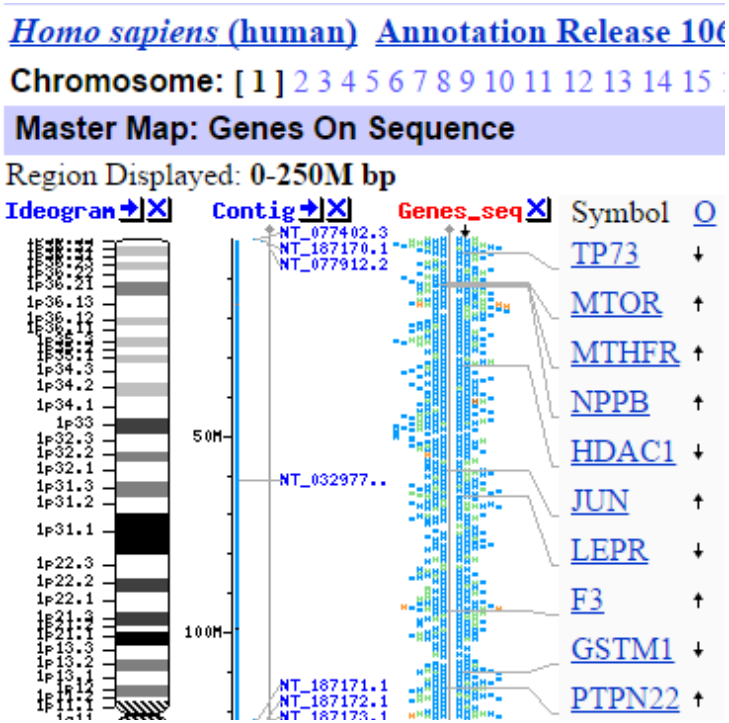


Diagram 1

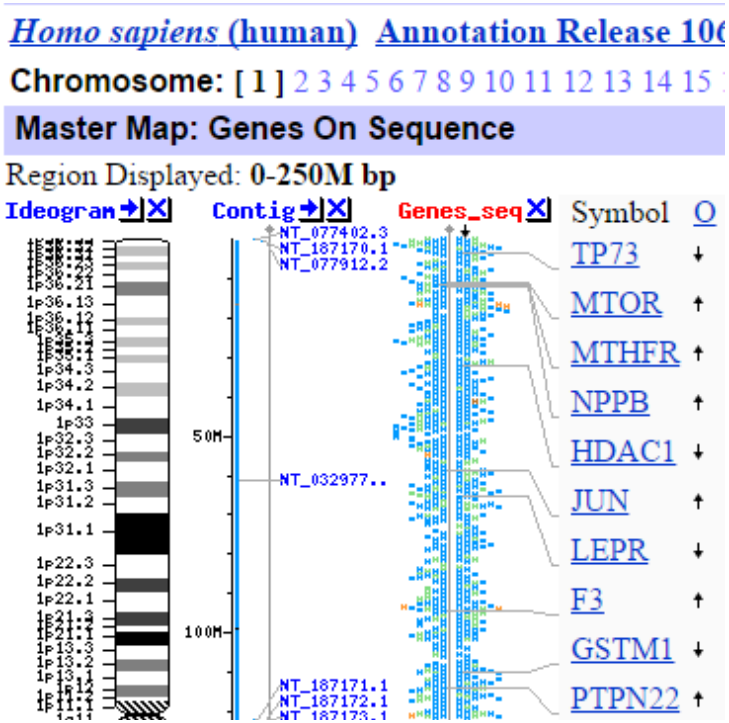
Insight: genomes of multiple individuals are collinear.

What is a reference genome?



$$\lim_{\text{data}\to\infty} (\text{physical map}) = \text{whole-genome sequence}$$

What is a reference genome?



$$\lim_{\text{data}\to\infty} (\text{physical map}) = \text{whole-genome sequence}$$

Why use a reference?

Practical reasons

- Convenience
- Efficiency
- Compression/summary
- Generalizability

Biological reasons

- Strong prior
- High-quality annotation

Why use a reference?

Practical reasons

- Convenience
- Efficiency
- Compression/summary
- Generalizability

Biological reasons

- Strong prior
- High-quality annotation

Why use a reference?

Practical reasons

- Convenience
- Efficiency
- Compression/summary
- Generalizability

Biological reasons

- Strong prior
- High-quality annotation

Inconvenient truths

1. The reference is an artificial construct.
2. The reference introduces biases which are asymmetric.
3. The most consequential genomic variation is hardest to see against the reference.

Inconvenient truths

1. The reference is an artificial construct.
2. The reference introduces biases which are asymmetric.
3. The most consequential genomic variation is hardest to see against the reference.

Inconvenient truths

1. The reference is an artificial construct.
2. The reference introduces biases which are asymmetric.
3. The most consequential genomic variation is hardest to see against the reference.

Inconvenient truths

1. The reference is an artificial construct.
2. The reference introduces biases which are asymmetric.
3. The most consequential genomic variation is hardest to see against the reference.

The reads are the data.

An unruly beast

Challenges:

- *De novo* assembly is hard
- Few tools available for exploring unaligned reads
- Raw datasets are (very) big and unindexed

An unruly beast

Challenges:

- *De novo* assembly is hard
- Few tools available for exploring unaligned reads
- Raw datasets are (very) big and unindexed

An unruly beast

Challenges:

- *De novo* assembly is hard
- Few tools available for exploring unaligned reads
- Raw datasets are (very) big and unindexed

An unruly beast

Challenges:

- *De novo* assembly is hard
- Few tools available for exploring unaligned reads
- Raw datasets are (very) big and unindexed

Enter the Burrows-Wheeler transform (BWT)

Constructing the BWT for the string `APPLE`.

rotation	sorted	BWT
⋆APPLE$	APPLE$⋆	**⋆**
$⋆APPLE	E$⋆APPL	**L**
E$⋆APPL	LE$⋆APP	**P**
LE$⋆APP	PLE$⋆AP	**P**
PLE$⋆AP	PPLE$⋆A	**A**
PPLE$⋆A	⋆APPLE$	**$**
APPLE$⋆	$⋆APPLE	**E**

The related **FM-index** allows fast *searching* and *counting* of substrings.

Enter the Burrows-Wheeler transform (BWT)

Constructing the BWT for the string `APPLE`.

rotation	sorted	BWT
⋆APPLE$	APPLE$⋆	⋆
$⋆APPLE	E$⋆APPL	L
E$⋆APPL	LE$⋆APP	P
LE$⋆APP	PLE$⋆AP	P
PLE$⋆AP	PPLE$⋆A	A
PPLE$⋆A	⋆APPLE$	$
APPLE$⋆	$⋆APPLE	E

The related **FM-index** allows fast *searching* and *counting* of substrings.

Enter the Burrows-Wheeler transform (BWT)

Constructing the BWT for the string `APPLE`.

rotation	**sorted**	**BWT**
`*APPLE$`	`APPLE$*`	**`*`**
`$*APPLE`	`E$*APPL`	**`L`**
`E$*APPL`	`LE$*APP`	**`P`**
`LE$*APP`	`PLE$*AP`	**`P`**
`PLE$*AP`	`PPLE$*A`	**`A`**
`PPLE$*A`	`*APPLE$`	**`$`**
`APPLE$*`	`$*APPLE`	**`E`**

The related **FM-index** allows fast *searching* and *counting* of substrings.

Enter the Burrows-Wheeler transform (BWT)

Constructing the BWT for the string `APPLE`.

rotation	**sorted**	**BWT**
`*APPLE$`	`APPLE$*`	**`*`**
`$*APPLE`	`E$*APPL`	**`L`**
`E$*APPL`	`LE$*APP`	**`P`**
`LE$*APP`	`PLE$*AP`	**`P`**
`PLE$*AP`	`PPLE$*A`	**`A`**
`PPLE$*A`	`*APPLE$`	**`$`**
`APPLE$*`	`$*APPLE`	**`E`**

The related **FM-index** allows fast *searching* and *counting* of substrings.

Extending BWT to many strings, efficiently

BIOINFORMATICS ORIGINAL PAPER

2014, pages 1–8
doi:10.1093/bioinformatics/btu584

Sequence analysis

Advance Access publication August 28, 2014

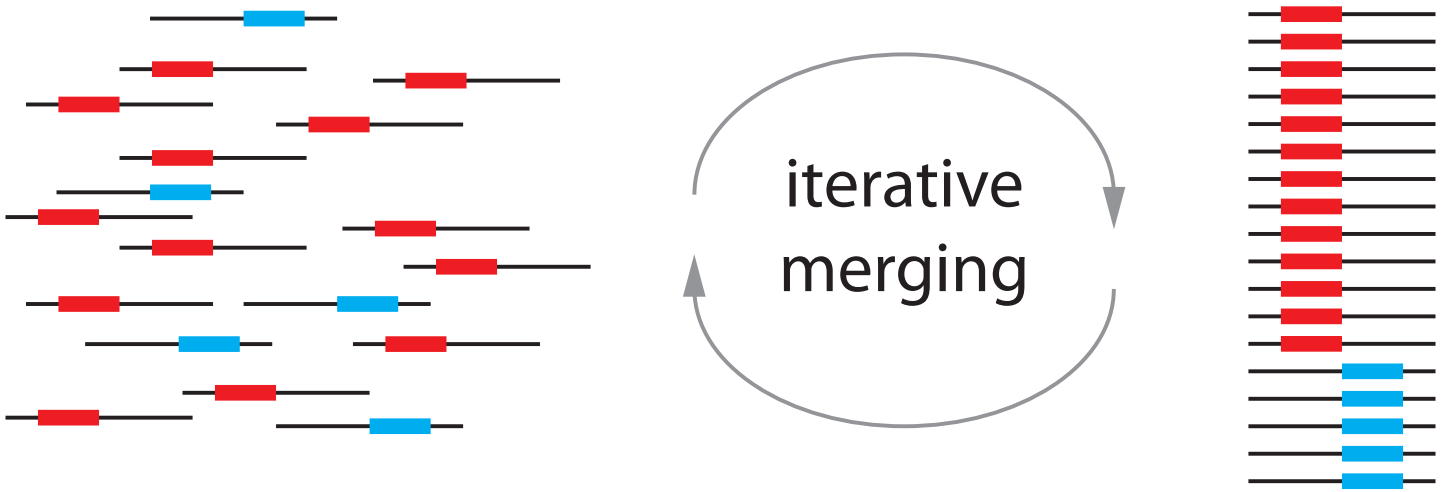
Merging of multi-string BWTs with applications

James Holt* and Leonard McMillan

Department of Computer Science, 201 S. Columbia St. UNC-CH, Chapel Hill, NC 27599, USA

Associate Editor: Michael Brudno

Extending BWT to many strings, efficiently



Theoretical properties of merged msBWT

Define

ℓ = length of longest common substring among strings

n = total number of strings

N = sum of length of all n strings

m = total number of msBWTs being merged

k = word size (for later)

Time and space requirements

- Construction takes $O(\ell N)$ time
- Final msBWT requires $O(N \log_2 m)$ space
- Searches take $O(k)$ time

Theoretical properties of merged msBWT

Define

$$\ell = \text{length of longest common substring among strings}$$
$$n = \text{total number of strings}$$
$$N = \text{sum of length of all } n \text{ strings}$$
$$m = \text{total number of msBWTs being merged}$$
$$k = \text{word size (for later)}$$

Time and space requirements

- Construction takes $O(\ell N)$ time
- Final msBWT requires $O(N \log_2 m)$ space
- Searches take $O(k)$ time

Practical properties of msBWT for NGS

- Lossless compression (and rate improves with more data)
  - can reconstitute fastq file[1] from a msBWT
- Easy to add data to an existing msBWT, without complete recomputation
- Search for arbitrary k-mer takes $O(k)$ time — no matter how many reads (!!)

Existing tools use BWT to speed up query of **reference**; we use it to speed up query of **reads**.

[1] Sort of — storage of quality scores not yet implemented.

Practical properties of msBWT for NGS

- Lossless compression (and rate improves with more data)
  - can reconstitute fastq file[1] from a msBWT
- Easy to add data to an existing msBWT, without complete recomputation
- Search for arbitrary k-mer takes $O(k)$ time — no matter how many reads (!!)

Existing tools use BWT to speed up query of **reference**; we use it to speed up query of **reads**.

[1] Sort of — storage of quality scores not yet implemented.

Practical properties of msBWT for NGS

- Lossless compression (and rate improves with more data)
  - can reconstitute fastq file[1] from a msBWT
- Easy to add data to an existing msBWT, without complete recomputation
- Search for arbitrary k-mer takes $O(k)$ time — no matter how many reads (!!)

Existing tools use BWT to speed up query of **reference**; we use it to speed up query of **reads**.

[1] Sort of — storage of quality scores not yet implemented.

Practical properties of msBWT for NGS

- Lossless compression (and rate improves with more data)
  - can reconstitute fastq file[1] from a msBWT
- Easy to add data to an existing msBWT, without complete recomputation
- Search for arbitrary k-mer takes $O(k)$ time — no matter how many reads (!!)

Existing tools use BWT to speed up query of **reference**; we use it to speed up query of **reads**.

[1] Sort of — storage of quality scores not yet implemented.

Practical properties of msBWT for NGS

- Lossless compression (and rate improves with more data)
  - can reconstitute fastq file[1] from a msBWT
- Easy to add data to an existing msBWT, without complete recomputation
- Search for arbitrary k-mer takes $O(k)$ time — no matter how many reads (!!)

Existing tools use BWT to speed up query of **reference**; we use it to speed up query of **reads**.

[1] Sort of — storage of quality scores not yet implemented.

Practical properties of msBWT for NGS

- Lossless compression (and rate improves with more data)
  - can reconstitute fastq file[1] from a msBWT
- Easy to add data to an existing msBWT, without complete recomputation
- Search for arbitrary k-mer takes $O(k)$ time — no matter how many reads (!!)

Existing tools use BWT to speed up query of **reference**; we use it to speed up query of **reads**.

[1] Sort of — storage of quality scores not yet implemented.

What can we do with it?

Basic operations:

- search for a k-mer and return strings (ie. reads) which contain it
- count occurrences of a k-mer

Example use cases:

- targeted *de novo* assembly
- finding structural variants
- analysis of mRNA splicing and editing
- profiling sequence composition
- informed design of molecular assays

What can we do with it?

Basic operations:

- search for a k-mer and return strings (ie. reads) which contain it
- count occurrences of a k-mer

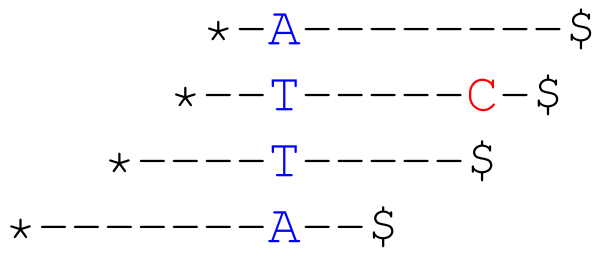
Example use cases:

- targeted *de novo* assembly
- finding structural variants
- analysis of mRNA splicing and editing
- profiling sequence composition
- informed design of molecular assays

Preliminaries

Modelling the sequencing process

Assume **reads** of length p are generated from **template** sequences at random, with per-base error rate ε.



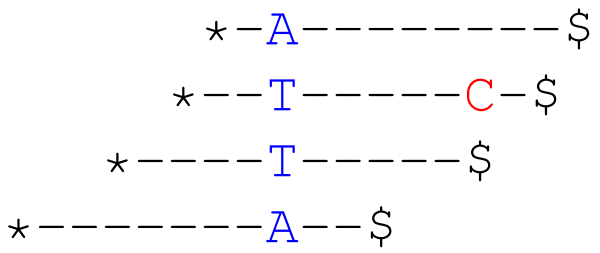
```
ACAGTCAGAGCTAGCACAGCTAGCTAACGGCCTA   (diploid template)
ACAGTCAGAGCTTGCACAGCTAGCTAACGGCCTA

ACAGTCAGAGCTTGCACAGCTAGCTAACGGCCTA   (reference)
```

We can search a msBWT for k-mers for $k \leq p$. How to choose k?

Preliminaries

Modelling the sequencing process

Assume **reads** of length p are generated from **template** sequences at random, with per-base error rate ε.



```
ACAGTCAGAGCTAGCACAGCTAGCTAACGGCCTA   (diploid template)
ACAGTCAGAGCTTGCACAGCTAGCTAACGGCCTA

ACAGTCAGAGCTTGCACAGCTAGCTAACGGCCTA   (reference)
```

We can search a msBWT for k-mers for $k \leq p$. How to choose k?

Preliminaries

Choosing a useful k-mer size

Let w be distance between sequencing errors.

$$w \sim \text{Expo}(\varepsilon) \tag{1}$$

so $\mathbb{E}[w] = 1/\varepsilon$.

Let π be the pairwise sequence divergence between the templace and the reference, and s the distance between variants.

$$s \sim \text{Expo}(\pi) \tag{2}$$

so $\mathbb{E}[s] = 1/\pi$.

Pick $k \leq \min(1/\varepsilon, 1/\pi)$.

Preliminaries

Choosing a useful k-mer size

Let w be distance between sequencing errors.

$$w \sim \text{Expo}(\varepsilon) \tag{1}$$

so $\mathbb{E}[w] = 1/\varepsilon$.

Let π be the pairwise sequence divergence between the templace and the reference, and s the distance between variants.

$$s \sim \text{Expo}(\pi) \tag{2}$$

so $\mathbb{E}[s] = 1/\pi$.

Pick $k \leq \min(1/\varepsilon, 1/\pi)$.

Preliminaries

Choosing a useful k-mer size

Let w be distance between sequencing errors.

$$w \sim \text{Expo}(\varepsilon) \tag{1}$$

so $\mathbb{E}[w] = 1/\varepsilon$.

Let π be the pairwise sequence divergence between the templace and the reference, and s the distance between variants.

$$s \sim \text{Expo}(\pi) \tag{2}$$

so $\mathbb{E}[s] = 1/\pi$.

Pick $k \leq \min(1/\varepsilon, 1/\pi)$.

Preliminaries

Interpreting k-mer counts

Let c be the count of occurrences of some k-mer, and X be estimated sequencing depth.

Case 1: $c > 0$

Note that

$$\mathbb{E}[c|k] \leq \mathbb{E}[c|k-1] \leq \ldots \mathbb{E}[c|1]$$

. . . and can choose k sufficiently large that $\mathbb{E}[c|k] \approx X$ for "nice" queries.

Case 2: $c = 0$

- this part of template not sequenced: $P(c = 0) \approx e^{-X}$
- sequencing error(s): $P(c = 0) \approx (1 - e^{-p\varepsilon})^X$
- this part of template contains variant relative to reference

Preliminaries

Interpreting k-mer counts

Let c be the count of occurrences of some k-mer, and X be estimated sequencing depth.

Case 1: $c > 0$

Note that

$$\mathbb{E}[c|k] \leq \mathbb{E}[c|k-1] \leq \ldots \mathbb{E}[c|1]$$

. . . and can choose k sufficiently large that $\mathbb{E}[c|k] \approx X$ for "nice" queries.

Case 2: $c = 0$

- this part of template not sequenced: $P(c = 0) \approx e^{-X}$
- sequencing error(s): $P(c = 0) \approx (1 - e^{-p\varepsilon})^X$
- this part of template contains variant relative to reference

Preliminaries

Interpreting k-mer counts

Let c be the count of occurrences of some k-mer, and X be estimated sequencing depth.

Case 1: $c > 0$

Note that

$$\mathbb{E}[c|k] \leq \mathbb{E}[c|k-1] \leq \ldots \mathbb{E}[c|1]$$

. . . and can choose k sufficiently large that $\mathbb{E}[c|k] \approx X$ for "nice" queries.

Case 2: $c = 0$

- this part of template not sequenced: $P(c = 0) \approx e^{-X}$
- sequencing error(s): $P(c = 0) \approx (1 - e^{-p\varepsilon})^X$
- this part of template contains variant relative to reference

Example uses of the msBWT

1. Assembling the mouse mitochondrial genome
2. A complex structural variant in mouse
3. Profiling the gut microbiota

Example 1: assembling the 17kbp mouse mitochondrial genome

pick a seed sequence ("bait")

generate de Bruijn graph

traverse Eulerian path through graph

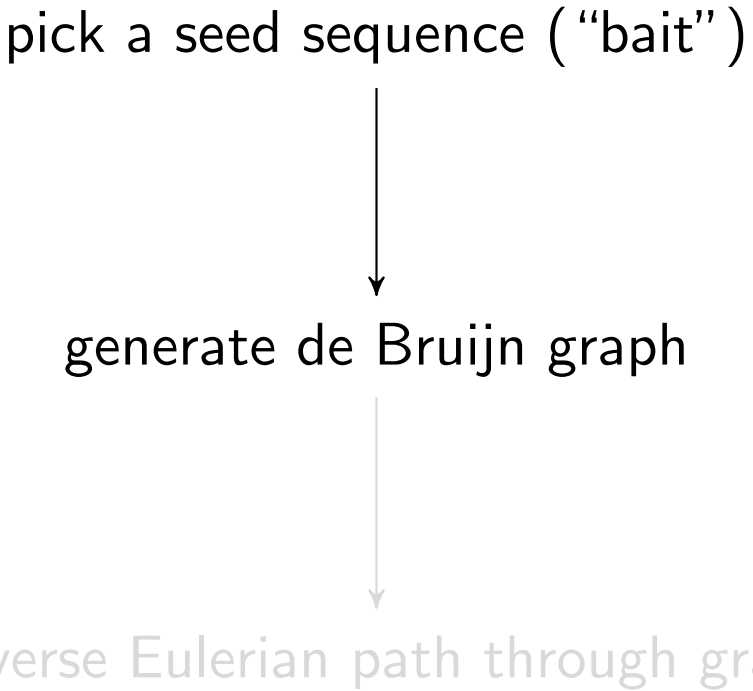
Example 1: assembling the 17kbp mouse mitochondrial genome

pick a seed sequence ("bait")

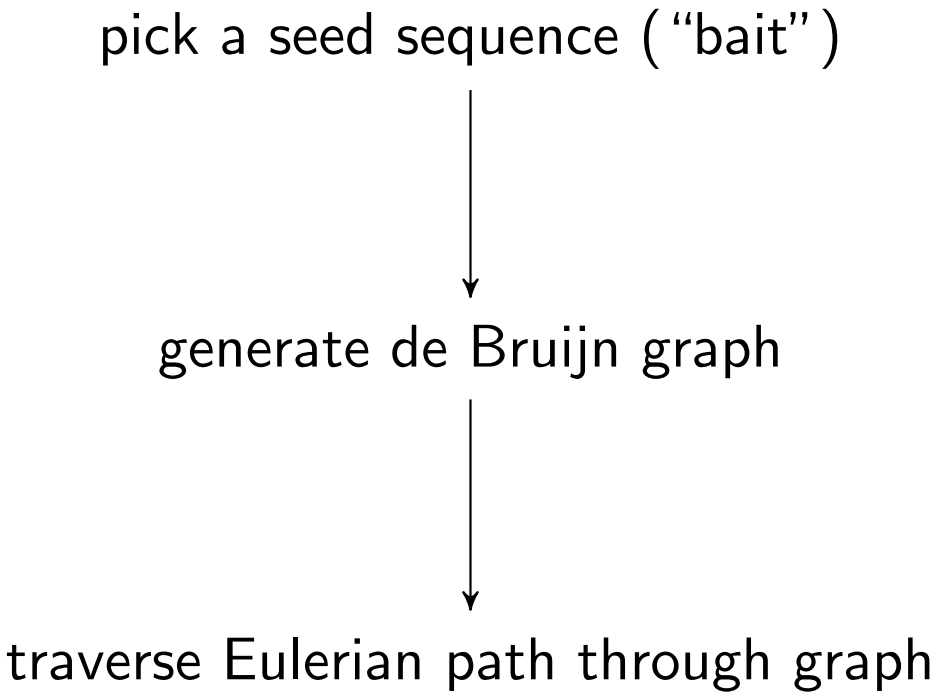
generate de Bruijn graph

traverse Eulerian path through graph

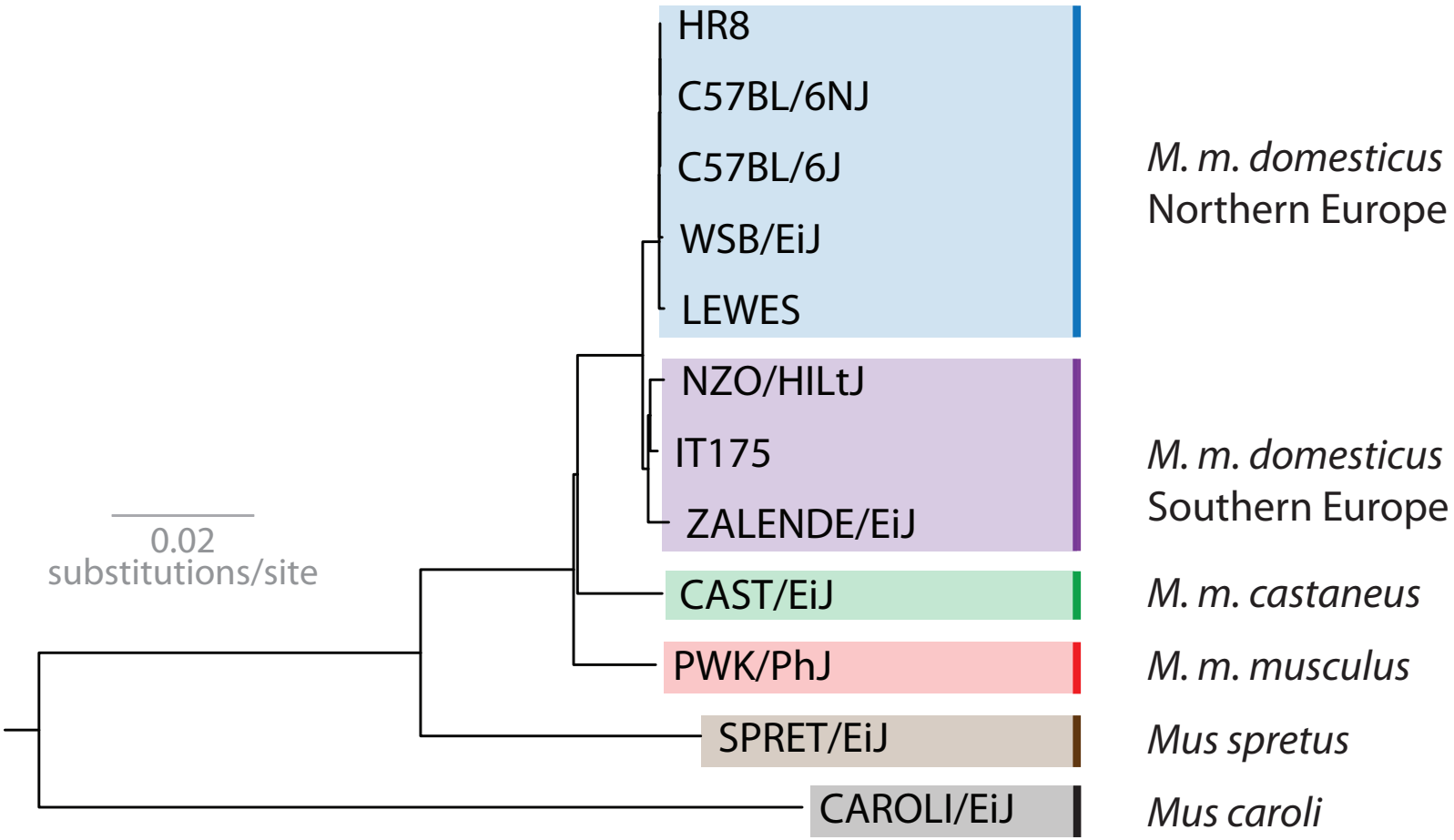
Example 1: assembling the 17kbp mouse mitochondrial genome



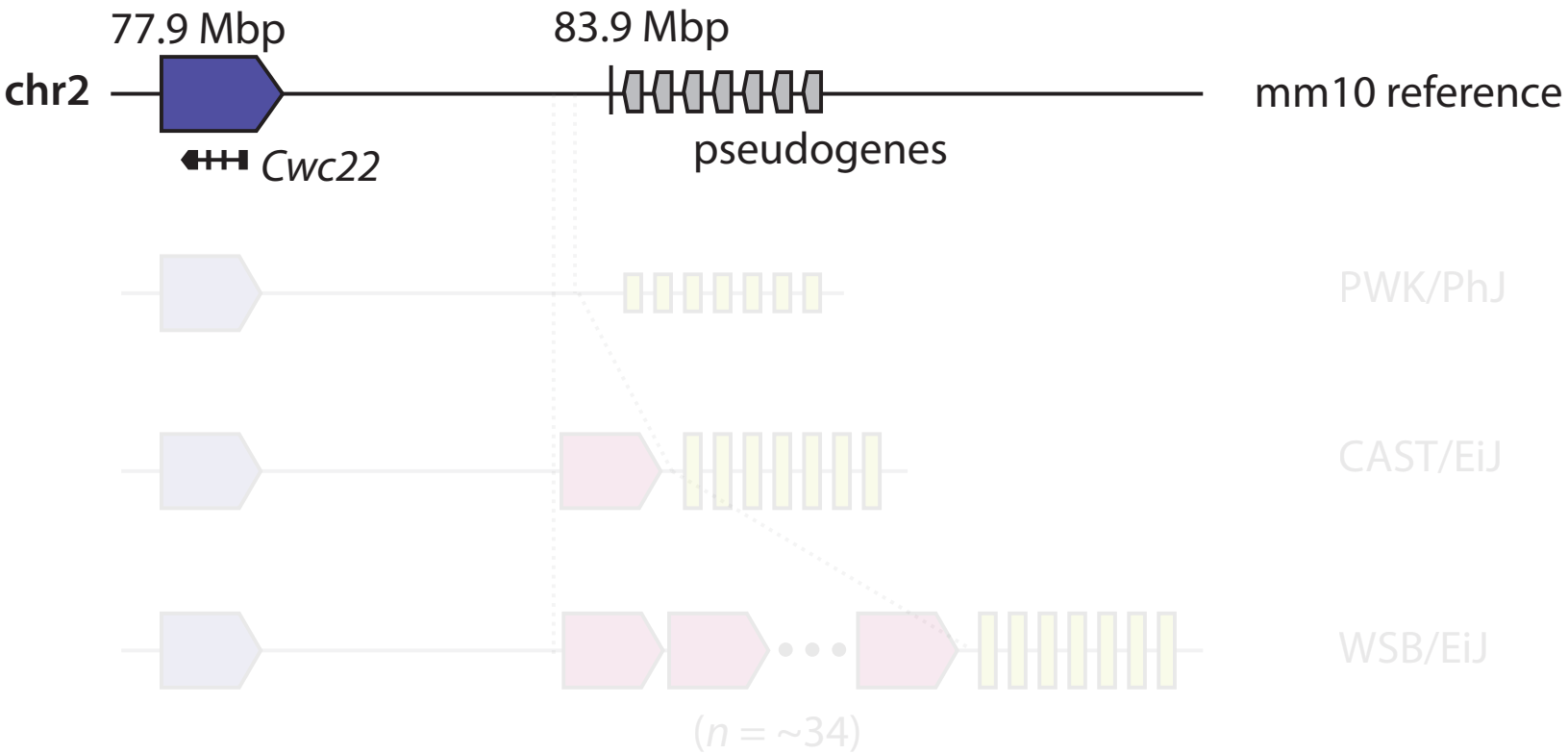
Example 1: assembling the 17kbp mouse mitochondrial genome



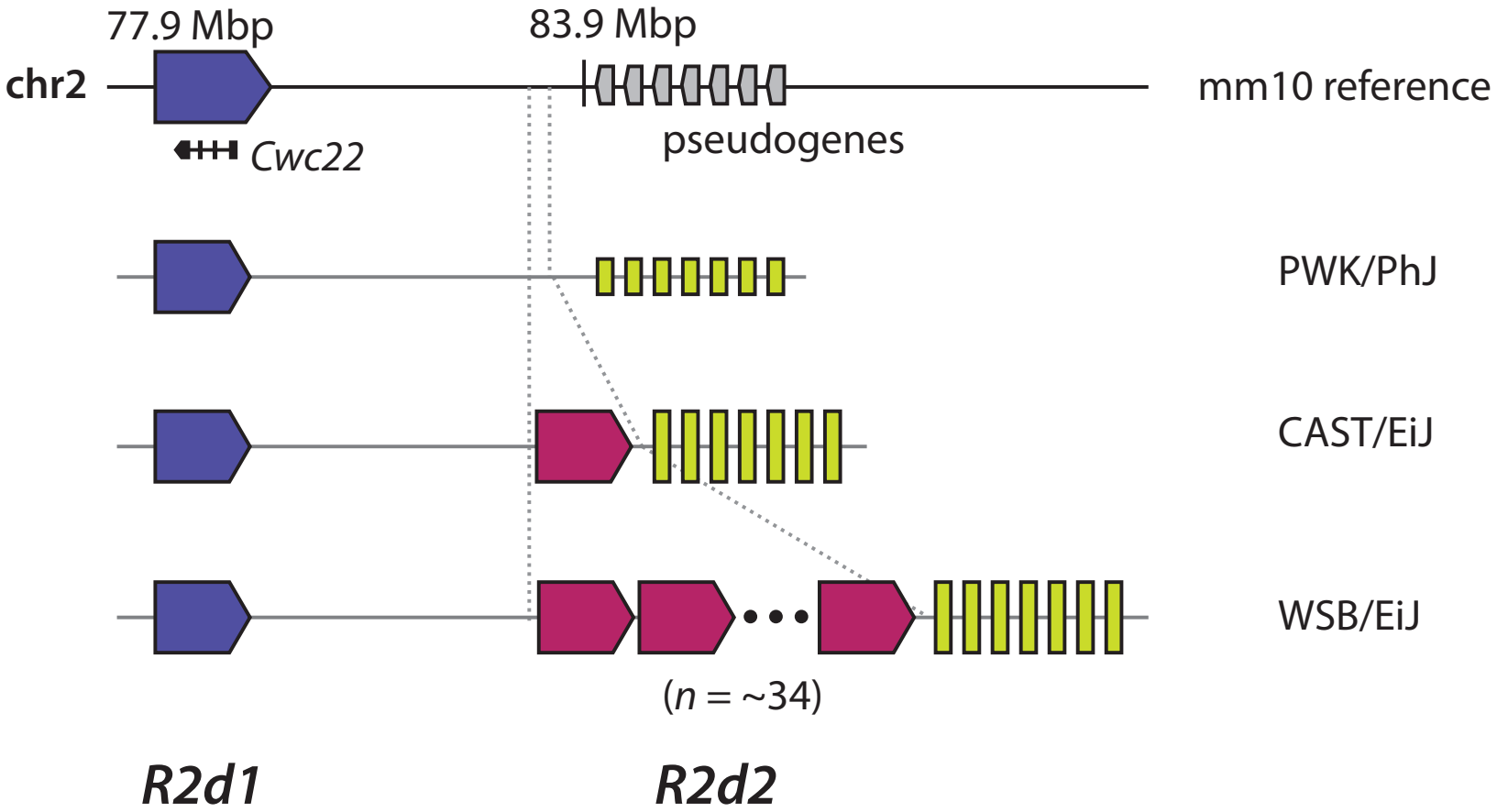
Example 1: assembling the 17kbp mouse mitochondrial genome



Example 2: Complex structural variant in mouse



Example 2: Complex structural variant in mouse



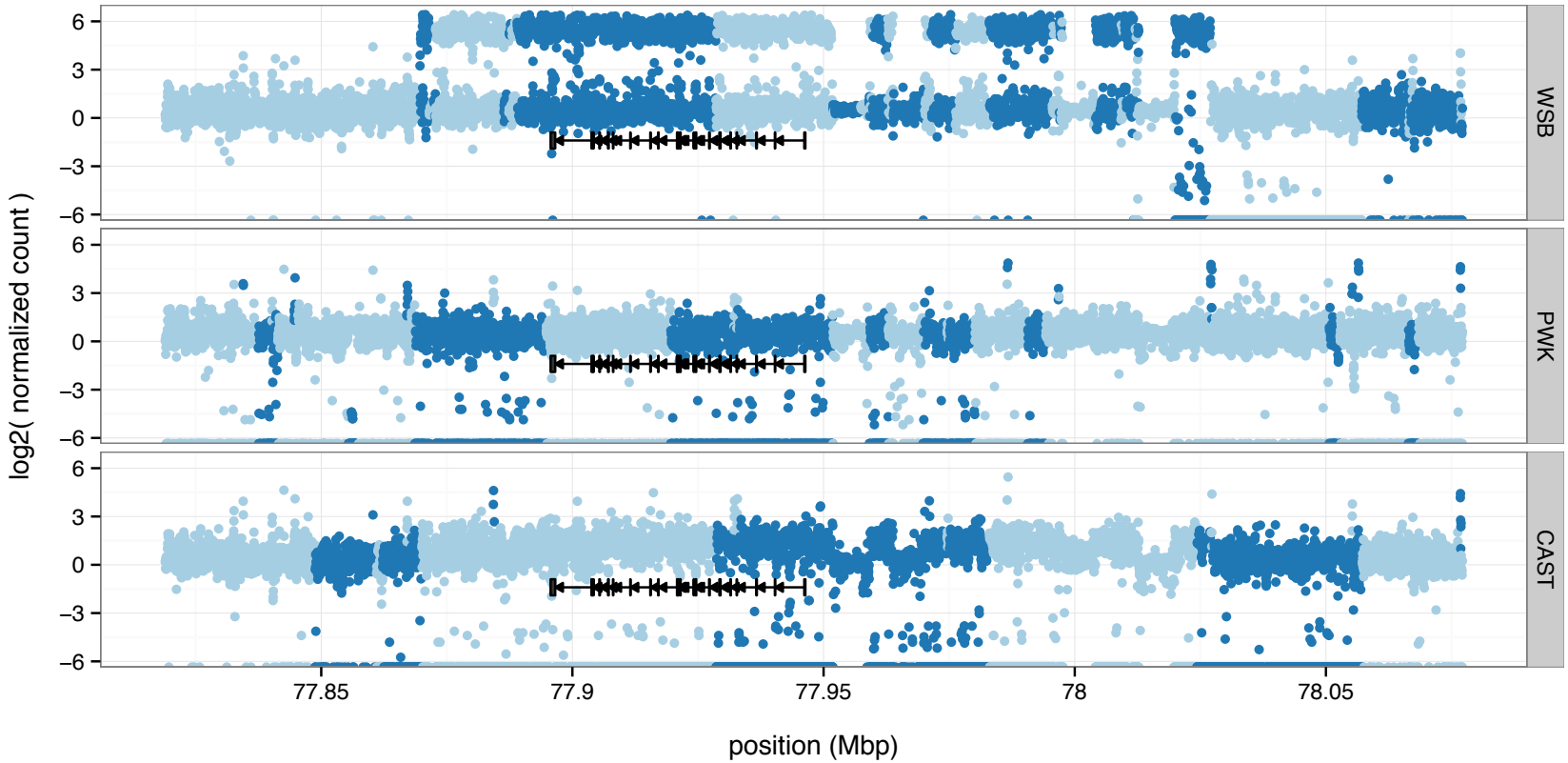
Example 2: Complex structural variant in mouse

Interactive *de novo* assembly

[online demo]

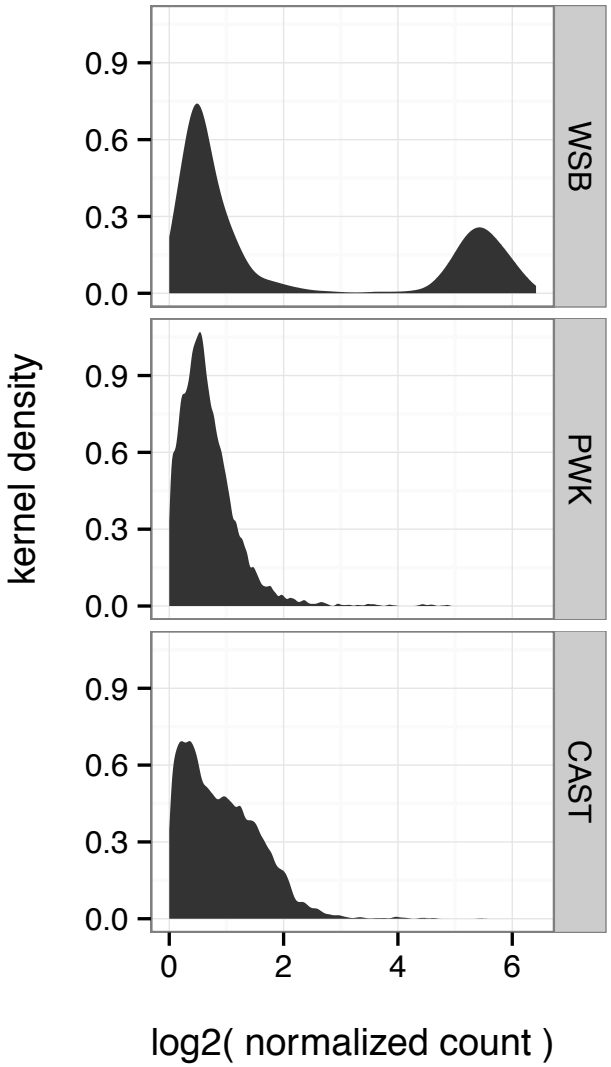
Example 2: Complex structural variant in mouse

"Confetti plots"



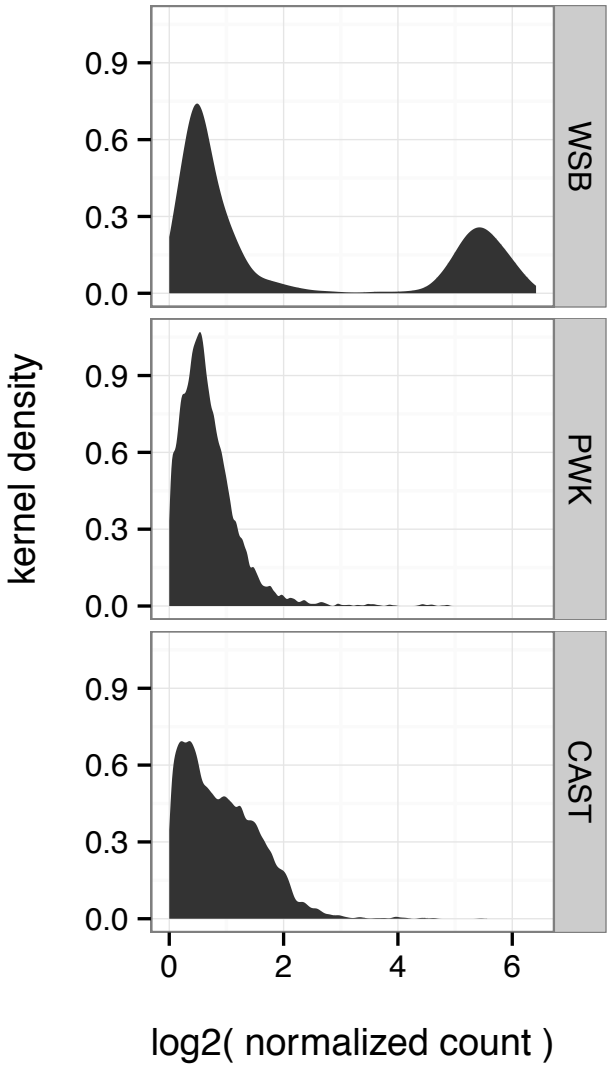
Example 2: Complex structural variant in mouse

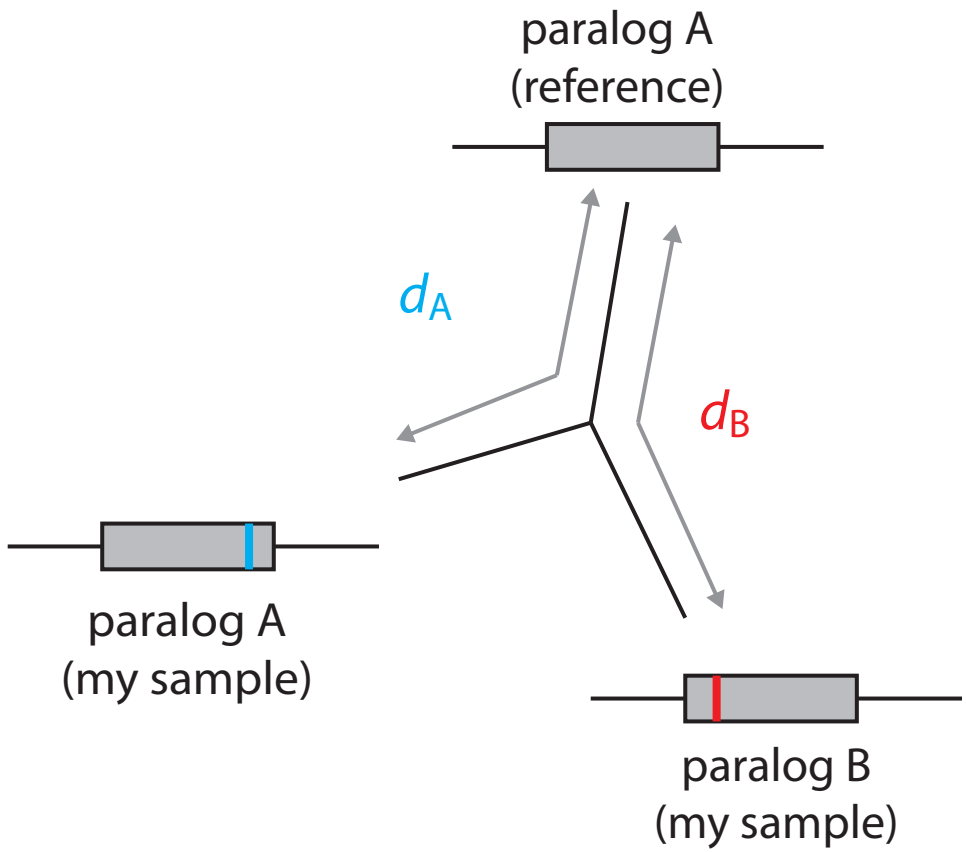
Untangling multiple copies



Example 2: Complex structural variant in mouse

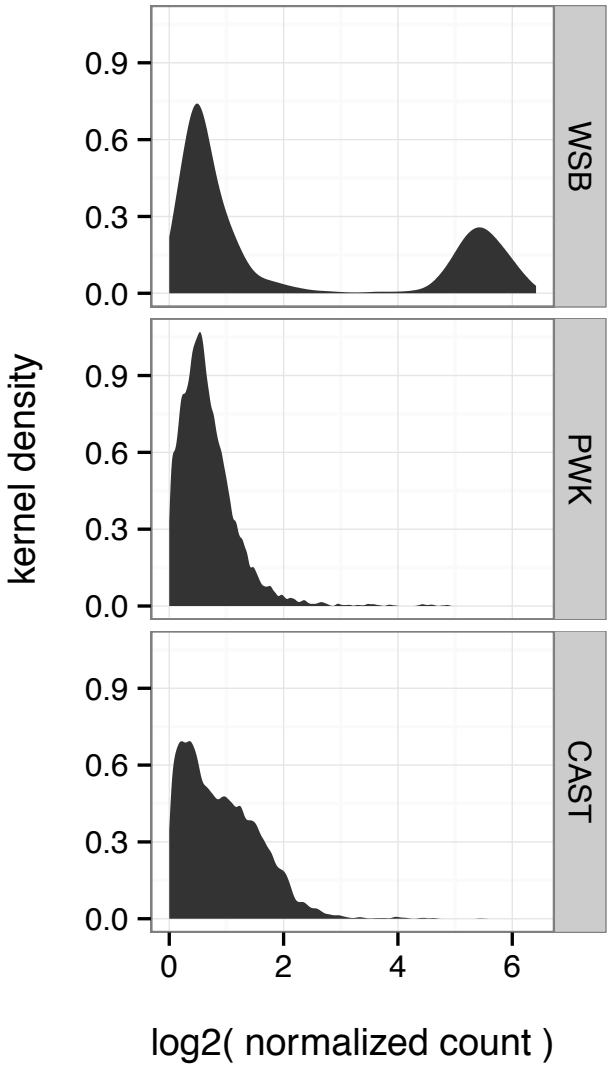
Untangling multiple copies

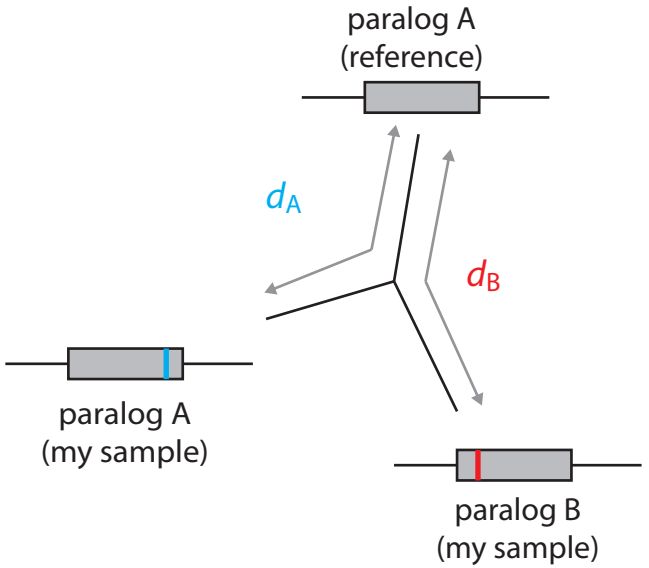




Example 2: Complex structural variant in mouse

Untangling multiple copies



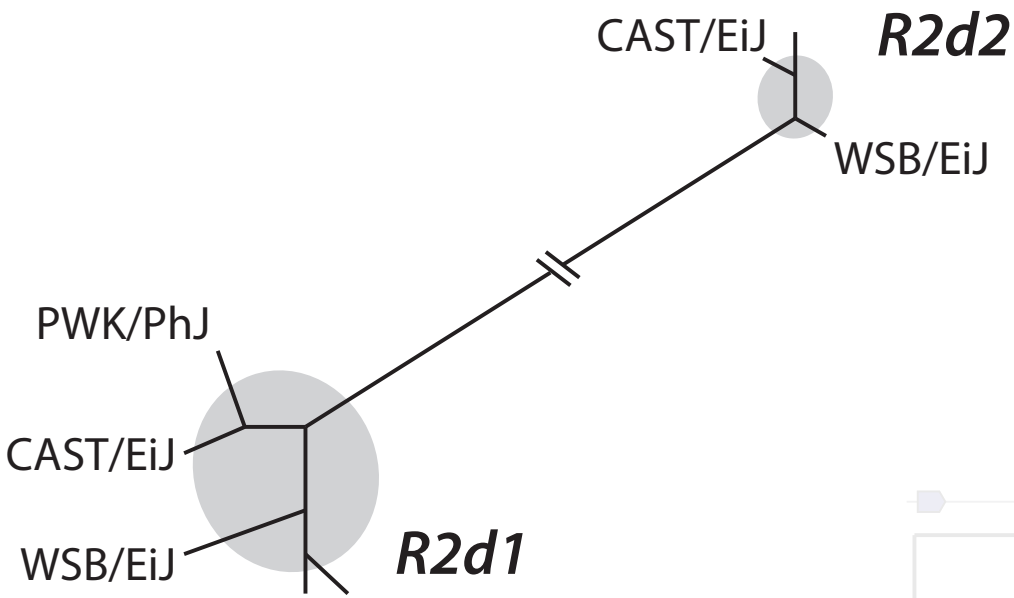


Can assign k-mers to copies using mixture model. Define mixing proportions π_i; then

$$\frac{\pi_A}{\pi_B} \approx \frac{d_A}{d_B}$$

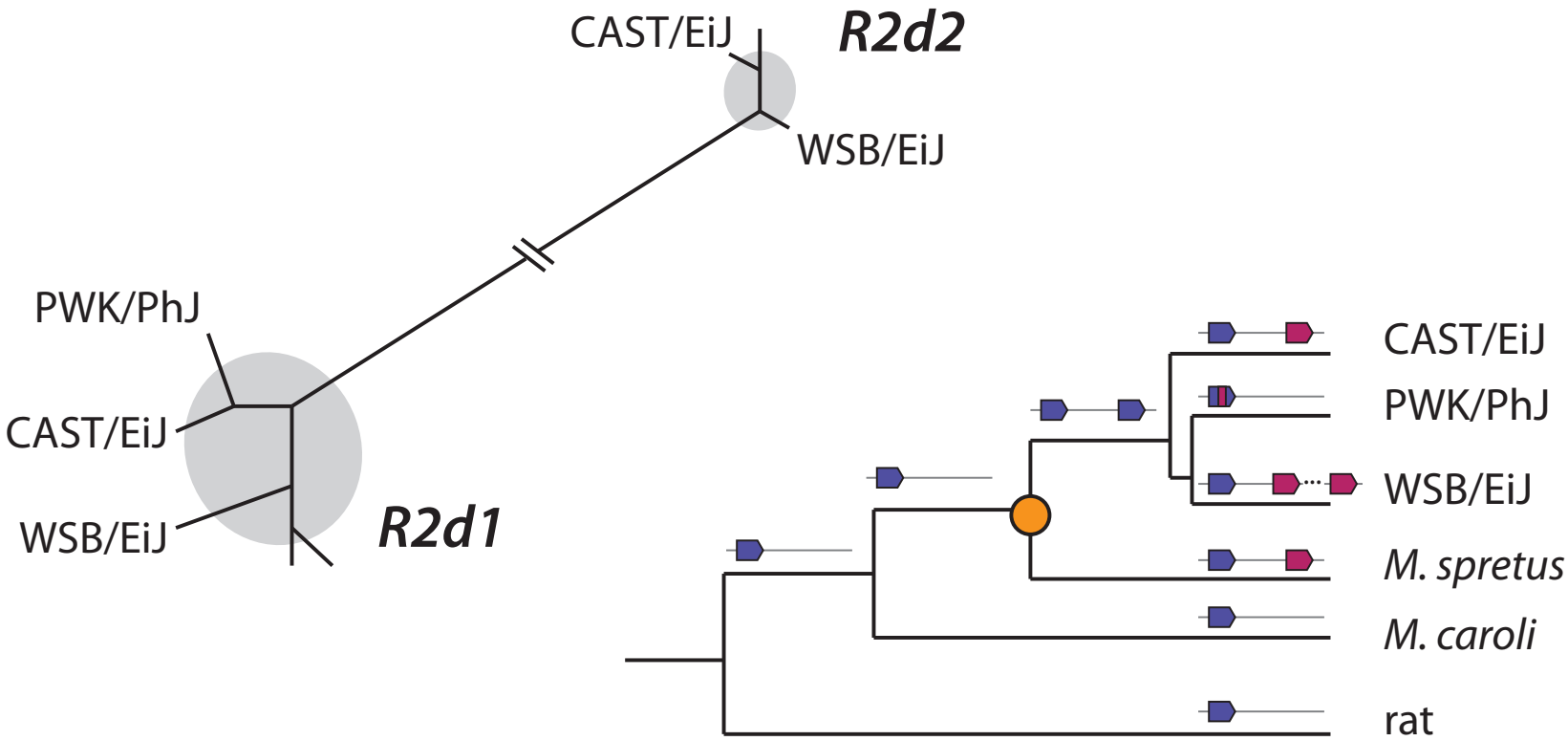
Example 2: Complex structural variant in mouse

Phylogeny with novel sequence

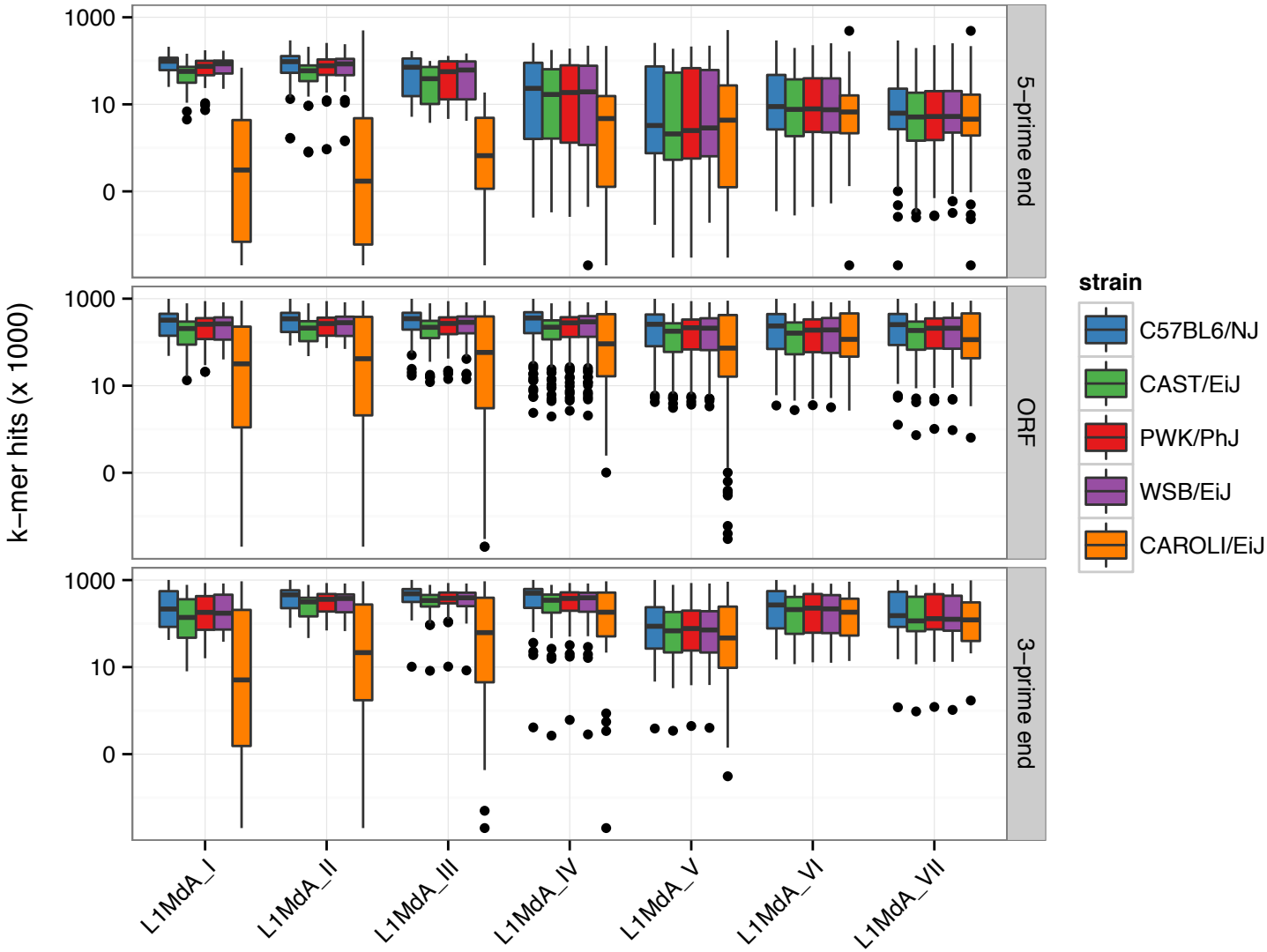


Example 2: Complex structural variant in mouse

Phylogeny with novel sequence



Aside: age of repeat elements



Applications of msBWT for molecular biology

- design of PCR assays
- design of oligonucleotide probes (qPCR, microarray)
- direct query of structural variants
- . . .

Applications of msBWT for molecular biology

- design of PCR assays
- design of oligonucleotide probes (qPCR, microarray)
- direct query of structural variants
- . . .

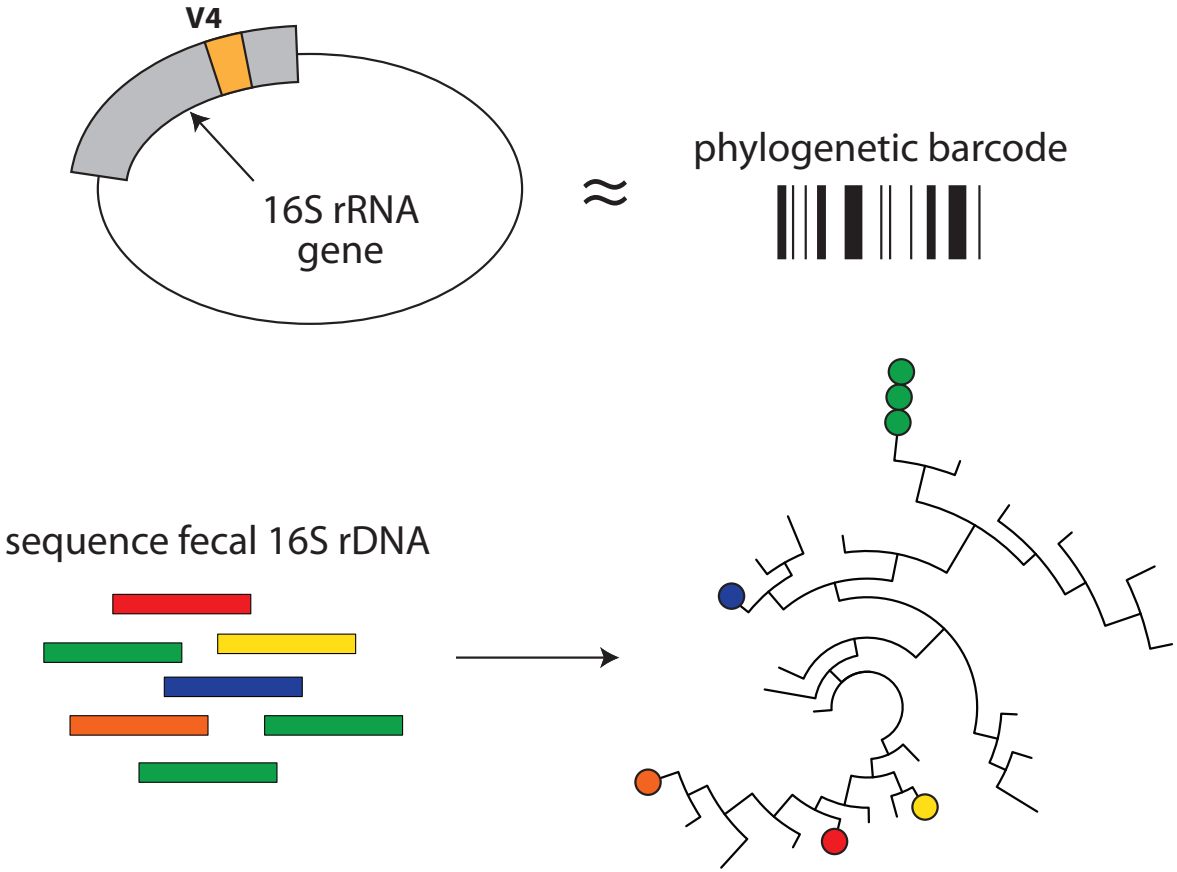
Applications of msBWT for molecular biology

- design of PCR assays
- design of oligonucleotide probes (qPCR, microarray)
- direct query of structural variants
- . . .

Applications of msBWT for molecular biology

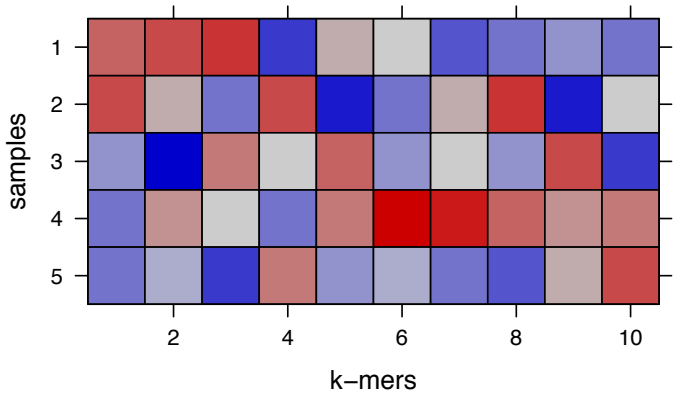
- design of PCR assays
- design of oligonucleotide probes (qPCR, microarray)
- direct query of structural variants
- ...

Example 3: Profiling the gut microbiota

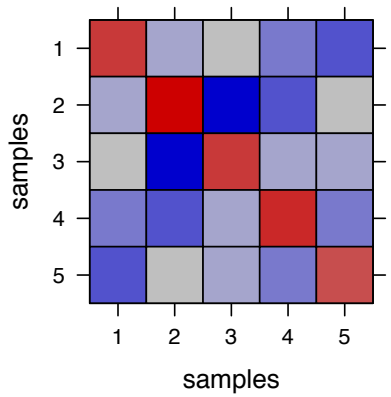


Example 3: Profiling the gut microbiota

$$\mathbf{Y}_{n \times p} =$$



$$\frac{1}{n}\mathbf{Y}\mathbf{Y}^T = \mathbf{D} =$$

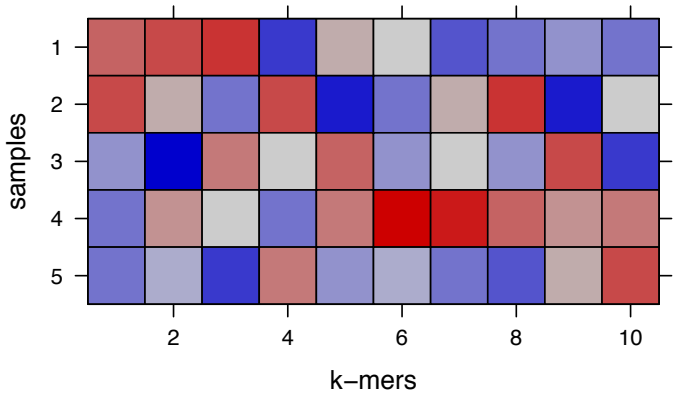


Multivariate ANOVA, given a matrix $\mathbf{X}_{n \times q} = [x_1 \ldots x_q]$ of covariates:

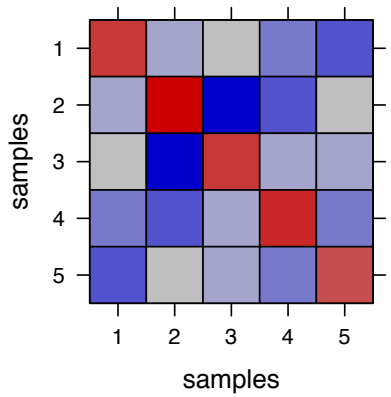
$$\mathbf{D} \sim x_1 + \cdots + x_q$$

Example 3: Profiling the gut microbiota

$$\mathbf{Y}_{n \times p} =$$



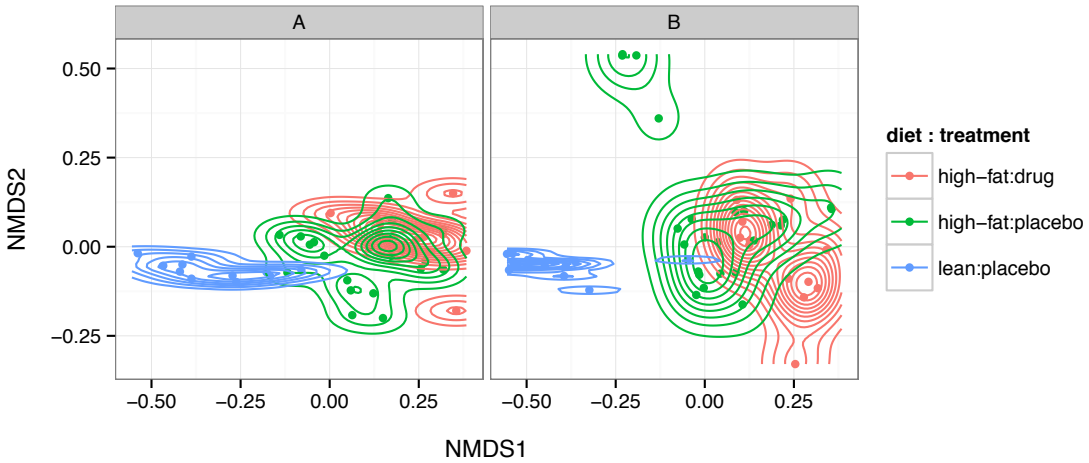
$$\frac{1}{n}\mathbf{Y}\mathbf{Y}^T = \mathbf{D} =$$



Multivariate ANOVA, given a matrix $\mathbf{X}_{n \times q} = [x_1 \ldots x_q]$ of covariates:

$$\mathbf{D} \sim x_1 + \cdots + x_q$$

Example 3: Profiling the gut microbiota



Using naïve *k*-mer frequencies from msBWT

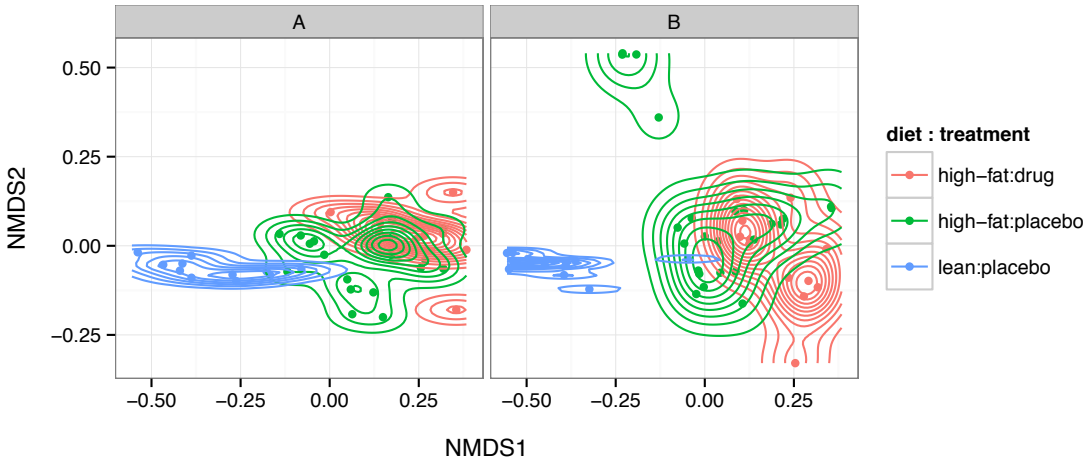
Time: ∼ 2 hours

Morgan, Crowley et al. *PLoS One*, in process.

Using `MTToolbox` + `qiime` + `UniFrac`

Time: ∼ 8 hours

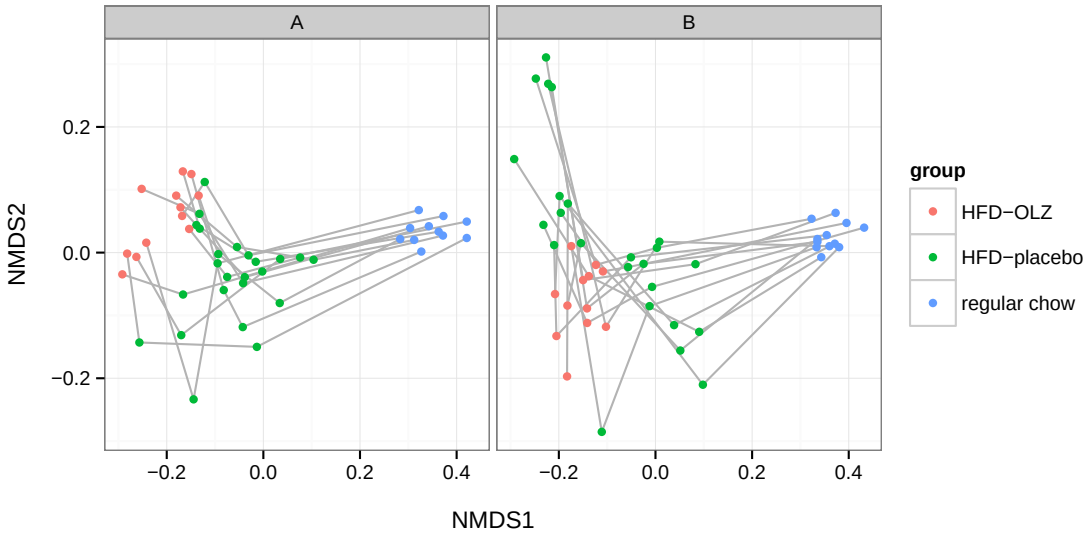
Example 3: Profiling the gut microbiota



Using naïve *k*-mer frequencies from msBWT

Time: ∼ 2 hours

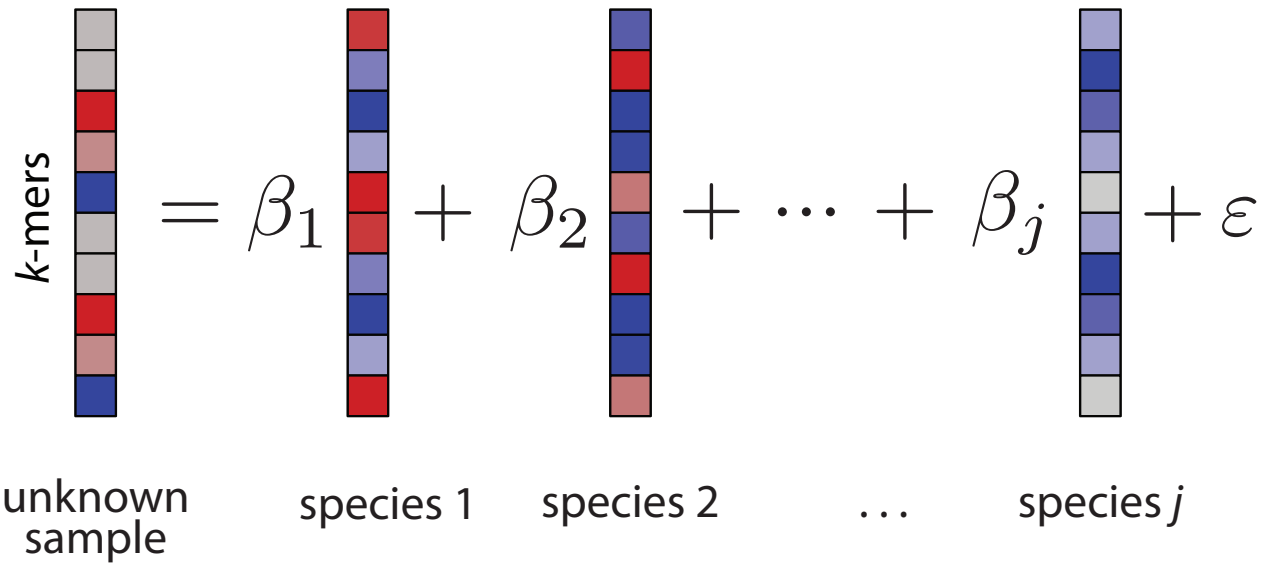
Morgan, Crowley et al. *PLoS One*, in process.



Using `MTToolbox` + `qiime` + `UniFrac`

Time: ∼ 8 hours

Extension: microbial abundance estimation from *k*-mers



Future directions (among many)

- Unsupervised structural variant detection
- Unsupervised *de novo* assembly, starting from arbitrary seed
- Joint analysis of RNAseq and DNAseq from same sample
  - splicing/isoform diversity
  - RNA editing
  - fusion transcripts
- Identification and quantification of transposable elements
- Contaminant detection
- . . .

Future directions (among many)

- Unsupervised structural variant detection
- Unsupervised *de novo* assembly, starting from arbitrary seed
- Joint analysis of RNAseq and DNAseq from same sample
  - splicing/isoform diversity
  - RNA editing
  - fusion transcripts
- Identification and quantification of transposable elements
- Contaminant detection
- . . .

Future directions (among many)

- Unsupervised structural variant detection
- Unsupervised *de novo* assembly, starting from arbitrary seed
- Joint analysis of RNAseq and DNAseq from same sample
  - splicing/isoform diversity
  - RNA editing
  - fusion transcripts
- Identification and quantification of transposable elements
- Contaminant detection
- . . .

Future directions (among many)

- Unsupervised structural variant detection
- Unsupervised *de novo* assembly, starting from arbitrary seed
- Joint analysis of RNAseq and DNAseq from same sample
  - splicing/isoform diversity
  - RNA editing
  - fusion transcripts
- Identification and quantification of transposable elements
- Contaminant detection
- . . .

Future directions (among many)

- Unsupervised structural variant detection
- Unsupervised *de novo* assembly, starting from arbitrary seed
- Joint analysis of RNAseq and DNAseq from same sample
  - splicing/isoform diversity
  - RNA editing
  - fusion transcripts
- Identification and quantification of transposable elements
- Contaminant detection
- . . .

Future directions (among many)

- Unsupervised structural variant detection
- Unsupervised *de novo* assembly, starting from arbitrary seed
- Joint analysis of RNAseq and DNAseq from same sample
  - splicing/isoform diversity
  - RNA editing
  - fusion transcripts
- Identification and quantification of transposable elements
- Contaminant detection
- . . .

Future directions (among many)

- Unsupervised structural variant detection
- Unsupervised *de novo* assembly, starting from arbitrary seed
- Joint analysis of RNAseq and DNAseq from same sample
  - splicing/isoform diversity
  - RNA editing
  - fusion transcripts
- Identification and quantification of transposable elements
- Contaminant detection
- ...

Try it yourself

Interactive queries hosted on `csbio` cluster

`www.csbio.unc.edu/CEGSseq/?run=msBWT`

Download the `msbwt` Python package

`pypi.python.org/pypi/msbwt/0.2.4`

See my (rudimentary) scripts

`github.com/andrewparkermorgan/snoop`

Questions?